Optimal leader allocation in UAV formation pairs under no-cost switching

Dean Richert Jorge Cortés

Abstract— **This paper considers a group of UAVs that travel from origin to destination locations. Individual agents can choose to either fly directly to their destination or pair with other agents into a leader-follower formation to conserve fuel. In the latter case, only the follower experiences a cost benefit, and hence UAVs must negotiate how to fairly allocate the task of leading. We show that selfish agents cannot reach satisfactory collaboration agreements, which leads us to propose the notion of ϵ-cooperativeness. For this class of UAVs, we introduce the PARTITION REFINEMENT ALGORITHM to strategically schedule alternating leading and following intervals that induce cooperation. We show that the proposed strategy is guaranteed to find leader allocations with the minimum number of leader-follower switches. Moreover, these allocations are optimal with regards to the cost that UAVs can attain while collaborating with other UAVs. Several simulations illustrate our results.**

I. INTRODUCTION

We consider a group of robots where agent-to-agent interactions are not necessary, but may be beneficial. Each robot has an individual task and is only concerned with accomplishing its task as efficiently as possible. Further, there is a nominal cost to an agent when it performs the task solo and a lower cost of performing the task when another robot assists the agent. We suppose that the cost benefit is only awarded to the agent that is being assisted. Although there is no immediate motivation for one robot to assist another, if the assisting robot has assurance that the other agent will reciprocate in the future, it may indeed be mutually beneficial for both agents to cooperate with each other. The degree to which agents believe the other will reciprocate certainly affects the amount that one agent is willing to help.

Inspired by the scenario described above, we aim to solve the following problem: given two agents and a bid from one, what is the cost that the other agent can expect to incur if they were to collaborate? Moreover, how can the task of helping be properly distributed between the agents to ensure they cooperate? Our prime motivation comes from UAV formation pairs, where flying in the wake of another UAV reduces aerodynamic drag and improves fuel economy. In this setting, leading the formation is costly to an agent and following is beneficial. A closely related work [1] considers an algorithm for determining which UAVs should join in formation given origin and destination locations but does not address the allocation of the leading task. Other scenarios where cooperation-inducing task allocation plays an important role include power scheduling for wireless networks [2], distributed data processing [3], autonomous resource transportation, surveillance, and foraging [4].

The authors are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA 92093, USA, {drichert,cortes}@ucsd.edu

Literature review. Task allocation as a means to improve individual and global utilities has been studied extensively [5], [6], [7]. In particular, [8] motivates the case for flight formation in UAVs. Recent advances in aircraft control has made autonomous formation flight viable in real-life scenarios [9], [10]. However, few works consider the behavior of agents or the task execution once it has been allocated. For example, [11] studies the emergence of coalitions between selfish agents, but assumes that once the coalition is formed the agents act in favor of the common good. If the agents cannot be trusted to execute their task, [12] shows that auction algorithms can be compromised in the presence of "cheaters". For a particular class of agents, [13] shows that despite unbalanced relationships, reciprocal altruism can emerge as a dominated strategy. The fact that the behavior of agents greatly affects how or whether a task is carried out has motivated game theorists to offer several notions of player behavior and study how these affect the outcome [14], [15], [16]. In this study, the particular notion of equilibrium and how it relates to the agent behavior plays a key role. An appropriate choice of equilibrium in one case can cause significant inefficiencies in another, as illustrated in [17]. The works [18], [19] discuss ways to remedy these inefficiencies.

Statement of contributions. We start by showing that neither selfish nor fully cooperative models for agent behavior are adequate. Selfish robots cannot agree on leader allocations. Cooperative robots do not realize the goal of performing their task as efficiently as possible. This leads us to introduce a behavioral model that we term ϵ-cooperative, which ensures collaboration between robots occurs while being consistent with each robot's objective. Next, under this model, we develop the PARTITION REFINEMENT ALGORITHM which strategically schedules intervals of leading and following for each UAV so that both agents cooperate. The proposed algorithm generates the minimum number of leader to follower switches. Finally, we find the optimal cost that an agent can attain while collaborating with another robot under the assumption that switching the lead does not incur a fuel cost. The PARTITION REFINEMENT ALGORITHM is able to schedule leader allocations so that UAVs are guaranteed to attain their optimal cost. Several simulations illustrate our results. For reasons of space, all proofs are omitted and will appear elsewhere.

Notation. For $a, b \in \mathbb{R}^d$, let $d(a, b) = \|a - b\|$ denote the Euclidean distance between a and b. The closed segment between $a, b \in \mathbb{R}^d$ is denoted $[a, b]$ and the ray starting at a in the direction of $b-a$ is $\mathrm{ray}(a, b-a)$. Given the Cartesian product of $X \subset \mathbb{R}^{d_1}$, $Y \subset \mathbb{R}^{d_2}$, let $\pi_1 : X \times Y \rightrightarrows X$ and $\pi_2 : X \times Y \rightrightarrows Y$ denote the projections onto X and Y, respectively. This definition can be analogously extended to Cartesian products of multiple spaces. The set of even, odd, and natural numbers are denoted $\mathbb{E}, \mathbb{O}$, and $\mathbb{N}$ respectively. The indicator function of $A \subset Y$, denoted $\mathbf{1}_A : Y \to \{0,1\}$, is given by $\mathbf{1}_A(y) = 1$ if $y \in A$, and $\mathbf{1}_A(y) = 0$ otherwise. The positive part of $x \in \mathbb{R}$ is denoted by $x_+ = \max\{0, x\}$. Finally, the cardinality of a set A is given by $|A|$.

