An Energy-Optimal Framework for Assignment and Trajectory Generation in Teams of Autonomous Agents

Logan Beaver\textsuperscript{a}, Andreas Malikopoulos\textsuperscript{a}

\textsuperscript{a}Mechanical Engineering Department, University of Delaware, Newark, Delaware, 19716, USA

Abstract

In this paper, we present a decentralized approach to solving the problem of moving \( N \) homogeneous agents into \( N \), or more, goal locations along energy-minimizing trajectories. We propose a decentralized framework which only requires knowledge of the goal locations by each agent. The framework includes guarantees on safety through dynamic constraints, and a method to impose a dynamic, global priority ordering on the agents. A solution to the goal assignment and trajectory generation problems are derived in the form of a binary program and a linear system of equations. We also present the conditions for global optimality and characterize the assumptions under which the algorithm is guaranteed to converge to a unique assignment of agents to goals. Finally, we validate the efficacy of our approach through a numerical simulation in MATLAB.

Keywords: Multiagent systems, decentralized control, formations

1. INTRODUCTION

1.1. Motivation

Complex systems consist of diverse entities that interact both in space and time; see Malikopoulos (2016). Referring to something as complex implies that it consists of interdependent entities or agents that can adapt, i.e., they can respond to their local and global environment. Complex systems appear in many applications, including cooperation between components of autonomous systems, sensor fusion, and natural biological systems. As we move to increasingly complex systems; see Malikopoulos (2015), new control approaches are needed to optimize the impact of individuals on system-level behavior through the control of individual entities; see Malikopoulos et al. (2015), Malikopoulos (2011).

Robotic swarms are a complex system and have attracted considerable attention in many applications, e.g., transportation, exploration, construction, surveillance, and manufacturing. As discussed in Oh et al. (2017), swarms are especially attractive due to their natural parallelization and general adaptability. One of the typical multiagent applications is creating desired formations. However, due to cost constraints on any real swarm of autonomous agents, e.g., limited computation capabilities, battery capacity, and sensing capabilities, any efficient control approach needs to take into account the energy consumption of each agent. Moving agents into a desired formation has been explored previously, however, creating this formation while minimizing energy consumption is an open problem.

1.2. Related Work

Brambilla et al. (2013) classified approaches to swarms into two distinct groups, macroscopic, and microscopic. Macroscopic approaches generate group behavior from a system of partial differential equations which are spatially discretized and applied to individual agents; this approach is fundamentally based on work in Turing (1952),
and is used extensively in bio-inspired formation and pattern forming; see Oh et al. (2015). Our approach is microscopic; that is, we control the behavior of individual agents to achieve some desired global outcome. Microscopic approaches are based on the seminal work in Reynolds (1987), which applied an agent-based method to capture the flocking behavior of birds.

There is a rich literature of work on the creation of a desired formation, Guo et al. (2010), Hanada et al. (2007) construct rigid formations from triangular sub-structures, Song and O’Kane (2016) presented a formation algorithm inspired by crystal growth, and Lee and Nak (2008) explored growing swarm formations in a lattice structure. It is possible to build formations using only scalar, bearing, or distance measurements to move agents into a desired formation; see Swartling et al. (2014), Lin et al. (2004). In Olfati-Saber et al. (2007) it was proven that many formation problems may be solved by applying a modified form of the basic consensus algorithm. However, none of these approaches consider the energy cost to individual agents in the swarm.

A significant amount of work has applied optimization methods to designing potential fields for agent interaction; see Wang and Xin (2013), Sun and Cassandras (2015), Xu and Carrillo (2015), Rajasree (2010), Vásárhelyi et al. (2018). However, these approaches optimize the shape of the potential field, and do not consider the energy consumption of individual agents. Some work has been done to generate optimal assignments using a centralized planner; see Turpin et al. (2013b). Other approaches require global information about the system (e.g., Turpin et al. (2013b) requires globally unique assignments a priori, Morgan et al. (2016) requires the graph diameter of the system, and Rubenstein et al. (2012) imposes global ‘seed’ agents on the swarm.).

Our approach is decentralized, and thus each agent may only partially observe the entire system state. The latter results in a non-classical information structure, and many techniques for solving centralized systems do not hold; see Dave and Malikopoulos (2019). To address this, one may impose a priority ordering on the agents. This has been done previously through a centralized controller; see Turpin et al. (2013a), Chalaki and Malikopoulos (2019). In general, finding an optimal ordering is generally NP-Hard and an optimal ordering is not always guaranteed to exist; see Ma et al. (2019). To reduce the complexity of ordering agents, much work has been done to decentralize the ordering problem. This includes applying discretized path-based heuristics; see Wu et al. (2019), and reinforcement learning; see Sartoretti et al. (2018). In contrast, our approach applies a decentralized method of dynamically ordering agents which is path independent and relies only on information directly observable by each agent.

The three major contributions of this paper are: (1) a decentralized set of interaction dynamics, which dynamically impose a priority order on agents in a decentralized manner, (2) an assignment algorithm that exploits the dynamics of the agents and considers the relative speed between each agent and the available formation positions, and (3) guarantees on the stability of our proposed control policy under relaxed assumptions relative to work reported in the literature.

1.3. Organization of This Paper

The remainder of this paper proceeds as follows. In Section 2, we formulate the decentralized optimal control problem, and we decompose it into the coupled assignment and trajectory generation subproblems. In Section 3, we present the conditions which guarantee system convergence along with the assignment problem. Then, in Section 4, we prove that these conditions are satisfied by our framework and solve the optimal trajectory generation problem. Finally, in Section 5, we present a series of MATLAB simulations to show the performance of the algorithm, and we presented concluding remarks in Section 6.

2. PROBLEM FORMULATION

We consider a swarm of \( N \in \mathbb{N} \) autonomous agents indexed by the set \( \mathcal{A} = \{1, \ldots, N\} \). Our objective is to designed a decentralized control framework to move the \( N \) agents into \( M \in \mathbb{N} \) goal positions, indexed by the set \( \mathcal{F} = \{1, \ldots, M\} \). We consider the case where \( N \leq M \), i.e., no redundant agents are brought to fill the formation. This requirement can be relaxed by defining a behavior for excess agents, such as idling; see Turpin et al. (2014). Each agent \( i \in \mathcal{A} \) follows double-integrator dynamics,

\[
\dot{\mathbf{p}}_i(t) = \mathbf{v}_i(t),
\]

\[
\dot{\mathbf{v}}_i(t) = \mathbf{u}_i(t),
\]
where \( \mathbf{p}_i(t) \in \mathbb{R}^2 \) and \( \mathbf{v}_i(t) \in \mathbb{R}^2 \) are the time-varying position and velocity vectors respectively, \( \mathbf{u}_i(t) \in \mathbb{R}^2 \) is the control input (acceleration/deceleration) over time \( t \in [0, t'_i] \), where \( t'_i \in \mathbb{R}_{>0} \) is the terminal time for agent \( i \). Additionally, each agent’s control input and velocity are bounded by

\[
\begin{align*}
\mathbf{p}_i(t) &= \mathbf{p}_i(0) + \int_0^t \mathbf{u}_i(s) \, ds \\
\mathbf{v}_i(t) &= \mathbf{v}_i(0) + \int_0^t \mathbf{u}_i(s) \, ds
\end{align*}
\]

where \( \mathbf{p}_i(t) \) and \( \mathbf{v}_i(t) \) are the position and velocity vectors respectively, \( \mathbf{u}_i(t) \) is the control input (acceleration/deceleration) over time \( t \in [0, t'_i] \), where \( t'_i \in \mathbb{R}_{>0} \) is the terminal time for agent \( i \). Additionally, each agent’s control input and velocity are bounded by

\[
\begin{align*}
\mathbf{v}_{\min} \leq ||\mathbf{v}_i(t)|| &\leq \mathbf{v}_{\max}, \\
\mathbf{u}_{\min} \leq ||\mathbf{u}_i(t)|| &\leq \mathbf{u}_{\max},
\end{align*}
\]

where \( ||\cdot|| \) is the Euclidean norm. Thus, the state of each agent \( i \in \mathcal{A} \) is given by the time-varying vector

\[
x_i(t) = \begin{bmatrix} \mathbf{p}_i(t) \\ \mathbf{v}_i(t) \end{bmatrix}.
\]

The energy consumption of any agent \( i \in \mathcal{A} \) is given by

\[
\dot{E}_i(t) = \frac{1}{2} ||\mathbf{u}_i(t)||^2.
\]

We select the \( L^2 \) of the control input as our energy model since, in general, acceleration/deceleration requires more energy than applying no control input. Therefore, we expect that minimizing the acceleration/deceleration of each agent to yield a proportional reduction in energy consumption.

