

Distributed Optimization for Cooperative Agents: Application to Formation Flight*

Robin L. Raffard[†], Claire J. Tomlin[‡], and Stephen P. Boyd[§]

Abstract— We present a simple decentralized algorithm to solve optimization problems involving cooperative agents. Cooperative agents share a common objective and simultaneously pursue private goals. Furthermore, agents are constrained by limited communication capabilities. The algorithm is based on dual decomposition techniques and appears to be very intuitive. It solves the dual problem of an artificially decomposed version of the primal problem, replacing one large computationally intractable problem with many smaller tractable problems. It returns a feasible solution to the primal problem as well as an upper bound on the distance between this solution and the global optimum. Both convex and nonconvex examples are presented, the complexity of the convex case is analyzed, and the savings in complexity are demonstrated for both examples. Finally, by showing that there is no duality gap in these examples, optimality is certified.

I. INTRODUCTION

We consider optimization problems involving multiple cooperative agents, which are large in dimension and are in general NP-hard. Examples include trajectory optimization for formation control of vehicles [11], [18], [19], [9], [20], [10], decentralized control of power systems [17], distributed control of large scale systems [7], [5], [12], and problems of distributed optimization on sensor networks [14]. In most of these instances, the treatment is control oriented: the goal is to prove connectivity and stability of the system under decentralized control. In this paper, the approach is different but complementary. We pose the problem of distributing the decision-making between agents while optimizing objectives, both local (private) to each agent and global (common) for the system.

Two techniques are commonly used for dealing with optimization of large, nonlinear problems. The first is the *branch-and-bound* method, which consists of a search that maintains provable lower and upper bounds on the global solution. However, there is no guarantee that a solution will be returned in polynomial time, and an oracle for finding a ‘good’ lower bound might not be available. The second technique is the *descent method*, in which the objective is progressively optimized, while maintaining a feasible solution for the problem. This method results in guaranteed convergence to a local minimum; no information about the

global solution can be derived. Furthermore, this algorithm may not be implementable in a decentralized manner.

In this paper, we present a third approach which returns a feasible solution to the problem as well as a lower bound on the global optimal solution. It is particularly well suited to interconnected systems of multiple cooperative agents, in which communication passing may be restricted. It consists of an iterative algorithm, in which agents solve small problems involving their own variables as well as the variables of agents they communicate with. However, they do not need communicate with all other agents. The advantages of the algorithm are first, that it satisfies the physical constraints on information flow, and second, that it breaks the global problem into a sequence of smaller and hence more tractable problems. Whereas solving the global problem directly is prohibitively complex due to the high dimension of the system, distributing the computation of the smaller problems over each agent could be much more efficient.

Mathematically, the method is based on the dual decomposition technique. The method of dual decomposition has been used since the 1960’s, with the historical work of [8]. An excellent modern reference is the sixth chapter of [2]. When solving the dual problem, the dual function should be assumed to be a nondifferentiable concave function [2]. Therefore, nondifferentiable optimization tools have to be used [3], [1]. The most well known of these are the cutting plane method, the ellipsoid method, or more simply, the subgradient method [2], [16], [15]. While breaking the problem into a sequence of smaller problems, dual decomposition introduces a duality gap [6], which represents the difference between the primal and dual optimal costs. Here, we use a dualization scheme which maintains a small duality gap and which results in a very simple algorithm, easily implementable in a decentralized manner.

This paper is organized as follows. In Section 2, we present our algorithm and in Section 3, we show how it has been derived from dual decomposition techniques. In Section 4 and 5, we solve the formation flight problem in both convex and nonconvex cases and we prove that, even in the nonconvex example, the duality gap is zero.

II. DECENTRALIZED ALGORITHM FOR COOPERATIVE VEHICLE PROBLEMS

A. Problem Formulation

In this section we consider a multiple vehicle coordination problem, in which vehicles pursue private objectives as well as global objectives shared with other agents. For

*Research supported by NASA under Grant NCC 2-5422 and by the ONR MURI award “Computational Methods for Collaborative Motion”.

[†]Ph.D Student, Aeronautics and Astronautics, Stanford University. Corresponding author. rraffard@stanford.edu.

[‡]IEEE Member, Assistant Professor, Aeronautics and Astronautics, Courtesy Assistant Professor, Electrical Engineering.

[§]IEEE Fellow, Professor, Electrical Engineering.

each vehicle, we introduce $x_i \in \mathbf{R}^{n_i}$ and $u_i \in \mathbf{R}^{d_i}$, $i \in 1, \dots, N$, the state and the control input of the vehicle. The position of each vehicle is designated by $y_i = Cx_i$. For convenience, we also note $\mathbf{x}_i = (x_i, u_i)$. The dynamics of each vehicle is represented by nonlinear constraints on the state and the control input: $\mathbf{x}_i \in \mathcal{D}_i$.

For each vehicle, private objectives are specified by a linear equality on \mathbf{x}_i : $A_i \mathbf{x}_i = b_i$. In practice, private objectives range from fuel consumption minimization to target tracking to trajectory optimization along a specified path. Therefore, a target tracking objective could be represented by $Cx_i = y_{\text{target}}$, and a fuel minimization objective could be represented by $u_i = 0$.