II. Problem statement

Consider a pair of UAVs, each with unique identifiers (UIDs) i and j evolving in $X \subset \mathbb{R}^3$. We assume that i and j have synchronized clocks and can communicate with each other. A superscript i denotes a quantity associated with agent i. Thus, the position of UAV i is denoted by x^i. Each agent has an origin, a target location, and an objective which is to arrive at its target location while consuming the least amount of fuel. A UAV's fuel consumption can be reduced by flying in formation in the wake of another UAV (e.g. reducing aerodynamic drag during flight). The energy consumption per unit distance of following in formation is $\gamma \in \mathbb{R}_{>0}$, whereas flying solo or leading in a formation incurs a fuel consumption of $\Gamma \in \mathbb{R}_{>0}$. By assumption, $\Gamma > \gamma$.

A. Formations and partitions

A *formation* is a pair $(x_r, u_h) \in X^2$, where x_r is a rendezvous location and the unit vector u_h is the heading direction of the formation. When we say that i and j are *in formation*, we mean that they originated at x_r, the distance between them is effectively zero and they are flying in the direction of u_h.

The execution of a formation (x_r, u_h) is completely described by a *partition* and the UID of the UAV which leads the formation first (without loss of generality, from here on UAV i leads the formation first). A partition is a finite tuple $P = (x_1, x_2, \dots, x_n) \in \mathrm{ray}(x_r, u_h)$ satisfying,

$$x_r = x_1, \quad d(x_p, x_{p+1}) \geq \delta, \quad x_p \in (x_{p-1}, x_{p+1}),$$

for $2 \leq p < n-1$. The components of P denote when the two UAVs in formation initiate a swap in the lead (except for $x_1 = x_r$, the rendezvous location, and x_n, the location where the UAVs break formation and fly directly to their targets). The parameter $\delta \geq 0$ is called the *distance to switch* and represents the distance required for i and j to switch from leading to following (resp. following to leading). During a switch, both UAVs consume $\Gamma_\delta \geq \Gamma$ fuel per unit distance. As such, the cost of switching the lead is $\delta\Gamma_\delta$ to both i and j. For the sake of our model, we use $\delta = \delta\Gamma_\delta = 0$ to represent instantaneous switching with no cost.

We use $\mathscr{P}(x_r, u_h)$ to refer to the set of all partitions of a formation. When clear from the context, we simply use $\mathscr{P}$. The *trivial partition* is $P = (x_r)$, where the UAVs fly directly to their target locations from x_r. For brevity, the function $x_b : \mathscr{P} \to \mathrm{ray}(x_r, u_h)$ returns the breakaway location of the UAVs given a partition. That is, $x_b(P) = \pi_{|P|}(P)$.

Figure 1 shows the execution of a formation given a partition as well as the process of UAVs transitioning from following to leading (resp. leading to following).

We conclude this subsection with the definition of two functions that will be used subsequently in this paper. Given a partition and position $x \in \mathrm{ray}(x_r, u_h)$, let $\mathrm{LS} : \mathscr{P} \times \mathrm{ray}(x_r, u_h) \to \mathrm{ray}(x_r, u_h)$ return the location when the last switch was initiated before x Precisely,

$$\mathrm{LS}(P, x) = \sum_{p=1}^{|P|-1} x_p \mathbf{1}_{[x_p, x_{p+1})}(x).$$

Also, let $\mathcal{N}(P, x) = \big|P \cap (x_r, x]\big|$ be the number of switches that have been initiated up until x.

B. Objective function

At any point $x \in [x_r, x_b(P)]$, a UAV can compute the cost of the flight from x_r to its target if the UAV were to leave the formation at position x. For i, we represent this cost as $J^i_{\mathrm{eff}} : \mathscr{P} \times \mathrm{ray}(x_r, u_h) \to \mathbb{R}_{>0}$. An analogous function J^j_{eff} exists for j. The function J^i_{eff} can be expressed as,

$$\begin{aligned} J^i_{\mathrm{eff}}(P, x) = {} & \Gamma\, \mathrm{DL}^i(P, x) + \Gamma_\delta\, \mathrm{DS}(P, x) \\ & + \gamma\, \mathrm{DF}^i(P, x) + \Gamma d(x, \bar{x}^i), \end{aligned} \tag{1}$$

