JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 34, No. 5, September–October 2011

Joint Metering and Conflict Resolution in Air Traffic Control

Jerome Le Ny* and George J. Pappas†
University of Pennsylvania, Philadelphia, Pennsylvania 19104

DOI: 10.2514/1.53253

This paper describes a novel optimization-based approach to conflict resolution in air traffic control, based on geometric programming. A key feature of this approach is its ability to also take into account various metering directives issued by the traffic flow management level, in contrast to most methods that focus purely on aircraft separation issues. Moreover, the proposed methodology can account for some of the nonlinearities present in the formulations of conflict resolution problems, while incurring only a small penalty in computation time with respect to the fastest linear-programming-based approaches. Integer variables can be introduced to improve the quality of the solutions and to include combinatorial choices, for example, to optimize over aircraft sequences in merging streams. Simulation results demonstrate the efficiency of the approach on various aircraft separation problems, including miles-in-trail and minutes-in-trail restrictions through airspace fixes and boundaries.

I. Introduction

GROWING airspace congestion has spurred significant research activity in the past decade to develop automated support for air traffic conflict detection and resolution. Conflict resolution consists in the local modification of aircraft trajectories in order to maintain a mandatory minimum distance between airborne aircraft. For example, under radar separation, this distance is usually set to 3–5 n mile of horizontal separation and 1000–2000 ft of vertical separation, depending on the airspace and aircraft types ([1], chapter 9). In addition to ensuring separation between aircraft, however, air traffic controllers (ATCs) often need to enforce traffic flow management directives in the form of restricted flow rates through various airspace boundaries and fixes [2] ([1], chapter 10). For example, handoffs between sectors and traffic flows entering a terminal area or a flow-constrained area during an airspace flow program can all be limited [3]. Such rate limitations are typically enforced by using miles-in-trail or minutes-in-trail restrictions at certain airspace fixes [4] ([1], chapter 10). As a result, the ATCs face path-planning problems that lie at the interface between traffic flow management and separation assurance [5]. With the exception of the Stream Option Manager of Niedringhaus [6], however, there is little research that integrates in the same framework aircraft separation and metering in order to support this task. This paper addresses this problem by presenting a trajectory optimization methodology that is capable of resolving conflicts between many aircraft in the presence of various metering constraints.

Numerous conflict resolution methods have been previously proposed (see, e.g., the survey of Kuchar and Yang [7]). These methods can essentially be divided into rule-based approaches [8–10], force field methods [11], and optimization-based approaches [6,12–14]. This paper belongs to the last category and adopts many of the simplifying assumptions made in previous research. First, the environment is two-dimensional, as in most of the literature [6,13–15]. Note, however, that except in case of imminent conflict, altitude changes are typically avoided for conflict resolution, in order to minimize passenger discomfort and maintain compatibility with the vertically layered structure of the airspace ([1], chapter 3). Second, aircraft are represented by kinematic models, as in, e.g., [6,13,14,16], with speed and heading changes assumed to occur instantaneously. More detailed dynamic models have been used in some previous work, at the expense of more complex computations [17,18]. Alternatively, one can implement the path-planning directives obtained here by perhaps increasing the separation distance used in our algorithms, similarly to the discussion in ([10], sec. 7), for example. Our method is also limited to small heading changes, which should not be too constraining for typical airspace configurations, as illustrated in Sec. V.C. Under assumptions similar to ours, Niedringhaus [6] proposes a general modeling framework for conflict resolution based on linear programming. Subsequent studies include optimization over the possible crossing patterns, using genetic algorithms [12], semidefinite-programming based relaxations [13], or mixed-integer linear programming [14]. Bilimoria [19] describes a geometric optimization method that resolves conflicts between pairs of aircraft in real time but cannot guarantee safety for multiple-aircraft conflicts. Hu et al. [16] consider the problem of designing three-dimensional conflict-free maneuvers minimizing a certain energy cost function and allow for a richer set of trajectories than the previously mentioned work: in particular, optimization over two-leg trajectories. Hwang et al. [10] propose a protocol based conflict resolution scheme where maneuvers consist of two straight paths of equal length. Moreover, while this paper assumes that the configurations and velocities of all aircraft involved in a particular conflict are known at a central computing facility, where the resulting path-planning problem is solved, many decentralized conflict resolution methods have been proposed as well (see, e.g., [10,15,20–22]).