Our objective is to develop a decentralized framework for the \( N \) agents to optimally, in terms of energy, create any desired formation of \( M \) points while avoiding collisions between agents.

**Definition 1.** The *desired formation* is the set of time-varying vectors \( \mathcal{G} = \{ \mathbf{p}_j(t) \in \mathbb{R}^2 \mid j \in \mathcal{F} \} \). The set \( \mathcal{G} \) can be prescribed offline, i.e., by a human designer, or online by a high-level planner.

In this framework, the agents are cooperative and capable of communication within a neighborhood, which we define next.

**Definition 2.** The *neighborhood* of agent \( i \in \mathcal{A} \) is the time-varying set

\[
\mathcal{N}_i(t) = \left\{ j \in \mathcal{A} \mid ||\mathbf{p}_i(t) - \mathbf{p}_j(t)|| \leq h \right\},
\]

where \( h \in \mathbb{R} \) is the sensing and communication horizon of each agent.

An agent \( i \in \mathcal{A} \) is also able to measure the relative position of any neighboring agent, \( j \in \mathcal{N}_i(t) \). This relative position is denoted by the vector

\[
s_i(t) = \mathbf{p}_j(t) - \mathbf{p}_i(t).
\]

Each agent \( i \in \mathcal{A} \) occupies a closed disk of radius \( R \); hence, to guarantee safety for agent \( i \in \mathcal{A} \) we impose the following constraints for all agents \( i \in \mathcal{A}, j \in \mathcal{N}_i(t), j \neq i \),

\[
||s_{ij}(t)|| > 2R, \quad \forall t \in [0, t'_i],
\]

\[
h >> 2R.
\]

Condition (8) ensures that no two agents collide, and (9) is a system-level constraint which ensures agents are able to detect each other prior to a collision.

In our modeling framework we impose the following assumptions:

**Assumption 1.** The state \( x_i(t) \) for each agent \( i \in \mathcal{A} \) is perfectly observed and there is negligible communication delay between the agents.

Assumption 1 is required to evaluate the idealized performance of the generated optimal solution.

**Assumption 2.** At most two agents violate (8) while following unconstrained trajectories under an optimal assignment.

Assumption 2 may be strong, but it has been shown in the literature that unconstrained trajectories are generally feasible; see Morgan et al. (2016). Additionally, agents tend to generate trajectories which do not cross; see Turpin et al. (2014). This assumption can be relaxed if the agents are given enough computing power to solve a nonlinear boundary-value differential equation numerically.

**Assumption 3.** The energy cost of communication is negligible; the only energy consumption is in the form of (6).

The strength of this assumption is application dependent. For cases with long-distance communications or high data rates, the trade-off between communication and
motion costs can be controlled by varying the sensing and communicating radius, $h$, of the agents.

To solve the desired formation problem, we first relax the inter-agent collision avoidance constraint to decouple the agent trajectories. This decoupling reduces the problem from a single mixed-integer program to a coupled pair of binary and quadratic programs, which we solve sequentially. This decoupling is common in the literature; see Turpin et al. (2013b), Morgan et al. (2016), and usually does not affect the outcome of the assignment problem.

Next, we present some preliminary results before decomposing the desired formation problem into the two subproblems, minimum-energy goal assignment and trajectory generation.

2.1. Preliminaries

For any agent $i \in \mathcal{A}$ using double-integrator dynamics, (1) and (2), energy model, (6), and travelling between two fixed states, the unconstrained minimum-energy trajectory is given in Malikopoulos et al. (2018) as

$$
\mathbf{u}_i(t) = \mathbf{a}_i t + \mathbf{b}_i, \\
\mathbf{v}_i(t) = \frac{\mathbf{a}_i}{2} t^2 + \mathbf{b}_i t + \mathbf{c}_i, \\
\mathbf{p}_i(t) = \frac{\mathbf{a}_i}{6} t^3 + \frac{\mathbf{b}_i}{2} t^2 + \mathbf{c}_i t + \mathbf{d}_i,
$$

where $\mathbf{a}_i, \mathbf{b}_i, \mathbf{c}_i,$ and $\mathbf{d}_i$ are constant vectors of integration determined by the linear system of equations

$$
\begin{bmatrix}
\frac{1}{6} t_0^3 & \frac{1}{2} t_0^2 & t_0 & 1 & \mathbf{a}_i \\
\frac{1}{2} t_0^2 & t_0 & 0 & \mathbf{b}_i \\
\frac{1}{6} t_i^3 & \frac{1}{2} t_i^2 & t_i & 1 & \mathbf{c}_i \\
\frac{1}{2} t_i^2 & t_i & 0 & \mathbf{d}_i
\end{bmatrix}
\begin{bmatrix}
\mathbf{u}_i(t_0) \\
\mathbf{v}_i(t_0) \\
\mathbf{p}_i(t_i)
\end{bmatrix}
= \begin{bmatrix}
\mathbf{a}_i \\
\mathbf{b}_i \\
\mathbf{c}_i \\
\mathbf{d}_i
\end{bmatrix}.
$$

Thus, the energy consumed for any unconstrained trajectory of agent $i \in \mathcal{A}$ travelling to goal $j \in \mathcal{F}$ is given by

$$
E_i^j(t) = \int_{t_0}^{t_f} ||\mathbf{u}_i(\tau)||^2 d\tau = (t_f^3 - t_i^3) \left( \frac{a_{ix}^2 + a_{iy}^2}{3} \right) + (t_f^2 - t_i^2) (a_{ix} b_{ix} + a_{iy} b_{iy}) + (t_f - t_i) \left( \frac{b_{ix}^2 + b_{iy}^2}{2} \right),
$$

where $t \in [t_0, t_f]$, and $\mathbf{a}_i = [a_{ix}, a_{iy}]^T$, $\mathbf{b}_i = [b_{ix}, b_{iy}]^T$, and $\mathbf{c}_i = [c_{ix}, c_{iy}]^T$ are determined by the solution of (13).

Next we present the interaction dynamics between agents. To resolve any conflict between agents, we consider the following objectively measurable constants:

1. Neighborhood size,
2. Energy required to reach goal,
3. Agent index.

Each of the above quantities has an associated indicator function for comparing two agents $i, j \in \mathcal{A}$, $i \neq j$,

$$
\mathbb{1}_{i > j}^N(t) := \begin{cases} 1 & |N_i(t)| > |N_j(t)|, \\
0 & |N_i(t)| \leq |N_j(t)|, \end{cases}
$$

$$
\mathbb{1}_{i > j}^E(t) := \begin{cases} 1 & E_i(t) > E_j(t), \\
0 & E_i(t) \leq E_j(t), \end{cases}
$$

$$
\mathbb{1}_{i > j}^C(t) := \begin{cases} 1 & i > j, \\
0 & i < j. \end{cases}
$$

Next we define the interaction dynamics by combining (15) - (17).