where $\mathrm{DL}^i(P, x)$ is the distance that i has led the formation up until x, $\mathrm{DF}^i(P, x)$ is the distance that i has followed in the formation up until x, $\mathrm{DS}(P, x)$ is the distance that i has traveled while switching from leader to follower (and vice versa) up until x, and $\Gamma d(x, \bar{x}^i)$ is the fuel required to fly directly to the target $\bar{x}^i$ should i decide to break the formation at x. Then, for $P = (x_r, x_2, x_3, \dots)$,

$$\begin{aligned} \mathrm{DL}^i(P, x) = {} & \min\big\{d(x_r, x), d(x_r, x_2)\big\} \\ & + \sum_{\substack{p=3 \\ p \text{ odd}}}^{\mathcal{N}(P,x)} \big(d(x_p, x_{p+1}) - \delta\big) \\ & + \Big(d\big(\mathrm{LS}(P, x), x\big) - \delta\Big)_+ \mathbf{1}_{\mathbb{E}\setminus\{0\}}\big(\mathcal{N}(P, x)\big). \end{aligned} \tag{2}$$

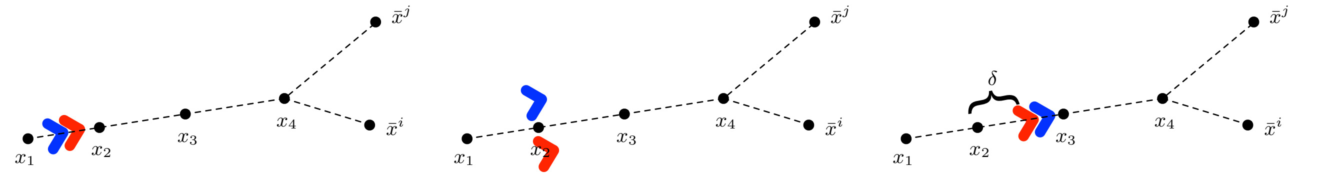
The first term accounts for the case when $x \in [x_r, x_2]$, the second term is the distance led by i up until $\mathrm{LS}(P, x)$, and the last term is any residual distance led by i after $\mathrm{LS}(P, x)$. The distance traveled due to switching the lead is,

$$\mathrm{DS}(P, x) = \delta\mathcal{N}(P, x) - \Big(\delta - d\big(LS(P, x), x\big)\Big)_+,$$

where the first term is the distance associated with all switches that have been initiated and the last term makes up for the case that i and j are in the process of switching at x. Finally, we use the relation,

$$d(x_r, x) = \mathrm{DF}^i(P, x) + \mathrm{DS}(P, x) + \mathrm{DL}^i(P, x), \tag{3}$$

to define DF^i. Although an expression like (2) can be invoked to define DF^i, we use (3) to emphasize the fact that in a formation, a UAV is either leading the formation, following in the formation, or in the process of switching.



(a) Red begins leading the formation (b) A leader switch is initiated at x_2 (c) δ distance past x_2, blue leads the formation

Fig. 1. Example flight behavior of UAVs given a partition $P = (x_1, x_2, x_3, x_4)$. The dashed lines represent the proposed flight paths. Upon arrival at the rendezvous location, the red UAV leads the formation. At x_2, the two UAVs deviate slightly from their current heading while the red UAV decreases its speed and the blue UAV increases its speed. The new speeds are maintained for a distance δ, after which the UAVs return to the original heading and speed of the formation. Upon completion of the maneuver, the red UAV is now following in the formation and the blue UAV is leading it. There will be one more leader switch initiated at x_3 before the formation is over at x_4. Beyond x_4, the UAVs fly directly to their respective targets.

C. Problem statement

Suppose that UAV i has notified j that it wishes to consume $F^i \in \mathbb{R}_{>0}$ fuel on its flight from x_r to $\bar{x}^i$. Then, j would like to solve the following program,

$$\underset{P \in \mathscr{P}}{\text{minimize}} \quad J^j_{\text{eff}}(P, x_b(P)) \tag{4a}$$
$$\text{subject to} \quad F^i = J^i_{\text{eff}}(P, x_b(P)). \tag{4b}$$

Implicit in (4) is the assumption that j can trust i to lead the formation when a partition dictates. In Section III, we discuss behavioral models of UAVs that can be used to determine whether or not UAVs will abide by a partition. Under this model, section IV proposes an algorithm for creating partitions that ensure UAVs will cooperate. The results of these sections are combined in Section V to solve (4).

III. Behavior of UAVs in a formation

We now discuss an appropriate model for the behavior of UAVs which can be used to analyze whether or not UAVs will abide by a partition. As motivation, consider the behavioral models we call *cooperative* and *selfish*.

Definition III.1 (cooperative and selfish agents). *Consider any formation (x_r, u_h) between i and j, partitioned by any $P \in \mathscr{P}(x_r, u_h)$. UAV i is cooperative if it will abide by P. UAV i is selfish if it will abide by P if and only if P satisfies*

$$J^i_{\text{eff}}(P, x_b(P)) \le J^i_{\text{eff}}(P, x), \quad \forall x \in [x_r, x_b(P)].$$

A network of cooperative UAVs is not an appropriate model for UAVs who have an objective of consuming the least amount of fuel. We can see this from the following example.