The research efforts described in the previous paragraph are only concerned with the aircraft separation problem. Much less attention has been given to the possible presence of traffic flow management directives further constraining aircraft trajectories. Niedringhaus [6] discusses a scenario where aircraft are merged into a metered stream, similar to the problem described in Sec. IV. Bilimoria and Lee [23] and Mueller et al. [5] extend the previously mentioned two-aircraft conflict resolution algorithm [19] to include arrival time and metering constraints. Dugail et al. [24] describe the dynamics governing the upstream propagation of sector flow constraints by considering a limited scenario where aircraft velocities and path lengths can be controlled to accommodate an output flow rate restriction in an en route sector. It is shown in this paper that geometric programming [25] constitutes an efficient optimization framework to solve similar problems involving both separation and metering constraints for scenarios with multiple aircraft. Specifically, the typically nonlinear constraints arising in these problems can be expressed or conservatively approximated by posynomial constraints [25] in the decision variables, namely, aircraft speeds and headings. A cost function also expressed in the form of a posynomial can then be minimized subject to these constraints by using a geometric program (GP) solver. GPs can be solved by specialized interior-point methods almost as efficiently as linear programs

Received 30 November 2010; revision received 28 January 2011; accepted for publication 8 February 2011. Copyright © 2011 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/11 and $10.00 in correspondence with the CCC.

*Postdoctoral Researcher, Department of Electrical and Systems Engineering. Levine Hall 465, 3330 Walnut Street. Member AIAA.

†Professor, Department of Electrical and Systems Engineering. Levine Hall 460, 3330 Walnut Street.

[26,27]. By adding integer variables and solving a mixed-integer geometric program (MIGP), combinatorial choices can also be included, for example, to optimize over the set of crossing patterns [13,14] or to determine the best arrival order at a fix. MIGP solvers are less mature than mixed-integer linear program solvers, however, and thus the computation times observed once integer variables are included are significantly longer that those reported in [14], for example. On the other hand, even for standard conflict resolution problems, the nonlinearities allowed in a GP let us jointly optimize over headings and speeds, whereas the linear programming approach in [14] is restricted to optimizing over either speeds or headings and in the latter case requires that all aircraft fly at the same speed.

The rest of the paper is organized as follows. Section II introduces the air traffic model and describes the constraints of the optimization problem obtained from the aircraft separation conditions and metering specifications. These constraints are reformulated or approximated as posynomial constraints in Sec. III, which also provides a brief introduction to geometric programming. Section IV discusses a specific scenario where aircraft must be merged at a metered intersection and illustrates the use of mixed-integer geometric programming to determine the best aircraft sequence in the merged stream. Section V introduces a fast but local optimization algorithm based on geometric programming. Simulation results show that complex problem instances involving many aircraft and metered boundaries can be solved very efficiently.