Concerning the common objective, we consider the particular case of flight formation. This consists of a set of objectives that constrains pairs of aircraft. More precisely, common objectives are a set of linear equalities of the form: $y_i - y_j = \Delta y_{ij}^{\text{desired}}$. Vehicles are able to maintain formation while communicating with a limited number of other vehicles, as determined by the constraints set by the desired formation. For this purpose, we introduce a map $\mathcal{R} : i \in \{1, \dots, N\} \rightarrow \mathcal{R}(i) \subseteq \{1, \dots, N\}$ where $\mathcal{R}(i)$ represents the set of aircraft which communicate with aircraft i .

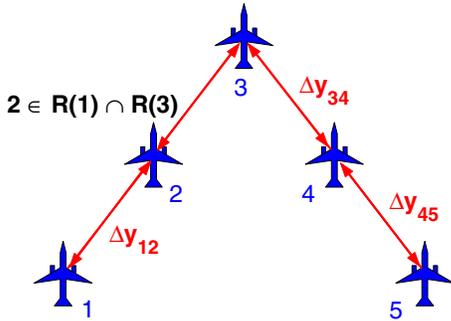


Fig. 1. V-formation of 5 aircraft. In this example, aircraft may only communicate with their closest neighbors and have no information about other aircraft. Without loss of generality, we may consider other, more general, communication patterns.

We introduce the quadratic deviation from these private and common objectives: $J_i = \|A_i \mathbf{x}_i - b_i\|_2^2$, $i = 1, \dots, N$ and $\|C(x_i - x_j) - \Delta y_{ij}\|_2^2$, $j \in \mathcal{R}(i)$, $i = 1, \dots, N$. The problem we are proposing to solve is to minimize these deviations under the constraints of the dynamics of the aircraft.

$$\begin{aligned}
 & \text{Minimize} && \sum_{i=1}^N \|A_i \mathbf{x}_i - b_i\|_2^2 + \\
 & && \sum_{i=1}^N \sum_{j \in \mathcal{R}(i)} \|C(x_i - x_j) - \Delta y_{ij}\|_2^2 & \text{(P1)} \\
 & \text{Subject to} && \mathbf{x}_i \in \mathcal{D}_i, \quad i = 1, \dots, N
 \end{aligned}$$

B. Decentralized Algorithm

Whereas most decentralized algorithms [10], [11] consist of computing the optimal trajectory for each vehicle by freezing the state of other agents, ours proceeds differently. We will first present an intuitive description of the algorithm; its mathematical justification and analysis will be presented in Section 3.

The key quantity for our algorithm is the deviation from the common objectives:

$$d_{ij} = C(x_i - x_j) - \Delta y_{ij}, \quad j \in \mathcal{R}(i), \quad i = 1, \dots, N. \quad (1)$$

First, all agents ignore the common objective. They compute their optimal trajectories, based on their private objectives only. They send these trajectories to the agents they share common objectives with, and likewise, receive the optimal trajectories from these agents. Next, each agent computes its deviation from the common objective (1), and then it recomputes its optimal trajectory as follows. Each agent sets up a constant force opposed to its deviation from the common objective, and then minimizes the sum of (a) the potential of this force, which tends to drive it back to the formation, and (b) the deviation from its own private objective. Once these trajectories have been computed, the agents calculate the new deviations from common objective (1) and update these deviations in the direction of the newly computed deviations. They iterate the procedure until the deviations converge.

Decentralized Algorithm

Each vehicle i sets its deviations from the common objective to 0. $d_{ij} = 0$, $j \in \mathcal{R}(i)$.

Repeat:

1. Each vehicle i sends (resp. receives) its d_{ij} (resp. d_{ji}) values to (resp. from) the aircraft it shares objectives with: $\{j | j \in \mathcal{R}(i)\}$.
2. Each vehicle i computes its optimal trajectory $x_i^* = \underset{x_i \in \mathcal{D}_i}{\text{argmin}} \{ \|A_i x_i - b_i\|_2^2 + \sum_{j \in \mathcal{R}(i)} 2(d_{ij} - d_{ji})^T C x_i \}$.
3. Each vehicle i sends (resp. receives) its (resp. their) optimal trajectory(ies) to (resp. from) vehicles $j \in \mathcal{R}(i)$.
4. Each vehicle i computes its new deviation (1) and updates: $d_{ij}^{\text{new}} := d_{ij} + \alpha_i (d_{ij}^{\text{new}} - d_{ij})$, where α_i is an update parameter.

Terminate when d_{ij} converge.

III. DERIVATION OF THE ALGORITHM USING DUAL SPLITTING METHOD

The decentralized algorithm presented above has been derived from the dual decomposition method. This allows us to prove that the algorithm returns a feasible solution to the primal problem and gives an upper bound on the distance between this solution and the global solution of

the problem.