Example III.2 (cooperative agents). Given a partition $P = (x_r, x_2, x_3)$ of the formation (x_r, u_h) between i and j,

$$\begin{aligned} J^j_{\text{eff}}(P, x_3) &= \gamma d(x_r, x_2) + \delta \Gamma_\delta \\ &\quad + \Gamma(d(x_2, x_3) - \delta) + \Gamma d(x_3, \bar{x}^j) \\ &> \gamma d(x_r, x_2) + \Gamma d(x_2, \bar{x}^j) = J^j_{\text{eff}}(P, x_2). \end{aligned}$$

where we have used the triangle inequality. Thus, j would do better (potentially, much better) to break the formation at x_2. As a result, a UAV being cooperative is at odds with the objective of consuming the least amount of fuel. •

The other end of the spectrum is no better as we show next.

Lemma III.3 (all UAVs cannot be selfish). *Consider a formation (x_r, u_h) between two selfish UAVs. If $\{\bar{x}^i, \bar{x}^j\} \cap \operatorname{ray}(x_r, u_h) = \emptyset$, the only partition the two UAVs will abide by is the trivial one, $P = (x_r)$.*

The reader may recognize similarities between the above result and the repeated Prisoner's Dilemma game in which the Nash equilibrium is to defect at every stage [15]. Such an example is a testament to the inefficiency of the Nash equilibrium in some cases. The discussion above motivates the introduction of the concept of *ϵ-cooperative*.

Definition III.4 (ϵ-cooperative) *Consider any formation (x_r, u_h) between i and j, partitioned according to any $P \in \mathscr{P}(x_r, u_h)$. UAV i is ϵ-cooperative for some $\epsilon \ge 0$ if it will abide by P if and only if P satisfies,*

$$J^i_{\text{eff}}(P, x_b(P)) \le J^i_{\text{eff}}(P, x) + \epsilon, \quad \forall x \in [x_r, x_b(P)].$$

In other words, an ϵ-cooperative UAV will abide by a partition so long as there is no point in the formation where its effective cost is ϵ less than the final effective cost it will get upon completion of the formation (using this model, one can model selfish behavior as 0-cooperative and cooperative behavior as ∞-cooperative). Note that marginal cost pricing and Pigouvian taxes [15] achieve the same desired result as ϵ-cooperative does. However, we prefer this formulation because it places an emphasis on the UAV behavior rather than a "tax" applied as a negative externality which has a less meaningful interpretation in the context of this problem. We refer to the set of partitions that both UAVs will abide by as *cooperation-inducing*. Formally,

$$\begin{aligned} \mathscr{P}_c(\epsilon^i, \epsilon^j) = \{P \in \mathscr{P} : {} & J^i_{\text{eff}}(P, x_b(P)) \le J^i_{\text{eff}}(P, x) + \epsilon^i, \\ & J^j_{\text{eff}}(P, x_b(P)) \le J^j_{\text{eff}}(P, x) + \epsilon^j, \\ & \text{for all } x \in [x_r, x_b(P)]\}. \end{aligned}$$

In words, if i and j are ϵ^i-cooperative and ϵ^j-cooperative respectively, then both i and j will abide by $P \in \mathscr{P}_c(\epsilon^i, \epsilon^j)$. When clear from the context, we simply write $\mathscr{P}_c$. The problem we want to solve can be reformulated as

$$\underset{P \in \mathscr{P}_c(\epsilon^i, \epsilon^j)}{\text{minimize}} \quad J^j_{\text{eff}}(P, x_b(P)) \tag{5a}$$
$$\text{subject to} \quad F^i = J^i_{\text{eff}}(P, x_b(P)) \tag{5b}$$

where we restrict $P \in \mathscr{P}_c \subset \mathscr{P}$.

IV. BUILDING COOPERATION-INDUCING PARTITIONS

As an intermediate step in solving the optimization problem (5), here we provide a strategy termed the PARTITION REFINEMENT ALGORITHM to construct partitions that induce cooperation, $P \in \mathscr{P}_c(\epsilon^i, \epsilon^j)$, and also satisfy,

$$J^i_{\text{eff}}(P, x_b(P)) = F^i, \tag{6a}$$
$$J^j_{\text{eff}}(P, x_b(P)) = F^j, \tag{6b}$$

when given input data $\mathcal{D} = (x_r, u_h, \bar{x}^i, \bar{x}^j, \epsilon^i, \epsilon^j, F^i, F^j)$. Once equipped with this strategy, we show how solving (5) boils down to finding its optimal value. We refer to data $\mathcal{D}$ for which a partition in $\mathscr{P}_c$ exists satisfying (6) *admissible*. The set of all admissible data is denoted by $\mathscr{D}_a$. It is interesting to note that partitions in $\mathscr{P}_c$ satisfying (6) might specify different numbers of leader switches. Even for the same number of leader switches, different partitions might exist because of the flexibility of the constraints that define $\mathscr{P}_c$.