**Definition 3.** We define the interaction dynamics between any agent $i \in \mathcal{A}$ and another agent $j \in N_i(t), j \neq i$ as

$$
\Delta_i^j(t) = \mathbb{1}_{i > j}^N(t) + (1 - \mathbb{1}_{i > j}^N(t)) (1 - \mathbb{1}_{i > j}^E(t))
\left( \mathbb{1}_{i > j}^E(t) + (1 - \mathbb{1}_{i > j}^E(t))(1 - \mathbb{1}_{i > j}^C(t)) \right),
$$

where $\mathbb{1}_{i > j}^C = 1$ implies agent $i$ has priority over agent $j$, and $\mathbb{1}_{i > j}^C = 0$ implies that agent $j$ has priority over agent $i$.

The interaction dynamics are instantaneously and noiselessly measurable by each agent under Assumption 1. Whenever two agents have a conflict (i.e., share an assigned goal, or have overlapping assignments) (18) is used to impose an order on the agents such that higher priority agents act first.

**Remark 1.** For any pair of agents $i \in \mathcal{A}, j \in N_i(t), j \neq i$, it is always the case that $\Delta_i^j(t) = 1 - \Delta_i^j(t)$, i.e., the outcome of the interaction dynamics (18) is always unambiguous and therefore imposes an order on any pair of agents.

Remark 1 can be proven by simply enumerating all cases of (15) - (17).
3. Optimal Goal Assignment

The optimal solution to the assignment problem must assign each agent to a goal such that the total unconstrained energy cost, given by (14), is minimized. In our framework, each agent \( i \in \mathcal{A} \) only has information about the positions of its neighbors, \( j \in \mathcal{N}_i(t) \), and the goal index set, \( \mathcal{F} \). Agent \( i \) derives the goal assignment using a binary matrix \( \mathbf{A}_i(t) \), which we define next.

**Definition 4.** For each agent \( i \in \mathcal{A} \), we define the assignment matrix, \( \mathbf{A}_i(t) \) as an \( |\mathcal{N}_i(t)| \times |\mathcal{F}| \) matrix with binary elements. The elements of \( \mathbf{A}_i(t) \) map each agent to exactly one goal, and each goal to no more than one agent.

The assignment matrix (Definition 4) for agent \( i \in \mathcal{A} \) assigns all agents in \( \mathcal{N}_i(t) \) by considering the cost (14). We discuss the details of the optimal assignment problem later in this section.

Next we define the prescribed goal, which determines how each agent \( i \in \mathcal{A} \) assigns itself a goal.

**Definition 5.** We define the prescribed goal for agent \( i \in \mathcal{A} \) as the goal assigned to agent \( i \) by the rule,

\[
p^g_i(t) = \{ p_k \in \mathcal{G} \mid a_{ik} = 1, \ a_k \in \mathbf{A}_i(t), \ k \in \mathcal{F} \}, \tag{19}
\]

where \( \mathbf{A}_i(t) \) is given by Definition 4, and the right hand side must be a singleton set, i.e., agent \( i \) must be assigned to exactly one goal.

Next, we present the goal assignment algorithm in terms of some agent \( i \in \mathcal{A} \). However, as this framework is cooperative, each step is performed by all individuals simultaneously.

In some cases, multiple agents may select the same prescribed goal. This may occur when two agents \( i, j \in \mathcal{A}, j \neq i \) have different neighborhoods and use conflicting information to solve their local assignment problem. This motivates the introduction of competing agents, which we define next.

**Definition 6.** For agent \( i \in \mathcal{A} \), we define the set of competing agents as

\[
\mathcal{C}_i(t) = \{ k \in \mathcal{N}_i(t) \mid p^g_i(t) = p^g_k(t) \}.
\]

When \(|\mathcal{C}_i| > 1\) there are at least two agents, \( i, j \in \mathcal{N}_i(t) \) which are assigned to the same goal. All but one agent in \( \mathcal{C}_i(t) \) must be permanently banned from goal \( p_i \). Next, we define the banned goal set.

**Definition 7.** The banned goal set for agent \( i \in \mathcal{A} \) is defined as

\[
\mathcal{B}_i(t) = \{ g \in \mathcal{F} \mid p^g_i(\tau) = p_k(\tau) \in \mathcal{G}, \ \left( \prod_{j \in \mathcal{F}_i(\tau), j \neq k} 1_{ji}(\tau) = 0, \ \exists \ \tau \in [t_0, t] \right) \} \tag{20}
\]

i.e., the set of all goals which agent \( i \in \mathcal{A} \) had a conflict over and did not have priority per Definition 3.

After the conflicts are resolved for agent \( i \in \mathcal{A} \), if the size of \( \mathcal{B}_i(t) \) has increased then the value of \( t_i \) must also be increased by

\[
t_i = t + T, \tag{21}
\]

where \( t \) is the current time, and \( T \in \mathbb{R}_{>0} \) is a system parameter. This allows agent \( i \) a sufficient amount of time to reach its new goal.

Finally, for each subsequent assignment, when \( \mathcal{B}_i(t) \neq \emptyset \), agent \( i \in \mathcal{A} \) broadcasts its banned goal set to all \( j \in \mathcal{N}_i(t) \). The assignment problem is then iterated until the condition

\[
|\mathcal{C}_i(t)| = 1 \ \forall j \in \mathcal{N}_i(t), \tag{22}
\]

is satisfied. We enforce the banned goals through a constraint on the assignment problem, which follows.

**Problem 1 (Goal Assignment).** Each agent \( i \in \mathcal{A} \) selects its prescribed goal (Definition 5) by solving the following binary program.

\[
\min_{a_{jk} \in \mathbb{A}} \left\{ \sum_{j \in \mathcal{N}_i(t)} \sum_{k \in \mathcal{F}} a_{jk} E^g_{jk}(t) \right\}, \tag{23}
\]

subject to

\[
\sum_{j \in \mathcal{F}} a_{jk} = 1, \ k \in \mathcal{N}_i(t), \tag{24}
\]

\[
\sum_{k \in \mathcal{N}_i(t)} a_{jk} \leq 1, \ j \in \mathcal{F}, \tag{25}
\]

\[
a_{jk} = 0, \ \forall \ j \in \mathcal{B}_k(t), \ k \in \mathcal{N}_i(t), \tag{26}
\]

\[
a_{jk} \in \{0, 1\}.
\]

This process is repeated by each agent, \( i \in \mathcal{A} \), until (22) is satisfied for all \( j \in \mathcal{N}_i(t) \).
As the conflict condition in Problem 1 explicitly depends on the neighborhood of agent \( i \in \mathcal{A} \), Problem 1 may need to be recalculated each time the neighborhood of agent \( i \) switches. Under some assumptions about the trajectories of each agent, the assignments generated by Problem 1 are guaranteed to bring each agent to a unique goal; we show this with the help of Lemma 1.

**Lemma 1 (Solution Existence).** For an agent \( i \in \mathcal{A} \), if \( \left| \bigcup_{j \in N_i(t)} B_j(t) \right| \geq |N_i(t)| \), then the feasible region of Problem 1 is nonempty for agent \( i \).