The motivation for using dual decomposition was primarily that, starting from a large nonlinear problem involving variables of all agents, the method of dual decomposition splits the dual of this primal problem into a sequence of smaller problems; each one associated with an individual vehicle, only containing the variables accessible by that vehicle. The procedure can therefore be distributed over each agent, leading to a decentralized algorithm, computationally inexpensive and highly desirable from a practical viewpoint, especially in situations with no centralized authority and physical communication constraints.

A. Decoupling variables

The first step of the dual decomposition method is to construct a new primal problem equivalent to (P1), in which the objective has been decoupled, *i.e.* no variable appears in two different terms of the sum. For this purpose, we introduce slack variables $\tilde{x}_i \in \mathbf{R}^{n_i}$, $i = 1, \dots, N$, as follows:

$$\begin{aligned} & \text{Minimize} && \sum_{i=1}^N J_i + \sum_{i=1}^N \sum_{j \in \mathcal{R}(i)} \|C(\tilde{x}_i - \tilde{x}_{ij}) - \Delta y_{ij}\|_2^2 \\ & \text{Subject to} && \tilde{x}_i = x_i, \quad i = 1, \dots, N \\ & && \tilde{x}_{ij} = \tilde{x}_j, \quad j \in \mathcal{R}(i), \quad i = 1, \dots, N \\ & && \mathbf{x}_i \in \mathcal{D}_i, \quad i = 1, \dots, N \end{aligned} \quad (\text{P2})$$

The decomposition method then consists of solving an appropriate dual problem of (P2).

Note that there are many ways to formulate a dual problem. The choice depends on the constraints that should or should not be dualized. Ultimately, we would like a dual problem which may be decoupled, and which results in a small duality gap. In the next section, we will show that we should dualize all the constraints except the dynamics constraints.

B. Constraint dualization

The following rules are useful in order to know which constraints should be dualized.

Rule 1: Dualize the minimum number of constraints.

Writing (P2) in a more compact form:

$$\begin{aligned} & \text{Minimize} && f(x) + h(\tilde{x}) \\ & \text{Subject to} && x \in \mathcal{D} \\ & && s_i(x, \tilde{x}) = 0, \quad i = 1, \dots, p, \end{aligned} \quad (\text{P3})$$

denote $g_1(\mu_1, \dots, \mu_p)$ as the dual function associated with (P3) when p equality constraints are dualized, and $g_2(\mu_1, \dots, \mu_{p-1})$ when only $p-1$ are dualized. Thus (see Appendix), $\sup_{\mu_1, \dots, \mu_p} \{g_1\} \leq \sup_{\mu_1, \dots, \mu_{p-1}} \{g_2\}$, which means that the duality gap is smaller when dualizing fewer constraints.

Rule 2: Dualize linear equality constraints involving variables which have convex constraints and on which the objective function has a convex dependence.

Suppose $h(\tilde{x})$ can be written as $h_2(\tilde{x}, A\tilde{x})$ in which the function h_2 is convex and A is a $pN \times nN$ matrix. Consider the problem:

$$\begin{aligned} & \text{Minimize} && f(x) + h_2(\tilde{x}, A\tilde{x}) \\ & \text{Subject to} && x \in \mathcal{D} \\ & && \tilde{x} = x \\ & && \tilde{x} = A\tilde{x} \end{aligned} \quad (\text{P4})$$

Let us denote by $g_4(\mu, \nu)$ the dual function associated to (P4), where μ is the dual variable associated with the constraint $\tilde{x} = x$ and ν is that associated with $\tilde{x} = A\tilde{x}$. $g_1(\mu)$ is the dual function associated with (P3), and $s_1(x, \tilde{x}) = x - \tilde{x}$, $p = 1$. Thus, (see Appendix) $\sup_{\mu, \nu} \{g_4(\mu, \nu)\} = \sup_{\mu} \{g_1(\mu)\}$,

which means that constraints of the type $\tilde{x} = A\tilde{x}$, whose variables the objective has a convex dependence on, can be dualized without augmenting the duality gap. Therefore, in order to decouple the dual function and to keep the duality gap small, we dualize all the constraints of the form $\tilde{x}_i = x_i$ and $\tilde{x}_{ij} = \tilde{x}_j$, but not the dynamics constraints.

C. Lagrangian and Dual Function

Following the rules of the previous section, the Lagrangian, denoted L , of the optimization problem (P2) may be decoupled. Indeed, for $i = 1, \dots, N$, the i^{th} term of the sum is a function only of variables relative to aircraft i and to aircraft $\{j \mid j \in \mathcal{R}(i)\}$.

$$\begin{aligned} L(x, \tilde{x}, \mu, \nu) &= \sum_{i=1}^N \left(\underbrace{J_i - \mu_i^T x_i}_{L_{1i}(x_i, \mu_i)} + \right. \\ & \left. \underbrace{\mu_i^T \tilde{x}_i + \sum_{j \in \mathcal{R}(i)} (\|C(\tilde{x}_i - \tilde{x}_{ij}) - \Delta y_{ij}\|_2^2 + \nu_{ij}^T \tilde{x}_{ij}) - \sum_{j \in \mathcal{R}(i)} \nu_{ji}^T \tilde{x}_i}_{L_{2i}(\tilde{x}_i, \tilde{x}_{ij}, \mu_i, \nu_{ij}, \nu_{ji} \mid j \in \mathcal{R}(i))} \right) \end{aligned} \quad (2)$$