Section IV-A introduces a procedure to create partitions, not necessarily in $\mathscr{P}_c$, that satisfy (6). Then, Section IV-B builds on this procedure to propose the PARTITION REFINEMENT ALGORITHM that finds a partition in $\mathscr{P}_c$. Finally, Section IV-C analyzes the properties of our strategy.

A. Creating partitions with known costs

First, we discuss a method for creating partitions in $\mathscr{P}$ satisfying (6). The method we employ is captured by the function $\mathrm{CP} : \mathscr{D}_a \times \mathbb{N} \to \mathscr{P}^2$ which uses admissible data and the desired number of leader switches to return two partitions in $\mathscr{P}$ satisfying (6) (the reason we are interested in two will be made clear later).

Our reasoning is as follows. Before trying to construct the desired partition, we need to determine the total leading distances of each UAV along the formation. Once this is synthesized, we build partitions that correspond to them. Let us then begin by reformulating the effective cost functions in terms of the distances led by each one of the UAVs. Let $f^i, f^j : \mathbb{R}^2_{>0} \to \mathbb{R}$ be defined by,

$$\begin{aligned} f^i(\mu, \nu : \mathcal{D}, N) &= \Gamma\mu + (N-2)\delta\Gamma_\delta + \gamma\nu \\ &\quad + \Gamma d\big(x_r + (\mu + (N-2)\delta + \nu)u_h, \bar{x}^i\big), \\ f^j(\mu, \nu : \mathcal{D}, N) &= \gamma\mu + (N-2)\delta\Gamma_\delta + \Gamma\nu \\ &\quad + \Gamma d\big(x_r + (\mu + (N-2)\delta + \nu)u_h, \bar{x}^j\big). \end{aligned}$$

The above definitions are motivated by the following fact: If a partition P satisfies,

$$\mathrm{DL}^i(P, x_b(P)) = \mu, \qquad \mathrm{DL}^j(P, x_b(P)) = \nu,$$

and $\mathrm{DS}(P, x_b(P)) = N\delta$, then f^i corresponds to $J^i_{\text{eff}}(P, x_b(P))$. Thus, to ensure (6), we are interested in the solutions to the system of equations,

$$f^i(\mu, \nu : \mathcal{D}, N) = F^i, \tag{7a}$$
$$f^j(\mu, \nu : \mathcal{D}, N) = F^j. \tag{7b}$$

Lemma IV.1 (two possible breakaway locations). *Given $\mathcal{D} \in \mathscr{D}_a$ and $N \in \mathbb{N}$, there exist two and only two solutions to* (7).

It is now clear why CP must create two partitions; a partition corresponding to each solution of (7). Denote the two solutions to (7) as $(\mu_1(\mathcal{D}, N), \nu_1(\mathcal{D}, N))$ and $(\mu_2(\mathcal{D}, N), \nu_2(\mathcal{D}, N))$. Here, we choose one method to create the desired partitions. Let $(x_1, \dots, x_N) = \mathrm{CP}_1(\mathcal{D}, N) \subset \mathrm{ray}(x_r, u_h)$ be given by $x_1 = x_r$, and

$$\begin{aligned} x_2 &= x_1 + \mu_1(\mathcal{D}, N)u_h, \\ x_3 &= x_2 + (\nu_1(\mathcal{D}, N) + \delta)u_h, \end{aligned}$$

and $d(x_n, x_{n+1}) = \delta$ for $3 \leq n \leq N-1$. That is, we schedule i and j to lead their respective segments at the beginning, placing the remaining leader switches at the end of the formation. An analogous CP_2 is defined using $(\mu_2(\mathcal{D}, N), \nu_2(\mathcal{D}, N))$. Note that the solutions to (7) may not be different. In this case, $\mathrm{CP}_1(\mathcal{D}, N) = \mathrm{CP}_2(\mathcal{D}, N)$.

B. The PARTITION REFINEMENT ALGORITHM

Here, we design an algorithm to refine the initial partitions of CP, creating a partition which induces cooperation. We begin with a high-level description.

> *[Informal description]:* Find initial partitions with one leader switch using CP. If either partition induces cooperation, terminate. Otherwise, find two partitions with an additional leader switch using CP. While maintaining (6), strategically refine the elements of the partitions. If cooperation can be induced, the algorithm terminates. Otherwise, repeat the process of finding two partitions with an additional leader switch using CP.