**Proof.** Let the set of goals available to all agents in the neighborhood of agent \( i \in \mathcal{A} \) be denoted by the set

\[
\mathcal{V}_i(t) = \{ p \in \mathcal{G} \mid p \notin B_j(t), \forall j \in N_i(t) \}. \tag{27}
\]

Let the injective function \( w : \mathcal{A} \to \mathcal{F} \) map each agent to a goal. Since \( |N_i(t)| \leq |\mathcal{V}_i(t)| \), such \( w \) exists. As \( w \) is injective, the imposed mapping must satisfy (24) and (25). Likewise, \( \mathcal{V}_i(t) \cap B_j(t) = \emptyset \) for all \( j \in N_i(t) \). Thus, \( w \) must satisfy (26). Therefore, the mapping imposed by the function \( w \) is a feasible solution to Problem 1. \( \square \)

Intuitively, Lemma 1 only requires that one agent does not ban many agents from many goals. Due to the minimum-energy nature of our framework this scenario is unlikely; additionally, the permanent banning may be relaxed to temporary banning, such that the premise of Lemma 1 is always satisfied. Next we show that for a sufficiently large value of \( t' \), the convergence of all agents to goals is guaranteed.

**Theorem 1 (Assignment Convergence).** Under the assumptions of Lemma 1, for a sufficiently large value of \( t' \), and if the energy-optimal trajectories for agent \( i \in \mathcal{A} \) never increase the unconstrained energy cost (14), then \( t'_i \) must have an upper bound for all \( i \in \mathcal{A} \).

**Proof.** Let \( \{ g_n \}_{n \in \mathbb{N}} \) be the sequence of goals assigned to agent \( i \in \mathcal{A} \) by the solution of Problem 1. By Lemma 1, \( \{ g_n \}_{n \in \mathbb{N}} \) must not be empty, and the elements of this sequence are integers bounded by \( 1 \leq g_n \leq | \mathcal{F} | \). Thus, the range of this sequence is compact, and the sequence must be either (1) finite, or (2) convergent, or (3) periodic.

1) For a finite sequence there is nothing to prove, as \( t'_i \) is constant.

2) Under the discrete metric, an infinite convergent sequence requires that there exists \( N \in \mathbb{N}_{>0} \) such that \( g_n = p \) for all \( n > N \) for some formation index \( p \in \mathcal{F} \). This reduces to case 1, as \( t'_i \) does not increase for repeated assignments to the same goal.

3) By the Bolzano-Weierstrass Theorem, an infinite non-convergent sequence \( \{ g_n \}_{n \in \mathbb{N}} \) must have a convergent subsequence, i.e., agent \( i \) is assigned to some subset of goals \( I \subseteq \mathcal{G} \) infinitely many times with some constant number of intermediate assignments, \( P_k \), for each goal \( g \in I \). Necessarily, \( I \cap B_j(t) = \emptyset \) for all \( t \in [0, t'_i] \) from the construction of the banned goal set. This implies that, by the update method of \( t'_i \), the position of all goals, \( g(t) \in I \) must only be considered at time \( t'_i \), which we denote as \( g := g(t'_i) \in I \).

This implies that the goals available to agent \( i \), i.e., \( I = \mathcal{G} \setminus B_t \), must be shared between \( n > 0 \) other periodic agents. Hence, at some time \( t_1 \) that a goal, \( g \in I \), must be an optimal assignment for agent \( i \), a non optimal assignment at time \( t_2 > t_1 \) and an optimal assignment at time \( t_3 \) which corresponds to the \( P_k \) assignment. This implies \( t_3 > t_2 > t_1 \), and that the energy required to move agent \( i \) to goal \( g \) satisfies

\[
E^i_1(t_1) \leq E^i_k(t_1), \tag{28}
\]
\[
E^i_k(t_2) \leq E^i_k(t_2), \tag{29}
\]
\[
E^i_k(t_3) \leq E^i_k(t_3). \tag{30}
\]

for some goal \( k \in I, k \neq g \). For agent \( i \) to follow an energy optimal trajectory under our premise, it must always decrease the energy required to reach assigned goal, which implies

\[
E^i_k(t_1) \geq E^i_k(t_2), \tag{31}
\]
\[
E^i_k(t_2) \geq E^i_k(t_3), \tag{32}
\]

this implies

\[
E^i_k(t_1) \geq E^i_k(t_3), \tag{33}
\]

which is satisfied for every period \( P_g \) and for all goals \( g' \in I \). This is only possible if agent \( i \) simultaneously approaches all goals \( k \in I \), which requires these goals to be arbitrarily close. However, this violates the minimum spacing requirement of the goals; therefore, no such periodic behavior may exist. \( \square \)
Next, we provide the optimal trajectory generation for each agent and prove that the resulting trajectories always satisfy the premise of Theorem 1.

4. Optimal Trajectory Generation

After the goal assignment is complete, each agent must generate a collision-free trajectory to their assigned goal. The trajectories must minimize the agent’s total energy cost subject to dynamic, boundary, and collision constraints. The initial and final state constraints for each agent $i \in \mathcal{A}$ are given by

$$ p_i(t^0) = p_i^0, \quad v_i(t^0) = v_i^0, \quad (34) $$

$$ p_i(t^f) = p_i^f, \quad v_i(t^f) = v_i^f, \quad (35) $$

where the conditions at $t^f$ come from the solution of Problem 1. We may then formulate the decentralized optimal trajectory generation problem.

**Problem 2.** For each agent $i \in \mathcal{A}$, find the optimal control input, $u_i(t)$, which minimizes the energy consumption of agent $i$ and satisfies its boundary conditions and safety constraints.

$$ \min_{u_i(t)} \frac{1}{2} \int_{t^0}^{t^f} ||u_i(t)||^2 \, dt, \quad (36) $$

subject to: (1), (2), (8), (34), and (35). (37)

We solve Problem 2 by applying Hamiltonian analysis; see Bryson and Ho (1975). The case where the control and state constraints are active has been extensively studied; see Malikopoulos et al. (2018). Thus, we consider the bounds are finite, but large enough to never become active. First, the safety constraint (8) must be derived until the control input $u_i(t)$ appears. To ensure smoothness in the derivatives we use the equivalent squared form of (8). This yields

$$ N_i(t) = \begin{bmatrix} 4R^2 - s_{ij}(t) \cdot \dot{s}_{ij}(t) & -s_{ij}(t) \cdot \dot{\dot{s}}_{ij}(t) \cr -s_{ij}(t) \cdot \dot{s}_{ij}(t) & -s_{ij}(t) \cdot \dot{\dot{s}}_{ij}(t) \end{bmatrix} \leq 0, \quad (38) $$

where the first two elements of $N_i(t)$ are the tangency conditions, and the third element is augmented to the unconstrained Hamiltonian. The Hamiltonian is

$$ H_i = \frac{1}{2} ||u_i(t)||^2 + \lambda_i^p(t) \cdot v_i(t) + \lambda_i^f(t) \cdot u_i(t) $$

$$ \quad - \sum_{j \in \mathcal{N}_i} \mu_{ij}(t) \left( s_{ij}(t) \cdot \dot{s}_{ij}(t) + \dot{s}_{ij}(t) \cdot s_{ij}(t) \right), \quad (39) $$

where $\lambda_i^p(t)$ and $\lambda_i^f(t)$ are the position and velocity co-vectors, and $\mu_{ij}(t)$ is an inequality Lagrange multiplier with values

$$ \mu_{ij}(t) = \begin{cases} \geq 0 & \text{if } s_{ij}(t) \cdot \dot{s}_{ij}(t) + \dot{s}_{ij}(t) \cdot s_{ij}(t) = 0, \\ 0 & \text{if } s_{ij}(t) \cdot \dot{s}_{ij}(t) + \dot{s}_{ij}(t) \cdot s_{ij}(t) > 0. \end{cases} \quad (40) $$

To solve (39) for agent $i \in \mathcal{A}$, we consider two cases:

1. all agents $j \in \mathcal{N}_i(t)$ satisfy $\mu_{ij} = 0$,
2. any agent $j \in \mathcal{N}_i(t)$ satisfies $\mu_{ij} > 0$,

and optimally piece the constrained and unconstrained arcs together to arrive at a piecewise-continuous, energy-optimal trajectory.