II. Problem Formulation

A. Air Traffic Model

Consider N aircraft, initially in configurations $(x_{i,0}, y_{i,0}, \psi_{i,0}) \in \mathbb{R}^2 \times \mathcal{S}^1$ with speeds $v_{i,0} \in \mathbb{R}_{>0}$ and headings $\psi_{i,0}$, $i = 1, \ldots, N$. Let $\hat{\mathbf{p}}_i = (x_{i,0}, y_{i,0})$ denote the initial position of aircraft i. Following much of the previous work on conflict resolution [6,13,14,16], aircraft are represented by a kinematic model and can hence change their heading and speed instantaneously. Any two aircraft must always maintain a mandatory separation distance, denoted d, between themselves. For simplicity, we assume a homogeneous airspace here, where all pairs of aircraft must be separated by the same fixed distance d. A *conflict* occurs if two or more aircraft experience a loss of minimum separation. The problem of interest in this paper is to specify new headings ψ_i and speeds v_i so that the resulting straight-line trajectories of the aircraft involve no conflict. Hence, a single maneuver is executed at time $t = 0$ by all aircraft in order to enforce separation, as in [13,14], for example. With short-enough computation times, however, such a scheme can be executed at regular intervals to obtain a closed-loop control policy and piecewise-linear trajectories, as discussed in Sec. V.C. The choice of the decision variables $(\psi_i, v_i) \in \mathcal{S}^1 \times \mathbb{R}_{>0}$ aims at minimizing a given objective function, for example, penalizing deviations with respect to the initial aircraft trajectories. Finally, in addition to enforcing separation between aircraft, an ATC might also need to impose metering constraints on the trajectories crossing certain boundaries and waypoints, further restricting the set of admissible speeds and headings. Metering constraints are discussed in Sec. II.C.

B. Separation Constraints

1. *Infinite-Horizon Separation Constraints*

The fact that any two aircraft must be separated by the distance d at all times imposes constraints on the admissible velocities. Consider two aircraft i, j with initial positions $\hat{\mathbf{p}}_i$ and $\hat{\mathbf{p}}_j$, current positions $\mathbf{p}_i$ and $\mathbf{p}_j$, and velocity vectors set to $\mathbf{v}_i$ and $\mathbf{v}_j$ (see Fig. 1). Note that throughout the paper, vector quantities are represented in boldface. In the mobile frame centered at $\mathbf{p}_i$ and attached to aircraft i, aircraft j has relative velocity $\mathbf{v}_{ij} = \mathbf{v}_j - \mathbf{v}_i$ and aircraft i is immobile. First, a safety condition considered in previous work [13,14] is presented, which guarantees that aircraft will never be in conflict assuming no future change of speeds or headings. Note that this approach is conservative, since, in practice, the ATC has the option of modifying these parameters again at a later time.

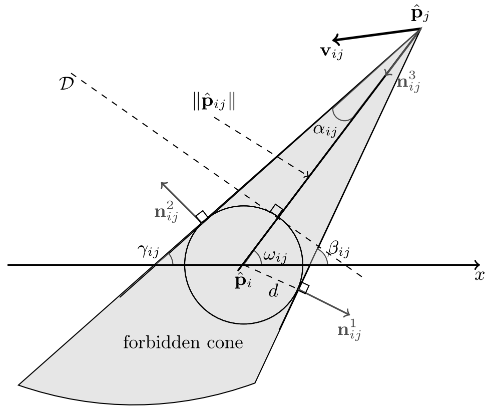

Fig. 1 Geometric representation of the forbidden cone of velocities.

No conflict arises between aircraft i and j if the distance from $\hat{\mathbf{p}}_i$ to the half-line $\hat{\mathbf{p}}_j + \mathbb{R}_{>0}\mathbf{v}_{ij}$ describing the trajectory of airplane j in the moving frame is at least d. Geometrically, this sufficient condition for safety means that the velocity vector $\mathbf{v}_{ij}$ lies outside of a "forbidden convex cone" with apex at $\hat{\mathbf{p}}_j$ and tangent to the disk of radius d centered at $\hat{\mathbf{p}}_i$ (see Fig. 1). This constraint has been formulated in previous work in at least two ways. Frazzoli et al. [13] represent it by a nonconvex quadratic inequality on $\| \mathbf{v}_{ij} \|$. This nonconvex constraint is then relaxed to obtain a semidefinite program, the solution of which serves to design a feasible solution using randomized rounding. This paper follows an alternative approach, as in, e.g., Pallottino et al. [14], which consists in representing the complement of the forbidden cone as the union of two half-spaces defined by two normal vectors $\mathbf{n}^1_{ij}$ and $\mathbf{n}^2_{ij}$ such that the admissible relative velocities satisfy