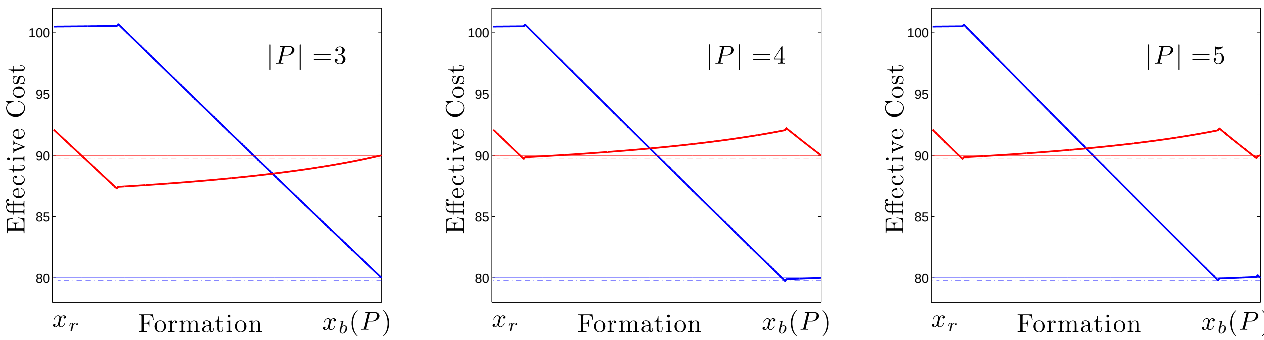
We refer to this strategy as the PARTITION REFINEMENT ALGORITHM and formally describe it as Algorithm 1.

Algorithm 1: PARTITION REFINEMENT ALGORITHM

input : $\mathcal{D} \in \mathscr{D}_a$
output: P

```
1  N := 3
2  repeat
3  |  foreach m ∈ {1,2} do
4  |  |  P := (x_1,...,x_N) = CP_m(D,N) %initialize
5  |  |  for n = 1,...,N−3 do % refine elements
6  |  |  |  x_{n+2} := x_{n+3} − (L_max(P,n) + δ(1−n)_+) u_h
7  |  |  |  x_{n+1} := x_n + (L_max(P,n) + δ(1−n)_+) u_h
8  |  |  |  if P ∈ P_c then terminate
9  |  |  end
10 |  end
11 |  N := N + 1                    % try additional switch
12 end
```

In what follows we discuss the algorithm's execution and motivate its design. Because the initializations $\mathrm{CP}_1, \mathrm{CP}_2$ ensure that (6) is satisfied, the trick is to move the elements of the partitions while maintaining (6). To this end, we propose moving the elements of the initial partition in pairs. That is, if we move x_{n+1} for a certain distance, we make sure to move x_{n+2} the same distance (see lines 6-7). Thus,



(a) Red UAV will not cooperate.

(b) Blue UAV will not cooperate.

(c) Both UAVs will cooperate.

Fig. 2. Partitions obtained during the execution of the PARTITION REFINEMENT ALGORITHM as the number of leader switches increase. Input to the algorithm is $\mathcal{D} = ((0,0),(1,0),(100,10),(90,-20),0.2,0.3,80,90)$ with parameters $\Gamma = 1, \gamma = 0.5, \Gamma_\delta = 1.7, \delta = 0.2$. The thin blue solid line is $F^i = J^i_{\text{eff}}(P, x_b(P))$ and the thin dotted blue line is $F^i - \epsilon^i$. A UAV's effective cost transitions from increasing to decreasing at leader switch locations. A close observation at the switch locations shows the cost associated with switching the lead. If a UAV's cost drops below its dotted line, it will not cooperate. For $|P| = 5$, the last switch occurs just before the breakaway location, which is all that is needed to ensure cooperation compared to $|P| = 4$.

the distances led/followed/switched remain constant for both UAVs. Our algorithm starts at the rendezvous location x_r and computes the *maximum lead distance* of UAVs, given by $L_{\max} : \mathscr{P} \times \mathbb{N} \to \mathbb{R}$. This function is described in general as follows. Suppose i is scheduled to lead on a segment $[x_n + \delta u_h, x_{n+1}]$. Then $L_{\max}(P, n)$ is the maximum distance that i could lead without j's effective cost falling below the threshold $F^j - \epsilon^j$. Defined implicitly, for $n \geq 3$ odd,

$$\begin{aligned} F^j - \epsilon^j &= J^j_{\text{eff}}(P, x_n + \delta u_h) - \Gamma d(x_n + \delta u_h, \bar{x}^j) \\ &\quad + \gamma L_{\max}(P, n) + \Gamma d(x_n + (L_{\max}(P, n) + \delta) u_h, \bar{x}^j). \end{aligned}$$

An analogous expression for $L_{\max}$ exists when $n \geq 2$ is even by replacing superscripts j with i. Finally, for $n = 1$,

$$F^j - \epsilon^j = \gamma L_{\max}(P, 1) + \Gamma d(x_r + L_{\max}(P, 1) u_h, \bar{x}^j).$$

We would like to move x_{n+1} to $x_n + (L_{\max}(P, n) + \delta) u_h$ while maintaining (6). Thus we also move x_{n+2} to preserve the original distance between x_{n+1} and x_{n+2} (lines 6-7). The process continues by computing these maximum lead distances for subsequent switch points (line 5). If at any time the partition is in $\mathscr{P}_c$, the algorithm terminates (line 8). Otherwise, we increase the number of leader switches in the partition and repeat (line 11). The terminating criteria for PARTITION REFINEMENT ALGORITHM is $P \in \mathscr{P}_c$.