Solving the unconstrained form of Problem 2 results in (10) - (12).

For the case where the trajectories of two or more agents overlap, we must piece together the unconstrained and constrained trajectories. This is achieved by using the agent interaction dynamics (Definition 3) to order the agents, such that lower priority agents steer to avoid the higher priority ones. Under this framework, we can prove that the premise of Theorem 1 is always satisfied. First, Lemma 2 shows that an unconstrained trajectory must never increase the energy required to reach a goal.

**Lemma 2.** For any agent $i \in \mathcal{A}$, following the unconstrained trajectory, the energy cost (14) required to reach a fixed goal $g \in \mathcal{F}$ is not increasing.

**Proof.** We may write the derivative of (14) along an unconstrained trajectory as

$$ \frac{dE_i^u(t)}{dt} = \lim_{\delta \to 0} \frac{1}{\delta} \left( \int_{t^f}^{t^f} ||u_i(\tau)||^2 \, d\tau - \int_{t^f}^{t^f} ||u_i(\tau)||^2 \, d\tau \right) $$

$$ = -\lim_{\delta \to 0} \frac{1}{\delta} \int_{t^f}^{t^f} ||u_i(\tau)||^2 \, d\tau, \quad (41) $$

which is never positive. Therefore, (14) is never increasing. \qed
Next, we introduce Lemma 3, which proves the premise of Theorem 1 is always satisfied with the trajectories generated by our framework.

**Lemma 3.** For each agent $i \in \mathcal{A}$, there always exists some interval of time $[t, t']$ with $t < t'$ such that agent $i$ follows an unconstrained trajectory for all $t \in [t, t']$.

**Proof.** The case when any agent $i \in \mathcal{A}$ is moving with an unconstrained trajectory, is covered by Lemma 2, so we focus on the case when any of the safety constraints is active.

Let $\mathcal{K} \subseteq \mathcal{A}$ be a group of agents which all have their safety constraint active over some interval $t \in [t_1, t_2]$. By Definition 3, there exists some $i \in \mathcal{K}$ such that $\Gamma^c_i(t) = 1$ for all $j \in \mathcal{K}, j \neq i$. Therefore, agent $i$ satisfies Lemma 2 and always moves toward its assigned goal by Theorem 1.

Next, consider agent $j \in \mathcal{V} \setminus \{i\}$ such that $\Gamma^c_j(t) = 1$ for all $k \in \mathcal{K} \setminus \{i\}$. As agent $j$ may never be assigned to the same goal as $i$, there must exist some time $t_j < \min\{t', t'_j\}$ such that $|s_{ij}(t_j)| > 2R$ by (27). Thus, agent $j$ moves with an unconstrained trajectory for all $t \in [t, t']$. The above steps can be recursively applied until only a single unconstrained agent remains, which follows an unconstrained trajectory for some finite time interval. This satisfies the premise of Theorem 1. \qed

In what follows, we present the solution to the constrained case and the optimal time to transition between the two cases.

### 4.1. Constrained Solution

As with the assignment problem, the constrained solution is presented in terms of some agent $i \in \mathcal{A}$. However, the steps presented here are performed simultaneously by all agents. First, we define the conflict set

**Definition 8.** We define the *conflict set* for agent $i \in \mathcal{A}$ at time $t \in [t_0, t'_i]$ as

$$\mathcal{V}_i(t) = \left\{ j \in \mathcal{N}(t) \mid \mu_{ij}(t) > 0, \Gamma^c_{ij} = 0 \right\}.$$  

i.e., the set of all agents which $i$ may collide with and have a higher priority than agent $i$.

Agent $i$ must then steer to avoid all agents $j \in \mathcal{V}_i(t)$. To solve for the constrained trajectory of agent $i$ we use the optimally and Euler-Lagrange conditions,

$$\frac{\partial H_i}{\partial u_i} = 0,$$

$$-\lambda_i = \frac{\partial H_i}{\partial \dot{x}_i}.$$  

Application of (43) to (39) yields

$$u_i(t) = -\lambda'_i(t) - \sum_{j \in \mathcal{V}_i(t)} \mu_{ij}(t) s_{ij}(t),$$

while (44) results in

$$-\lambda''_i(t) = \sum_{j \in \mathcal{V}_i(t)} \mu_{ij}(t) \dot{s}_{ij}(t),$$

$$-\lambda'_i(t) = \lambda''_i(t) + \sum_{j \in \mathcal{V}_i(t)} \mu_{ij}(t) \dot{s}_{ij}(t).$$

In general, agent $i \in \mathcal{A}$ must generate a trajectory which satisfies (45) - (47). From Assumption 2, we consider the case when $|V_i(t)| = 1$ for agents $i \in \mathcal{A}, j \in \mathcal{V}_i(t)$ over some interval $t \in [t_1, t_2] \subset [t_0, t'_i)$.

The optimality condition and Euler-Lagrange equations become

$$u_i(t) = -\lambda'_i(t) - \mu_{ij}(t) s_{ij}(t),$$

$$-\lambda''_i(t) = \mu_{ij}(t) \dot{s}_{ij}(t),$$

$$-\lambda'_i(t) = \lambda''_i(t) + \mu_{ij}(t) \dot{s}_{ij}(t).$$

We denote the relative speed between two agents $i$ and $j$ as

$$a_{ij}(t) = \|\dot{s}_{ij}(t)\|.$$  

Next, we define a new orthonormal basis for $\mathbb{R}^2$.

**Definition 9.** For an agent $i \in \mathcal{A}$ satisfying $|\mathcal{V}(t)| = 1$, over some nonzero interval $t \in [t_1, t_2]$, where $a_{ij}(t) \neq 0$, we define the contact basis as

$$\hat{p}_{ij}(t) = \frac{s_{ij}(t)}{|s_{ij}(t)|},$$

$$\hat{q}_{ij}(t) = \frac{s_{ij}(t)}{|s_{ij}(t)|}.$$  

where $\hat{p}_{ij}(t) \cdot \hat{q}_{ij}(t) = 0$ by (38), and both vectors are unit length. Thus, (52) and (53) constitute an orthonormal basis for $\mathbb{R}^2$.
Next, we find the projection of $\ddot{s}_{ij}(t)$ onto the new contact basis. From (38) we have

$$\ddot{s}_{ij}(t) \cdot \dot{p}_{ij}(t) = \ddot{s}_{ij}(t) \cdot s_{ij}(t) = \frac{-a_{ij}^2(t)}{2R}. \quad (54)$$

We apply integration by parts to find the $\dot{q}_{ij}(t)$ component of $\ddot{s}_{ij}(t)$. First,

$$\int \ddot{s}_{ij}(t) \cdot \dot{s}_{ij}(t) \, dt = \ddot{s}_{ij}(t) \cdot \dot{s}_{ij}(t) - \int \dddot{s}_{ij}(t) \cdot \dot{s}_{ij}(t) \, dt,$$

which implies

$$\int \dddot{s}_{ij}(t) \cdot \dot{s}_{ij}(t) \, dt = \frac{1}{2} \ddot{s}_{ij}(t) \cdot \dot{s}_{ij}(t) = \frac{1}{2} a_{ij}^2(t). \quad (56)$$

Taking a time derivative of (56) yields

$$\dddot{s}_{ij}(t) \cdot \dot{s}_{ij}(t) = a_{ij}(t) \dot{a}_{ij}(t). \quad (57)$$

Next we show that $a_{ij}(t) \neq 0$ over any constrained trajectory, and therefore (52) and (53) hold always over a constrained arc.