As a consequence, the dual function is also decoupled:

$$g(\mu, \nu) = \sum_{i=1}^N \left(\inf_{x_i \in \mathcal{D}_i} \{L_{1i}(x_i, \mu_i)\} + \inf_{\tilde{x}_i, \tilde{x}_{ij}} \{L_{2i}(\tilde{x}_i, \tilde{x}_{ij}, \mu_i, \nu_{ij}, \nu_{ji} \mid j \in \mathcal{R}(i))\} \right) \quad (3)$$

After some algebra, $g(\mu, \nu)$ can be computed as follows

$$g(\mu, \nu) = \sum_{i=1}^N \left(\inf_{(x_i, u_i) \in \mathcal{D}_i} \{L_{1i}(x_i, \mu_i)\} - \begin{cases} \sum_{j \in \mathcal{R}(i)} g_{2ij}(\nu_{ij}) & \text{if } \mu_i + \sum_{j \in \mathcal{R}(i)} (\nu_{ij} - \nu_{ji}) = 0 \\ & \text{and } D\nu_{ij} = 0, \quad \forall j \in \mathcal{R}(i) \\ \infty & \text{otherwise} \end{cases} \right), \quad (4)$$

in which $D = I - \begin{bmatrix} C \\ 0 \end{bmatrix}$, and where the convex function $g_{2ij}(\nu_{ij}) = \nu_{ij}^T \left(\frac{\nu_{ij}}{4} + C^T \Delta y_{ij} \right)$ is quadratic. Setting $\mu_i = \sum_{j \in \mathcal{R}(i)} (\nu_{ji} - \nu_{ij})$, for all $i = 1, \dots, N$, g can now

be considered as a function of the variable ν only, with the additional restriction that $D\nu_{ij} = 0$, $j \in \mathcal{R}(i)$, $i = 1, \dots, N$. Therefore, by noting $d_{ij} = C\nu_{ij}$, we can reduce the dimension of the dual space by introducing the function $\hat{g}(d) = g(\mu, \nu)$, where $\mu_i = \sum_{j \in \mathcal{R}(i)} (\nu_{ji} - \nu_{ij})$ and $D\nu_{ij} = 0$. We will use this function \hat{g} to maximize the dual function.

D. Dual function maximization

Finally, in order to solve for the dual problem, we need to maximize the dual function g , which is equivalent to maximizing \hat{g} . The maximization of \hat{g} is performed using a subgradient method. Denoting $\partial_{d_{ij}}(-\hat{g})$ as the set of subgradients of $-\hat{g}$ with respect to variables d_{ij} , we have

$$-Cx_i^* + Cx_j^* + \frac{d_{ij}}{2} + \Delta y_{ij} \in \partial_{d_{ij}}(-\hat{g}) \quad (5)$$

where

$$x_i^* = \operatorname{argmin}_{x_i \in \mathcal{D}_i} \left\{ J_i + \sum_{j \in \mathcal{R}(i)} (d_{ij}^T - d_{ji}^T) Cx_i \right\} \quad (6)$$

Therefore, maximizing the dual function is equivalent to the following iteration:

1. Compute x_i^* .
2. Update $d_{ij} := d_{ij} + \alpha_k (Cx_i^* - Cx_j^* - \Delta y_{ij} - \frac{d_{ij}}{2})$.

where k is the algorithm iteration index and $(\alpha_k)_{k \geq 0}$ is a square summable but not summable series.

We are now in a position to draw the link between this algorithm and the algorithm introduced in section 2. The dual variables d_{ij} have the units of position and are interpreted as the deviation from the formation. For example, at optimality, we have $d_{ij} = 2(Cx_i^* - Cx_j^* - \Delta y_{ij})$. During the algorithm, d_{ij} represent the iterative guesses of what the deviations from the formation should be. Given this guess, each vehicle computes its optimal trajectory as follows: It optimizes its private objective by taking into account the fact that deviating from the common objective tends to drive it back to the formation. Once it has computed its trajectory, it communicates this information to other vehicles, receives their trajectories and then computes the resulting deviations (1). It finally updates the guess for the deviation in the direction of the newly computed deviation.