Remark IV.2 (Execution of the PARTITION REFINEMENT ALGORITHM under no-cost switching). For instantaneous leader switching ($\delta = \delta \Gamma_\delta = 0$), there is a noteworthy simplification to the PARTITION REFINEMENT ALGORITHM. In this case, the solutions to (7) are the same every time CP_1, CP_2 are invoked (i.e., the breakaway locations remain constant). Thus, each computation of $L_{\max}$ (lines 6-7) up until the second last switch are a repetition of past maximum lead distances computations. Saving these values for later use makes the "loop" of lines 5-9 a single calculation. •

Figure 2 shows simulation results of the algorithm in use. Increasing the number of switches from 1 to 3 allows for a partition that induces cooperation to be found.

C. Algorithm analysis

Here we analyze the properties of the PARTITION REFINEMENT ALGORITHM regarding termination and computation rounds. We use *round* to refer to every time the algorithm executes line 8 (we use this line as an indicator flag, not because it is of particular computational complexity).

Proposition IV.3 (Algorithm terminates given admissible data). *For input $\mathcal{D}$, the* PARTITION REFINEMENT ALGORITHM *terminates with $P \in \mathscr{P}_c$ satisfying* (6) *iff $\mathcal{D} \in \mathscr{D}_a$. Moreover, $|P| = \#_s(\mathcal{D})$, where $\#_s(\mathcal{D})$ is the minimum number of switches specified by any partition in $\mathscr{P}_c$ satisfying* (6). *The number of rounds upon termination of the* PARTITION REFINEMENT ALGORITHM *is less than or equal to,*

$$\begin{cases} \big(\#_s(\mathcal{D}) - 2\big)\big(\#_s(\mathcal{D}) - 1\big) & \text{if } \delta > 0, \\ 2(\#_s(\mathcal{D}) - 2) & \text{if } \delta = 0. \end{cases}$$

The above results are relevant both from a practical and a computational viewpoint. In particular, the resulting partition will ensure that UAVs do not switch the lead more than strictly necessary. Moreover, the computation of the partition requires a number of steps that is proportional its cardinality squared. Of key importance, the algorithm's termination property given admissible data reduces the problem of solving (5) to finding its optimal value, which is addressed next.

V. OPTIMAL PARTITIONS

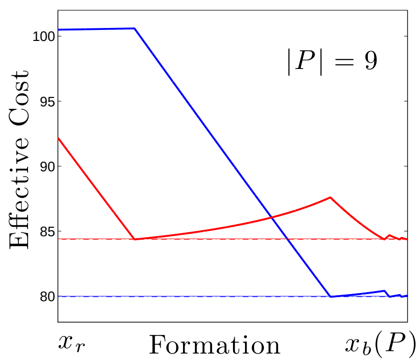
The last step towards fully solving (5) is determining its optimal value. We use P_* to denote a solution of (5) with optimal value $F^j_* = J^j_{\text{eff}}(P_*, x_b(P_*))$. As a candidate lower bound on F^j_*, we look to (4) for inspiration. To this end, the following result is particularly powerful.

Proposition V.1 (Optimal value under no-cost switching). *Given $(x_r, u_h, \bar{x}^i, \bar{x}^j, \epsilon^i, \epsilon^j, F^i)$, let U_* (resp. F^j_*) be the optimal value of* (4) *(resp.* (5)*). If $\max\{\epsilon^i, \epsilon^j\} > 0$ and $\delta = 0$ then $F^j_* = U_*$. Otherwise, $F^j_* \geq U_*$.*

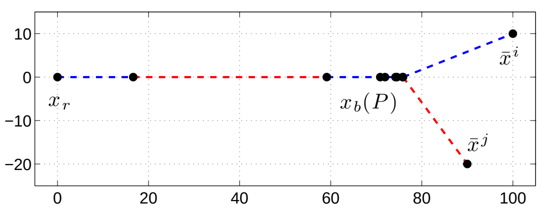
For the no-cost switching case, it remains to specify how to solve (4). The following result reveals that the optimal value of (4) is obtained from the solution of a convex program.

Lemma V.2 *Given* $(x_r, u_h, \bar{x}^i, \bar{x}^j, \epsilon^i, \epsilon^j, F^i)$*, let* U_* *be the optimal value of* (4)*. Then* U_* *is also the optimal value of the following convex program,*

$$\underset{\Delta \in \mathbb{R}_{\geq 0}}{\text{minimize}} \quad (\Gamma+\gamma)\Delta + \delta\Gamma_\delta + \Gamma d(x_r + (\Delta+\delta)u_h, \bar{x}^i) + \Gamma d(x_r + (\Delta+\delta)u_h, \bar{x}^j) - F^i.$$



(a) Partition attaining red's optimal cost.



(b) Resulting flight paths of UAVs under the optimal partition.