**Theorem 2.** For any agent $i \in \mathcal{A}$ such that $|V_i(t)| = 1$ and $V_j(t) = \emptyset$ for $j \in \mathcal{V}_i(t)$, over some interval $t \in [t_1, t_2]$, if the relative speed between agents (51) is ever zero, then agent $i$ must be moving with an unconstrained trajectory.

**Proof.** Subtracting (48) from $u_{ij}(t)$, $j \in \mathcal{V}_i(t)$ yields

$$\ddot{s}_{ij}(t) = u_{ij}(t) + \dot{X}_i(t) + \mu_{ij}(t) \dot{s}_{ij}(t). \quad (58)$$

Multiplying (58) with $s_{ij}(t)$ and from (38), yields

$$\ddot{s}_{ij}(t) \cdot s_{ij}(t) = -a_{ij}^2(t) = \dot{u}_{ij}(t) \cdot s_{ij}(t) + \mu_{ij}(t) \dot{s}_{ij}(t) \cdot s_{ij}(t). \quad (59)$$

Solving (59) for $\mu_{ij}(t)$ we have

$$\mu_{ij}(t) = \frac{u_{ij}(t) \cdot s_{ij}(t)}{4R^2} - \frac{\dot{X}_i(t) \cdot s_{ij}(t)}{4R^2} - \frac{a_{ij}^2(t)}{4R^2}. \quad (60)$$

Multiplying (48) with $s_{ij}(t)$ yields

$$u_{ij}(t) \cdot s_{ij}(t) = -\dot{X}_i(t) \cdot s_{ij}(t) - 4R^2 \mu_{ij}(t). \quad (61)$$

Setting (60) equal to (61), we have

$$\ddot{s}_{ij}(t) \cdot \dot{s}_{ij}(t) = a_{ij}(t) \ddot{s}_{ij}(t) \dot{s}_{ij}(t) - \dot{X}_i(t) \cdot s_{ij}(t) - a_{ij}^2(t) = u_{ij}(t) \cdot s_{ij}(t) - \dot{X}_i(t) \cdot s_{ij}(t). \quad (62)$$

Thus, if $a_{ij}(t) = 0$, it implies $u_{ij}(t) \cdot s_{ij}(t) = u_{ij}(t) \cdot s_{ij}$.

Next, we write (57) as

$$\dddot{s}_{ij}(t) \cdot \dot{s}_{ij}(t) = (u_{ij}(t) - u_{ij}) \cdot \dot{s}_{ij}(t) = a_{ij}(t) \dot{a}_{ij}(t). \quad (63)$$

Therefore, $a_{ij}(t) = 0$ implies $u_{ij}(t) \cdot s_{ij}(t) = u_{ij}(t) \cdot s_{ij}(t)$. Then $u_{ij}(t) = u_{ij}(t)$ in $\mathbb{R}^2$. As $\mathcal{V}_i(t) = \emptyset$ and $u_{ij}(t) = u_{ij}(t)$, it must be true that agent $i$ is following an unconstrained trajectory. Therefore, we may consider any segment where $a_{ij}(t) = 0$ to be an unconstrained arc and $\mu_{ij}(t) = 0$. \qed

Finally, we can use (54) and (57) to project $\ddot{s}_{ij}(t)$ onto the contact basis,

$$\ddot{s}_{ij}(t) = \left[ \frac{a_{ij}^2(t)}{2R} \hat{p}_{ij}(t) \right] \dddot{s}_{ij}(t) \dot{s}_{ij}(t). \quad (64)$$

which we use to solve for the time derivatives of (52) and (53). First,

$$\frac{d}{dt} \ddot{p}_{ij}(t) = \ddot{s}_{ij}(t) \cdot \dot{s}_{ij}(t) = \frac{a(t)}{2R} \dddot{q}_{ij}(t). \quad (65)$$

Then, by the quotient rule,

$$\frac{d}{dt} \ddot{q}_{ij}(t) = \frac{\dddot{s}_{ij}(t) \cdot \dddot{q}_{ij}(t) - \dddot{s}_{ij}(t) \dddot{q}_{ij}(t)}{a_{ij}^2(t)} = \frac{\dddot{s}_{ij}(t) \dddot{q}_{ij}(t) - \dddot{s}_{ij}(t) \dddot{q}_{ij}(t)}{a_{ij}^2(t)} \dddot{q}_{ij}(t) \dot{s}_{ij}(t) = \frac{a(t)}{2R} \dddot{q}_{ij}(t). \quad (66)$$

From (7), we may now write $\dddot{s}_{ij}(t)$ projected on to the contact basis (Definition 9) as

$$\dddot{s}_{ij}(t) = \left( u_{ij}(t) + \dot{X}_i(t) + \mu(t) \dot{s}_{ij}(t) \right) \left[ \begin{array}{c} \ddot{p}_{ij}(t) \\ \ddot{q}_{ij}(t) \end{array} \right] + \mu(t) \left[ \begin{array}{c} 2R \\ 0 \end{array} \right]. \quad (67)$$
Next, we set (64) equal to (67) and rewrite it as a system of scalar equations,

\[ \lambda_i'(t) \cdot \hat{p}_{ij}(t) = \frac{-a_i^2(t)}{2R} - 2R \mu_{ij}(t) - u_j(t) \cdot \hat{p}_{ij}(t), \quad (68) \]

\[ \lambda_i'(t) \cdot \hat{q}_{ij}(t) = \hat{a}_{ij}(t) - u_j(t) \cdot \hat{q}_{ij}(t). \quad (69) \]

Taking the time derivative of (68) yields

\[ \frac{a_{ij}(t)}{2R} \lambda_i'(t) \cdot \hat{q}_{ij}(t) + \lambda_i'(t) \cdot \hat{p}_{ij}(t) = -\frac{a_{ij}(t)}{2R} \hat{a}_{ij}(t) \]

\[ -2R \hat{a}_{ij}(t) - u_j(t) \cdot \hat{p}_{ij}(t) - \frac{a_{ij}(t)}{2R} u_j(t) \cdot \hat{q}_{ij}(t). \quad (70) \]

We then substitute (50) and (69) into (70), which yields

\[ \lambda_i'(t) \cdot \hat{p}_{ij}(t) = \frac{3a_{ij}(t) \hat{a}_{ij}(t)}{2R} + 2R \mu_{ij}(t) + u_j(t) \cdot \hat{p}_{ij}(t). \quad (71) \]

Taking a time derivative of (69),

\[ \frac{-a_{ij}(t)}{2R} \lambda_i'(t) \cdot \hat{p}_{ij}(t) + \lambda_i'(t) \cdot \hat{q}_{ij}(t) = \hat{a}_{ij}(t) \]

\[ -u_j(t) \cdot \hat{q}_{ij}(t) + \frac{a_{ij}(t)}{2R} u_j(t) \cdot \hat{p}_{ij}(t), \quad (72) \]

and substituting (50) and (68) into (72), yields

\[ \lambda_i'(t) \cdot \hat{q}_{ij}(t) = u_j(t) \cdot \hat{q}_{ij}(t) - \hat{a}_{ij}(t) + \frac{a_{ij}(t)}{2R} u_j(t) \cdot \hat{p}_{ij}(t). \quad (73) \]