IV. CONVEX CASE

We now investigate the particular case of the formation flying problem in which the problem (P1) is convex. We consider a linear dynamics for each of the N vehicles:

$$\dot{z}_i(t) = A_c z_i(t) + B_c u_i(t), \quad (7)$$

$$\text{where } A_c = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \lambda_d & 0 \\ 0 & 0 & 0 & \lambda_d \end{bmatrix}, \lambda_d \geq 0, \text{ and } B_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

which assumes that the net force applied to each aircraft

can be directly controlled. The dynamics can be exactly discretized:

$$x_i(t+1) = Ax_i(t) + Bu_i(t) \quad (8)$$

where $A = e^{A_c}$ and $B = \int_0^T e^{A_c \tau} B_c d\tau$. The local cost functions are chosen as follows:

$$J_i = \|Cx_i - y_{\text{target}i}\|_2^2 + \|u_i\|_2^2 \quad (9)$$

where Cx_i and $y_{\text{target}i}$ are respectively the position of aircraft i and the position of the target. The global cost function is chosen as before (cf. (P1)). Thus, the optimization problem is to minimize a positive definite quadratic form objective function over a linear set, and therefore it is a convex problem.

A. Results

At each iteration of the decentralized algorithm, optimal trajectories can be analytically derived by:

$$\begin{cases} u_i^* = (G^T C^T C G + I)^{-1} G^T C^T (C H x_i(0) - y_t + C \frac{\alpha_i}{2}) \\ x_i^* = G u_i^* + H x_i(0) \end{cases} \quad (10)$$

$$\text{where } G = \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ A^{T-1}B & A^{T-2}B & \dots & B \end{bmatrix}, H = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^T \end{bmatrix},$$

and $\alpha_i = \sum_{j \in \mathcal{R}(i)} (\nu_{ij}^T - \nu_{ji}^T)$

The scenario is chosen as a target tracking problem involving 5 followers required to fly in V-formation. The trajectories generated by both the decentralized algorithm and the centralized LQR are displayed in figure 2.

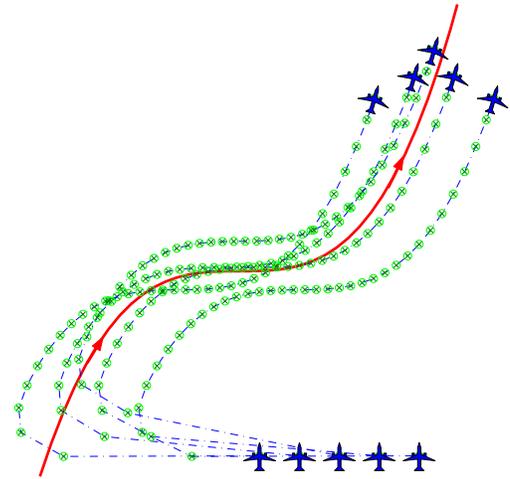


Fig. 2. Optimal trajectories for the target tracking problem. The solid curve corresponds to the trajectory of the target that the aircraft are following. The decentralized algorithm has performed 10 iterations. The centralized controller and the decentralized algorithm both give the same trajectories (represented by circles and crosses). However, the decentralized algorithm uses much less memory (25 times as less) and runs more quickly (1.2 times as fast).

B. Optimality

Since the initial problem (P1) is convex, (P2) is also convex and satisfies Slater's constraint qualification [4]. Therefore strong duality holds and the algorithm returns a primal optimal solution.

C. Convergence Analysis

The dual function is actually quadratic in ν and is therefore twice differentiable. Indeed, the dependence of x^* in ν is:

$$x_i^* = K\alpha_i = K\left(\sum_{j \in \mathcal{R}(i)} (\nu_{ij} - \nu_{ji})\right) \quad (11)$$

where $K = 2G(G^T C^T C G + I)^{-1} G^T C^T C$. Therefore, the gradient of g is affine in ν :

$$\frac{\partial g}{\partial \nu_{ij}} = K\left(\sum_{k \in \mathcal{R}(i)} \nu_{ik} - \sum_{k \in \mathcal{S}(i)} \nu_{ki} - \sum_{k \in \mathcal{R}(j)} \nu_{jk} + \sum_{k \in \mathcal{S}(j)} \nu_{kj}\right) - \frac{\nu_{ij}}{2} \quad (12)$$

And the Hessian is constant:

$$\frac{\partial^2 g}{\partial \nu_{ij} \partial \nu_{pl}} = (\delta_{ip} - \delta_{il} - \delta_{jp} + \delta_{jl})K - \frac{\delta_{ip} \delta_{jl}}{2} I \quad (13)$$

where δ_{km} is 1 when $k = m$ and 0 otherwise.

We can then derive a convergence analysis based on the condition number of this second derivative. For the gradient descent method, a bound on the number of iterations required for convergence can be considered as proportional to the condition number. Taking the example of the previous scenario, the Hessian is tridiagonal with diagonal terms equal to $-2K - \frac{I}{2}$ and tridiagonal terms equal to K .

We can easily show that its condition number does not grow as fast as n^2 , where n is the size of the matrix. This provides a bound on the complexity of the algorithm. As a function of the number of aircraft, the complexity of the algorithm cannot grow faster than N^2 . It actually turns out that this bound is rather crude. When simulated with realistic numerical values for the system, the complexity is found to grow like $\log(N)$ (Figure 3), which demonstrates the efficacy of the decentralized algorithm in performance improvement.