Fig. 3. An optimal partition solving (5). The input and parameters to the PARTITION REFINEMENT ALGORITHM are the same as in Figure 2 except $\delta = 0$, $\epsilon^i = \epsilon^j = 0.05$ and $F^j = F^j_*$. The small values of ϵ^i, ϵ^j show the ability of our algorithm to induce cooperation even when the UAVs are nearly selfish. With small ϵ^i and ϵ^j, many leader switches are needed to induce cooperation, with higher frequency near $x_b(P)$.

Combining Proposition V.1 and Lemma V.2, Figure 3 shows simulation results of the PARTITION REFINEMENT ALGORITHM creating a solution to (5) under no-cost switching.

VI. CONCLUSIONS AND FUTURE WORK

We have studied the problem of optimally allocating the leader role of UAV pairs flying in formation in order to guarantee a fuel benefit. Each UAV could choose to fly solo or in formation with another UAV. In formation, different costs are associated with "leading", "following", or switching between these two states. We introduced the notion of ϵ-cooperative UAVs and showed that it results in the successful collaboration of agents. We designed the PARTITION REFINEMENT ALGORITHM to schedule alternating leading intervals for each UAV in the formation. The proposed strategy is guaranteed to find an allocation of the leading task with the minimum number of leader-follower switches. Additionally, the resulting allocation is optimal with regards to flying cost when switching the lead does not incur a fuel cost. Future work will include exploring the optimality of our strategy in the cost-of-switching case and its extension to higher-dimensional formations involving more than two agents. We view the PARTITION REFINEMENT ALGORITHM as a building block for more complex scenarios where a network of agents bargain with different offers about the possibility of jointly performing a task. We plan to explore the convergence of the resulting bargaining dynamics as agents try to determine which other agent is the best to collaborate with.

ACKNOWLEDGMENTS

This research was supported by NSF award CMMI-0908508.

REFERENCES

[1] G. Ribichini and E. Frazzoli, "Efficient coordination of multiple-aircraft systems," in *IEEE Conf. on Decision and Control*, (Maui, Hawaii), pp. 1035–1040, 2003.

[2] M. Nokleby, A. L. Swindlehurst, Y. Rong, and Y. Hua, "Cooperative power scheduling for wireless MIMO networks," in *IEEE Global Telecommunications Conference*, (Washington, D.C.), pp. 2982–2986, nov 2007.

[3] W. W. Chu, L. J. Holloway, L. Min-Tsung, and K. Efe, "Task allocation in distributed data processing," *IEEE Transactions on Computers*, vol. 13, pp. 57–69, November 1980.

[4] M. A. Batalin and G. S. Sukhatme, "Coverage, exploration and deployment by a mobile robot and communication network," *Telecommunication Systems Journal*, vol. 26, no. 2, pp. 181–196, 2004. Special Issue on Wireless Sensor Networks.

[5] B. P. Gerkey and M. J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," *International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[6] P. R. Ma, E. Y. S. Lee, and M. Tsuchiya, "A task allocation model for distributed computing," *IEEE Transactions on Computers*, vol. 31, pp. 41–47, January 1982.

[7] E. H. Ostergaard, M. J. Mataric, and G. S. Sukhatme, "Distributed multi-robot task allocation for emergency handling," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, (Maui, HI), pp. 821–826, October 2001.

[8] D. Hummel, "Aerodynamic aspects of formation flight in birds," *Journal of Theoretical Biology*, vol. 104, pp. 321–347, October 1983.

[9] F. Giulietti, L. Pollini, and M. Innocenti, "Autonomous formation flight," *IEEE Control Systems Magazine*, vol. 20, pp. 34–44, December 2000.

[10] F. Borrelli, T. Keviczky, and G. J. Balas, "Collision-free UAV formation flight using decentralized optimization and invariant sets," in *IEEE Conf. on Decision and Control*, (Atlantis, Paradise Island, Bahamas), pp. 1099–1104, December 2004.

[11] O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation," *Artificial Intelligence*, vol. 101, pp. 165–200, 1998.

[12] M. H. Rothkopf, "Thirteen reasons why the Vickrey-Clarke-Groves process is not practical," *Operations Research*, vol. 55, pp. 191–197, March-April 2007.

[13] R. D. Morton, G. A. Bekey, and C. M. Clark, "Altruistic task allocation despite unbalanced relationships within multi-robot communities," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, (St. Louis, USA), pp. 5849–5854, October 2009.

[14] L. Samuelson, *Evolutionary Games and Equilibrium Selection*. MIT Press, 1997.

[15] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. Cambridge University Press, 2007.

[16] C. F. Camerer, *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, 2003.

[17] J. R. Correa, A. S. Schulz, and N. E. Stier-Moses, "On the inefficiency of equilibria in congestion games," in *Integer Programming and Combinatorial Optimization*, vol. 3509 of *Lecture Notes in Computer Science*, pp. 171–177, New York: Springer, 2005.

[18] W. J. Baumol, "On taxation and the control of externalities," *The American Economic Review*, vol. 62, pp. 307–322, June 1972.

[19] T. Groves, "Incentives in teams," *Econometrica*, vol. 41, pp. 617–631, July 1973.