Note that as agent \( j \) is moving in the unconstrained path, \( u_j(t) \) is a constant. We then take an additional time derivative of (71) and (73). This yields

\[ \lambda_i''(t) \cdot \hat{p}_{ij}(t) = -\frac{a_{ij}(t)}{2R} \lambda_i'(t) \cdot \hat{q}_{ij}(t) \]

\[ + \frac{3}{2R} (\hat{a}_{ij}^2(t) + a_{ij}(t) \hat{a}_{ij}(t)) \]

\[ - 2R \hat{a}_{ij}(t) - u_j(t) \cdot \hat{p}_{ij}(t), \quad (74) \]

\[ \lambda_i''(t) \cdot \hat{q}_{ij}(t) = \frac{a_{ij}(t)}{2R} \lambda_i'(t) \cdot \hat{p}_{ij}(t) - \frac{a_{ij}(t)}{2R} u_j(t) \cdot \hat{p}_{ij}(t) \]

\[ - \hat{a}_{ij}(t) + \frac{3}{4R} a_{ij}^2(t) \hat{a}_{ij}(t). \quad (75) \]

Substituting (49), (71), and (73) into (74) and (75) yields a system of nonlinear ordinary differential equations,

\[ 2R \hat{a}_{ij}(t) + \frac{a_{ij}^2(t)}{8R^3} = \frac{a_{ij}^2(t)}{2R} \mu_{ij}(t) \]

\[ + \frac{3}{2R} \hat{a}_{ij}(t) + \frac{a_{ij}(t) \hat{a}_{ij}(t)}{2R}, \quad (76) \]

\[ \hat{a}_{ij}(t) + \hat{a}_{ij}(t) \mu_{ij}(t) = a_{ij}(t) \hat{a}_{ij}(t) + \frac{3}{2R} a_{ij}^2(t) \hat{a}_{ij}(t). \quad (77) \]

Thus, for any constrained trajectory to be energy-optimal it must be a solution of (76) and (77) while also satisfying the boundary conditions (34) and (35). In general, this is difficult as both equations are nonlinear, and (77) is third order.

Our approach is to impose \( \hat{a}_{ij}(t) = 0, a_{ij}(t) > 0 \). This reduces (76) and (77) to

\[ 2R \hat{a}_{ij}(t) + \mu_{ij}(t) \hat{a}_{ij}(t) = \frac{a_{ij}^4(t)}{8R^3}, \]

\[ a_{ij} \hat{a}_{ij}(t) = 0, \quad (78) \]

which has the unique solution

\[ \mu_{ij}(t) = \frac{a_{ij}^2(t)}{2R}. \quad (80) \]

We may then write (49) as

\[ \lambda_i''(t) = \frac{a_{ij}^4(t)}{8R^3} \hat{p}_{ij}(t), \quad (81) \]

which we then integrate, resulting in

\[ \lambda_i'(t) = \frac{a_{ij}^3}{4R^2} \hat{q}_{ij}(t) + c_1, \quad (82) \]

Substituting (80) and (82) into (50) yields

\[ \lambda_i'(t) = \frac{a_{ij}^3}{4R^2} \hat{q}_{ij}(t) + c_1 + \frac{a_{ij}^3}{4R^2} \hat{q}_{ij}(t) = -c_1, \quad (83) \]

thus,

\[ \lambda_i'(t) = c_1 \cdot t + c_2. \quad (84) \]

We may solve for the constants of integration, \( c_1 \) and \( c_2 \), in terms of \( a_{ij} \) and \( u_j(t) \) by writing (58) on the contact basis

\[ \begin{bmatrix} -\frac{a_{ij}^4}{2R} \\ 0 \end{bmatrix} = u_j(t) - c_1 \cdot t - c_2 + \frac{1}{2R} \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (85) \]
Additionally, as $a_{ij}$ is constant, the rotation of the contact basis relative to the global frame is linear. Thus, the remaining unknown quantities are: the transition time, $t_1$, the exit time $t_2$, the initial orientation of the local frame, $\hat{p}_i(t_1)$, $\tilde{q}_i(t_1)$, and the orbital speed $a_{ij}$. These quantities are coupled to the proceeding and following unconstrained arcs by the jump condition; see Bryson and Ho (1975),

$$\mathbf{x}_i(t_1) = \mathbf{x}_i(t_1^+)$$  \hspace{1cm} (86)

$$\lambda_i(\mathbf{x}_i(t_1)) = \lambda_i(\mathbf{x}_i(t_1^+)) + \mathbf{v}_i^T \frac{\partial N_i(t)}{\partial \mathbf{x}(t)}|_{t=t_1}$$  \hspace{1cm} (87)

$$H(t_1^+) = H(t_1^-)$$  \hspace{1cm} (88)

$$H(t_2^+) = H(t_2^-)$$  \hspace{1cm} (89)

$$\mathbf{x}_i(t_2^-) = \mathbf{x}_i(t_2^-)$$  \hspace{1cm} (90)

$$\lambda_i(t_2^-) = \lambda_i(t_2^-)$$  \hspace{1cm} (91)

where $t_k^-$ and $t_k^+$ correspond to the time just before and after the transition at $t_k$ for $k \in \mathbb{N}$, and the constant vector $v_i$ is given by; see Bryson and Ho (1975)

$$v_i = \left[ -\frac{u_i(t_i^-) + u_i(t_i^+)}{2k_i(t_i^-)} , \frac{u_i(t_i^-)}{2k_i(t_i^-)} \right]$$  \hspace{1cm} (92)

For the case when agent $i$’s trajectory has only a single constrained arc, (86) - (91) coupled with the initial and final conditions, (35) and (34), constitute 26 scalar equations to solve for the 26 unknowns (8+8+8 constants of integration + 2 transition times). When additional constrained arcs become active, additional jump conditions must be computed using (86) - (91). The entire system of equations is then solved simultaneously to yield the energy-optimal trajectory for agent $i$.

5. Simulation Results

To provide an insight to the behavior of the agents, a series of simulations were performed with $M = N = 10$ agents and a time parameter of $T = 10$ s. The simulations were run for $t = 20$ s or until all agents reach their assigned goal, whichever occurred later. The center of the formation moved with a velocity of $\mathbf{v}_{cg} = [0.15, 0.35]$ m/s; the leftmost and rightmost three goals each move with an additional periodic velocity of $[0.125 \cos 0.75t, 0]$ m/s relative to the formation.

The minimum separating distance between agents, total energy consumed, and maximum velocity for the unconstrained solutions to Problem 2 are all given as a function of the horizon in Table 1. The energy consumption only considers the energy required to reach the goal, which, in this case, was significantly lower than the energy required to maintain the formation. The trajectory of each agent over time is given in Figures 1–3 for varying sensing horizon values. Although the trajectories may appear to cross in Figures 1–3, they are only crossing in space and not in time.

The results in Table 1 generally show no correlation between energy consumption and sensing horizon. The minimum energy consumption occurs at the centralized case, and tends to remain low for very large horizon values. However, energy consumption at lower sensing horizons are entirely problem dependent, and can not be predicted a priori, as demonstrated by the $h = 0.75$ m and $h = 0.95$ m cases.

6. Conclusion

We have proposed a decentralized framework for moving a group of autonomous agents into a desired formation. The only information required a priori is the posi-
Table 1: Numerical results for N=10 agents and M=10 goals for various sensing distances.

| $h$ [m] | min. separation [cm] | energy [J/kg] | $t_f$ [s] | Total Bans |
|---------|---------------------|--------------|-----------|------------|
| inf     | 25.25               | 0.85         | 20        | 0          |
| 1.60    | 1.64                | 1.10         | 20        | 4          |
| 1.50    | 1.60                | 1.17         | 20        | 24         |
| 1.40    | 2.01                | 1.96         | 23.3      | 31         |
| 1.30    | 0.33                | 1670         | 26.05     | 36         |
| 1.20    | 0.65                | 866          | 25.35     | 34         |
| 1.10    | 1.05                | 5370         | 26.85     | 40         |
| 1.00    | 1.96                | 7609         | 30.65     | 35         |
| 0.95    | 3.12                | 3149         | 25.05     | 27         |
| 0.75    | 1.37                | 6.87         | 20        | 35         |
| 0.50    | 0.27                | 692.0        | 26.65     | 35         |

Figure 2: Simulation result for $h = 1.30$ m. The agents do not start with a globally unique assignment, and several agents must re-route partway through the simulation. Although the trajectories cross in space they do not cross in time.