D. Comparison with a centralized approach

The decentralized algorithm presents two major advantages in comparison with a centralized approach. It is computationally less expensive, and it is easier to implement in practice. The advantages and disadvantages are summarized in the table below for the V-formation example, in which vehicles may communicate with no more than 2 other vehicles.

V. NONCONVEX CASE

We now show how to solve the nonlinear, nonconvex problem. The only difference with the convex case is that

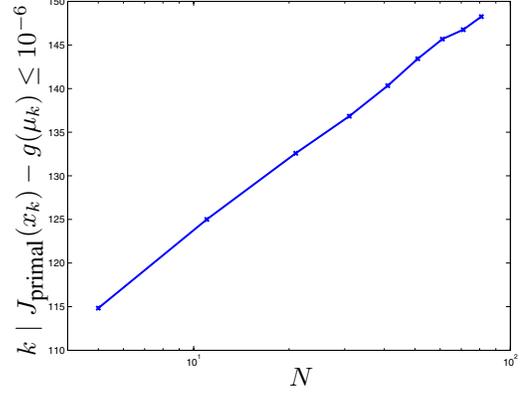


Fig. 3. Number of iterations of the decentralized algorithm required to obtain a duality gap of less than 10^{-6} as a function of the number of aircraft of the problem.

	Centralized	Decentralized
Algorithm	Solve primal (LQR)	Apply dual decomposition
Solution	Optimal solution	Optimal solution
Memory	$\mathcal{O}(T^2 N^3)$	N times $\mathcal{O}(T^2)$
Flop count	$\mathcal{O}(TN)^3$	N times $\mathcal{O}(\log(N)T^3)$
Practice	Communication with all agents, once	Communication with few agents, iteratively

TABLE I

COMPARISON OF THE DECENTRALIZED ALGORITHM WITH A CENTRALIZED LQR FOR THE V-FORMATION FLIGHT EXAMPLE.

each agent i , $i = 1, \dots, N$ has to solve the following problem in order to generate its optimal trajectory.

$$\begin{aligned} & \text{Minimize} && J_i(x_i) + \mu_i^T x_i \\ & \text{Subject to} && x_i \in \mathcal{D}_i \end{aligned} \quad (P5)$$

where μ_i is a dual variable, parameter of the problem; \mathcal{D}_i represents a nonlinear dynamics and J_i might not be convex. In order to solve (P5), we use optimal control techniques. Note that it is possible to solve the problem exactly using a Hamilton-Jacobi formulation (H-J). This approach is reasonable due to the low dimension of the problem (3 degrees of freedom for an aircraft modelled using a planar nonlinear kinematic model [13]). This is a major advantage over a direct centralized approach, on which this problem would be prohibitively complex. Typically, (H-J) equations can be solved up to 4 or 5 dimensions [13].

Note also that we can use another optimal control technique: the adjoint method. This method is computationally less expensive than solving (H-J) equations (because it only

computes the optimal control instead of the cost-to-go). Nevertheless, in theory, it only returns a local optimum and hence might fail to return the optimal trajectory. This is a crucial issue when we would like to provide a certificate on the duality gap. Indeed, in order to guarantee a lower bound on the primal global solution, a dual feasible point has to be generated, which means that (P5) must be solved exactly.

Solving (P5) exactly at each iteration of the decentralized algorithm would be quite expensive, therefore, in practice, the procedure is implemented as follows:

1) The subgradient method is performed using the adjoint method.

2) Once the algorithm has converged, the dual variables, to which the algorithm converged: μ_i are selected.

3) The exact solution of (P5) is computed at point μ_i , and hence, a certificate for the value of the dual is provided.

We now briefly explain how the nonlinear optimal trajectories are generated via the adjoint method.

Given a cost function $J = \int_0^T h(x(t), u(t)) dt$ and the dynamics $\dot{x}(t) = f(x(t), u(t))$, the gradient of the cost function with respect to the control input u is:

$$\nabla_u J(x, u) = \nabla_u f(x, u)p + \nabla_u h(x, u) \quad (14)$$

in which p is called the costate and satisfies the following backward equation:

$$\dot{p} = -\nabla_x H(x, u), \quad p(T) = 0 \quad (15)$$

where $H(x, u) = h(x, u) + p^T f(x, u)$ is the Hamiltonian of the system.

Adjoint-based Algorithm

1. Guess a control input (For example, start with the solution of the linearized problem.)

2. Repeat

- i) Compute the trajectory $\hat{x}(t) = f(x(t), u(t))$.
- ii) Solve the adjoint equation (15).
- iii) Construct the gradient $\nabla_u J$ of the cost function.
- iv) Update the control input: $u^{(k+1)} = u^{(k)} - \alpha_k \nabla_u J$

Terminate when the gradient is small.

A. Results

We choose the traditional non-linear kinematic model for the dynamics of each aircraft i .

$$\begin{cases} \dot{x}_i(t) &= v_i(t) \cos(\phi_i(t)) \\ \dot{y}_i(t) &= v_i(t) \sin(\phi_i(t)) \\ \dot{\phi}_i(t) &= \omega_i(t) \end{cases} \quad (16)$$

$(x_i(t), y_i(t))$ is the 2-D position of aircraft i at time t . $\phi_i(t)$ represents the heading angle. The velocity $v_i(t)$ and the turning rate $\omega_i(t)$ are the control variables. The scenario is similar to the one introduced in the previous section. A group of aircraft follows an aircraft leader while required to fly in V-formation. Figure 4 displays the trajectory of the aircraft given by our decentralized algorithm.

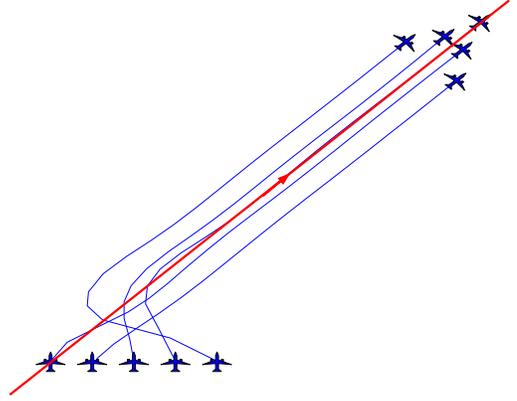


Fig. 4. An aircraft leader crosses the 2-D plane in straight line. 5 aircraft followers are required to fly in V-formation. The initial distribution of the followers has been chosen, on purpose, in contradiction to the relative positions required of the aircraft in formation.

B. Optimality

The primal problem (P1) is non convex and therefore strong duality may not hold. Nevertheless, the algorithm returns the duality gap at the end of the procedure, which gives an upper bound on the distance between our solution and the global optimum value.

Figure 5 shows the duality gap as a function of the number of iterations performed by the algorithm. The duality gap converges to zero. This gives the certificate that the trajectories generated are actually optimal.

Finally, note that the convergence rate of the algorithm is approximately the same as in the convex case (for 5 aircraft, 100 iterations are required to obtain a duality gap less than 10^{-5}).

VI. CONCLUSION

The problem of generating optimal trajectories for a set of cooperative aircraft is a high dimensional, non-linear problem. The aim of the paper is to derive a fast algorithm which would, first, return a good solution to the problem, and second, generate a decentralized procedure in agreement with the physical constraints on the information flow. Using the method of dual decomposition, we have demonstrated that a decentralized algorithm can be synthesized on board each aircraft which solve the problem exactly.

APPENDIX

Proof of Rule 1: For all (x, \tilde{x}) such that $s_p(x, \tilde{x}) = 0$,

$$f(x) + h(\tilde{x}) + \sum_{i=1}^{p-1} \mu_i^T s_i(x, \tilde{x}) = f(x) + h(\tilde{x}) + \sum_{i=1}^p \mu_i^T s_i(x, \tilde{x})$$

$$\geq \inf_{x \in \mathcal{D}, \tilde{x}} \{f(x) + h(\tilde{x}) + \sum_{i=1}^p \mu_i^T s_i(x, \tilde{x})\} = g_1(\mu_1, \dots, \mu_p) \quad (1)$$

Therefore, for all (μ_1, \dots, μ_p) ,

$$g_2(\mu_1, \dots, \mu_{p-1}) = \inf_{\substack{x \in \mathcal{D}, \tilde{x} \\ g_p(x, \tilde{x})=0}} \{f(x) + h(\tilde{x}) + \sum_{i=1}^{p-1} \mu_i^T s_i(x, \tilde{x})\} \geq g_1(\mu_1, \dots, \mu_p) \quad (2)$$

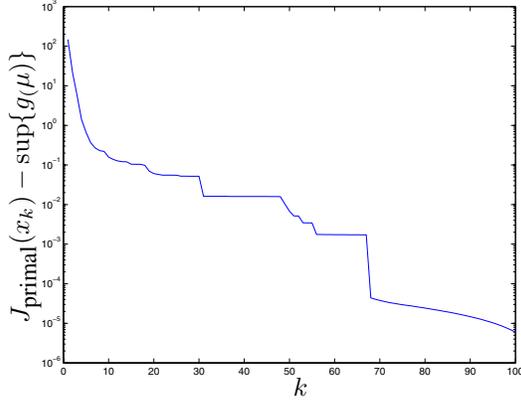


Fig. 5. Duality gap as a function of the number of iterations performed by the decentralized algorithm.

which proves that $\sup_{\mu_1, \dots, \mu_p} \{g_1\} \leq \sup_{\mu_1, \dots, \mu_{p-1}} \{g_2\}$.