Future areas of research include relaxing the need for permanent banning to increase optimality of the assignment and guarantee satisfaction of Lemma 3. Furthermore, exploration of (76) and (77) to find superior locally-optimal solutions, and efficient solutions to (45) - (47) in the case that multiple agents are in contact. An analysis of the optimality for the interaction dynamics and impact of noise and uncertainty in the system would be another area for future research.

References

M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo. Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013. ISSN 19353812. doi: 10.1007/s11721-012-0075-2.

A. E. J. Bryson and Y.-C. Ho. *Applied Optimal Control: Optimization, Estimation, and Control*. John Wiley and Sons, 1975.

B. Chalaki and A. A. Malikopoulos. An Optimal Coordination Framework for Connected and Automated Vehicles in two Interconnected Intersections. In *2019 IEEE Conference on Control Technology and Applications*, Hong Kong, 2 2019. URL http://arxiv.org/abs/1903.00120.

A. Dave and A. A. Malikopoulos. The Prescription Approach to Decentralized Stochastic Control with Word-of-Mouth Communication. In *ArXiv:1907.12125*, 7 2019. URL http://arxiv.org/abs/1907.12125.

J. Guo, Z. Lin, M. Cao, and G. Yan. Adaptive control schemes for mobile robot formations with triangularised structures. *IET Control Theory & Applications*, 4(9):1817–1827, 2010. ISSN 1751-8644. doi: 10.1049/iet-cta.2009.0513. URL http://digital-library.theiet.org/content/journals/10.1049/iet-cta.2009.0513.

Y. Hanada, G. Lee, and N. Y. Chong. Adaptive Flocking of a Swarm of Robots Based on Lo-
Figure 3: Simulation result for $h = 0.75$ m. Although the horizon for this case is smaller than in Figure 2, the system dynamics happen to result in more efficient trajectories overall.

G. Lee and Y. C. Nak. Self-configurable mobile robot swarms with hole repair capability. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2008. ISBN 9781424420582. doi: 10.1109/IROS.2008.4650696.

Z. Lin, M. Broucke, and B. Francis. Local control strategies for groups of mobile autonomous agents. *IEEE Transactions on Automatic Control*, 49(4):622–629, 2004. ISSN 00189286. doi: 10.1109/TAC.2004.825639.

H. Ma, D. Harabor, P. J. Stuckey, J. Li, and S. Koenig. Searching with Consistent Prioritization for Multi-Agent Path Finding. In *33rd AAAI Conference on Artificial Intelligence*, pages 7643–7650, 2019. URL www.aaai.org.

A. A. Malikopoulos. Equilibrium Control Policies for Markov Chains. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 7093–7098, 2011.

A. A. Malikopoulos. Centralized stochastic optimal control of complex systems. In *Proceedings of the 2015 European Control Conference*, pages 721–726, 2015.

A. A. Malikopoulos. A duality framework for stochastic optimal control of complex systems. *IEEE Transactions on Automatic Control*, 61(10):2756–2765, 2016.

A. A. Malikopoulos, V. Maroulas, and J. Xiong. A multi-objective optimization framework for stochastic control of complex systems. In *Proceedings of the 2015 American Control Conference*, pages 4263–4268, 2015.

A. A. Malikopoulos, C. G. Cassandras, and Y. J. Zhang. A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections. *Automatica*, 93(April):244–256, 2018. ISSN 00051098. doi: 10.1016/j.automatica.2018.03.056.

D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh. Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming. *International Journal of Robotics Research*, 35(10):1261–1285, 2016. ISSN 17413176. doi: 10.1177/0278364916632065.

H. Oh, A. R. Shirazi, C. Sun, and Y. Jin. Bio-inspired self-organising multi-robot pattern formation: A review. *Robotics and Autonomous Systems*, 91:83–100, 2017. doi: 10.1016/j.robot.2016.12.006.

K.-K. Oh, M.-C. Park, and H.-S. Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015. doi: 10.1016/j.automatica.2014.10.022.

R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007. ISSN 00189219. doi: 10.1109/JPROC.2006.887293.

R. Rajasree. Optimal Formation Control of Unmanned Aerial Vehicles with Obstacle Avoidance. In *International Conference on Control, Instrumentation, Communication and Computational Technologies*, 2010.
C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. Computer Graphics, 21(4):25–34, 1987. ISSN 0097-8930. doi: 10.1145/37402.37406.

M. Rubenstein, C. Ahler, and R. Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, 2012. ISBN 9781467314039. doi: 10.1109/ICRA.2012.6224638.

G. Sartoretti, J. Kerr, Y. Shi, G. Wagner, T. K. S. Kumar, S. Koenig, and H. Choset. PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning. IEEE Robotics and Automation Letters, 4(3): 2378–2385, 2018. URL http://arxiv.org/abs/1809.03531.

Y. Song and J. M. O’Kane. Forming repeating patterns of mobile robots: A provably correct decentralized algorithm. In IEEE International Conference on Intelligent Robots and Systems, volume 2016-Novem, pages 5737–5744, 2016. ISBN 9781509037629. doi: 10.1109/IROS.2016.7759844.

X. Sun and C. G. Cassandras. Optimal Dynamic Formation Control of Multi-Agent Systems in Environments with Obstacles. In IEEE Conference on Decision and Control, pages 2359–2364, 2015.

J. O. Swartling, I. Shames, K. H. Johansson, and D. V. Dimarogonas. Collective Circumnavigation. Unmanned Systems, 02(03):219–229, 2014. ISSN 2301-3850. doi: 10.1142/S2301385014400019. URL http://www.worldscientific.com/doi/abs/10.1142/S2301385014400019.

A. M. Turing. The chemical basis of morphogenesis. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, 237(641):37–72, 1952. doi: 10.1007/BF02459572.

M. Turpin, N. Michael, and V. Kumar. Concurrent Assignment and Planning of Trajectories for Large Teams of Interchangeable Robots. In 2013 IEEE International Conference Robotics and Automation (ICRA), pages 842–848, Karlsruhe, Germany, 2013a. ISBN 9781467356435.

M. Turpin, K. Mohta, N. Michael, and V. Kumar. Goal Assignment and Trajectory Planning for Large Teams of Aerial Robots. Proceedings of Robotics: Science and Systems, 37:401–415, 2013b. doi: 10.1109/ICRA.2013.6630671.

M. Turpin, N. Michael, and V. Kumar. CAPT: Concurrent assignment and planning of trajectories for multiple robots. International Journal of Robotics Research, 33(1):98–112, 2014. ISSN 02783649. doi: 10.1177/0278364913515307.

G. Vásárhelyi, C. Virág, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek. Optimized flocking of autonomous drones in confined environments. Technical report, 2018. URL http://robotics.sciencemag.org/.

J. Wang and M. Xin. Integrated Optimal Formation Control of Multiple Unmanned Aerial Vehicles. IEEE Transactions on Control Systems Technology, 21(5): 1731–1744, 2013. doi: 10.1109/TCST.2012.2218815.

W. Wu, S. Bhattacharya, and A. Prorok. Multi-Robot Path Deconfliction through Prioritization by Path Prospects. Technical report, 8 2019. URL http://arxiv.org/abs/1908.02361.

H. Xu and R. G. C. Carrillo. Distributed Near Optimal Flocking Control for Multiple Unmanned Aircraft Systems. In International Conference on Unmanned Aerial Vehicles, pages 879–885, Denver, CO, USA, 2015. ICUAS.