Proof of Rule 2: For all (μ, ν) ,

$$g_4(\mu, \nu) = \inf_{x \in \mathcal{D}} \{f(x) - \mu^T x\} + \inf_{\tilde{x}, \tilde{\tilde{x}}} \{h_2(\tilde{x}, \tilde{\tilde{x}}) + \mu^T \tilde{x} + \nu^T (\tilde{x} - A\tilde{\tilde{x}})\} \quad (3)$$

Therefore,

$$\sup_{\nu} \{g_4(\mu, \nu)\} = \inf_{x \in \mathcal{D}} \{f(x) - \mu^T x\} + \sup_{\nu} \left\{ \inf_{\tilde{x}, \tilde{\tilde{x}}} \{h_2(\tilde{x}, \tilde{\tilde{x}}) + \mu^T \tilde{x} + \nu^T (\tilde{x} - A\tilde{\tilde{x}})\} \right\} \quad (4)$$

Let us now consider the problem:

$$\text{Minimize } h(\tilde{x}) + \mu^T \tilde{x} \quad (\text{P1})$$

It is equivalent to the following problem:

$$\begin{aligned} &\text{Minimize } h_2(\tilde{x}, \tilde{\tilde{x}}) + \mu^T \tilde{x} \\ &\text{Subject to } \tilde{\tilde{x}} = A\tilde{x} \end{aligned} \quad (\text{P2})$$

Since h_2 is convex, strong duality holds for (P2), hence:

$$\sup_{\nu} \left\{ \inf_{\tilde{x}, \tilde{\tilde{x}}} \{h_2(\tilde{x}, \tilde{\tilde{x}}) + \mu^T \tilde{x} + \nu^T (\tilde{x} - A\tilde{\tilde{x}})\} \right\} = \inf_{\tilde{x}} \{h(\tilde{x}) + \mu^T \tilde{x}\} \quad (5)$$

We conclude that for all μ ,

$$\sup_{\nu} \{g_4(\mu, \nu)\} = g_1(\mu) \quad (6)$$

which gives the result.

REFERENCES

- [1] J.P. AUBIN and I. EKELAND. Estimates of the duality gap in nonconvex optimization. *Math. Oper. Res.* **1**, 225-245, 1976.
- [2] D.P. BERTSEKAS. *Nonlinear Programming*. Athena Scientific, second edition, 1999.
- [3] D.P. BERTSEKAS. Constrained optimization and lagrange multiplier methods. *Academic Press, N.Y.*, 1982.
- [4] S. BOYD and L. VANDENBERGHE. *Convex Optimization*. Cambridge University Press, 2003.
- [5] P. CATHALIFAUD and T.R. BEWLEY. A noncausal framework for model-based feedback control of spatially-developing perturbations in boundary-layer flow systems. part i: Formulation. *Systems and Control Letters* **51** (1), 1-1, 2004.
- [6] C. LEMARECHAL and A. RENAUD. A geometric study of duality gaps, with applications. *Math. Program., Ser. A* **90**: 399-427 (2001).
- [7] R. D'ANDREA and G.E. DULLERUD. Distributed control design for spatially interconnected systems. *IEEE Transactions on Automatic Control*, **48**(9):1478-1495, 2003.
- [8] G.B. DANTZIG and P. WOLFE. Decomposition principle for linear programs. *Operations Research*, Vol. 8, pp. 101-111, 1960.
- [9] J.A. FAX. *Optimal and Cooperative Control of Vehicle Formations*. PhD Dissertation, California Institute of Technology, 2002.
- [10] G. INALHAN, D.M. STIPANOVIC, and C.J. TOMLIN. Decentralized optimization, with application to multiple aircraft coordination. *In the Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas*, December 2002.
- [11] A. JADBABAIE, J. LIN, and A.S. MORSE. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, Vol. 48, No. 6, pp. 988-1001, June 2003.
- [12] H-T L LEE, I.M. KROO, and S BIENIAWSKI. Flutter suppression for high aspect ratio flexible wings using microflaps. *AIAA paper 2002, 43rd AIAA/ASME/ASCE/AHS/ASC*, April 2002.
- [13] I. MITCHELL. *Application of Level Set Methods to Control and Reachability Problems in Continuous and Hybrid Systems*. PhD Dissertation, Stanford University, 2002.
- [14] P. OGREN, E. FIORELLI, and N.E. LEONARD. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *Preprint, submitted for publication*, 2003.
- [15] B.T. POLIAK. *Introduction to Optimization*. Optimization Software Inc., N.Y., 1987.
- [16] N.Z. SHOR. *Minimization Methods for Nondifferentiable Functions*. Springer-Verlag, Berlin, 1985.
- [17] D. SILJAK, S. S. STANKOVIC, X. B. CHEN, and M. R. MATAUSEK. Stochastic inclusion principle applied to decentralized automatic generation control. *International Journal of Control*, vol. 72, no. 5, pp. 276-288, 1999.
- [18] D.M. STIPANOVIC, G. INALHAN, R. TEO, and C.J. TOMLIN. Decentralized overlapping control of a formation of unmanned aerial vehicles. *To appear in Automatica*, 2004.
- [19] P. TABUADA, G.J. PAPPAS, and P. LIMA. Decentralizing formations of multi-agent systems. *Proceedings of the 10th Mediterranean Conference on Control and Automation, Lisbon, Portugal*, July 2002.
- [20] J.D. WOLFE, D.F. CHICHKA, and J.L. SPEYER. Decentralized controllers for unmanned aerial vehicle formation flight. *Proceedings of the AIAA Guidance, Navigation, and Control Conference, San Diego*, August 1996.