$$[\langle \mathbf{v}_{ij}, \mathbf{n}^1_{ij} \rangle \geq 0] \quad \vee \quad [\langle \mathbf{v}_{ij}, \mathbf{n}^2_{ij} \rangle \geq 0] \tag{1}$$

where $\vee$ denotes a disjunction (logical "or"). These separation constraints (1) can be written in coordinates as follows. Let $\hat{\mathbf{p}}_{ij} = \hat{\mathbf{p}}_j - \hat{\mathbf{p}}_i$. Denote $\alpha_{ij} = \arcsin(d/ \parallel \hat{\mathbf{p}}_{ij} \parallel) \in (0, \pi/2]$, and define the angles $\omega_{ij} = \arg(\hat{\mathbf{p}}_{ij})$, $\beta_{ij} = \omega_{ij} + \alpha_{ij}$, and $\gamma_{ij} = \omega_{ij} - \alpha_{ij}$ (see Fig. 1) with respect to a coordinate system centered at $\hat{\mathbf{p}}_i$. Then

$$\mathbf{n}\,^1_{ij} = [\cos(\beta_{ij} - \pi/2), \sin(\beta_{ij} - \pi/2)]^T = [\sin \beta_{ij}, -\cos \beta_{ij}]^T$$

and similarly $\mathbf{n}^2_{ij} = [-\sin \gamma_{ij}, \cos \gamma_{ij}]^T$. Hence, the first constraint in the disjunction becomes

$$\begin{aligned}
&(v_j \cos \psi_j - v_i \cos \psi_i) \sin \beta_{ij} - (v_j \sin \psi_j - v_i \sin \psi_i) \cos \beta_{ij} \geq 0 \\
&v_i(\sin \psi_i \cos \beta_{ij} - \cos \psi_i \sin \beta_{ij}) \\
&\quad + v_j(\cos \psi_j \sin \beta_{ij} - \sin \psi_j \cos \beta_{ij}) \geq 0 \\
&\text{i.e.,}\ \ v_i \sin(\psi_i - \beta_{ij}) - v_j \sin(\psi_j - \beta_{ij}) \geq 0
\end{aligned} \tag{2}$$

Similarly the second constraint in Eq. (1) becomes

$$\begin{aligned}
&-(v_j \cos \psi_j - v_i \cos \psi_i) \sin \gamma_{ij} + (v_j \sin \psi_j - v_i \sin \psi_i) \cos \gamma_{ij} \geq 0 \\
&v_i(\cos \psi_i \sin \gamma_{ij} - \sin \psi_i \cos \gamma_{ij}) + v_j(\sin \psi_j \cos \gamma_{ij} \\
&\quad - \cos \psi_j \sin \gamma_{ij}) \geq 0 \\
&\text{i.e.,}\ \ -v_i \sin(\psi_i - \gamma_{ij}) + v_j \sin(\psi_j - \gamma_{ij}) \geq 0
\end{aligned} \tag{3}$$

With the choice of a global coordinate system (Oxy) on the plane, aircraft can be ordered by increasing x coordinate. Then for any two aircraft i, j with $i < j$, i is the leftmost aircraft (i.e., the one with the smallest x coordinate), resulting in one of the two situations shown in Fig. 2. Now let $s_i^{\beta_{ij}} = \sin(\psi_i - \beta_{ij})$ and $s_{i,0}^{\beta_{ij}} = \sin(\psi_{i,0} - \beta_{ij})$, and similarly define $s_i^{\gamma_{ij}}$, $s_{i,0}^{\gamma_{ij}}$, $s_j^{\beta_{ij}}$, $s_{j,0}^{\beta_{ij}}$, $s_j^{\gamma_{ij}}$, and $s_{j,0}^{\gamma_{ij}}$. If at least one of the constraints (2) or (3) is satisfied, aircraft i and j are not in conflict. The

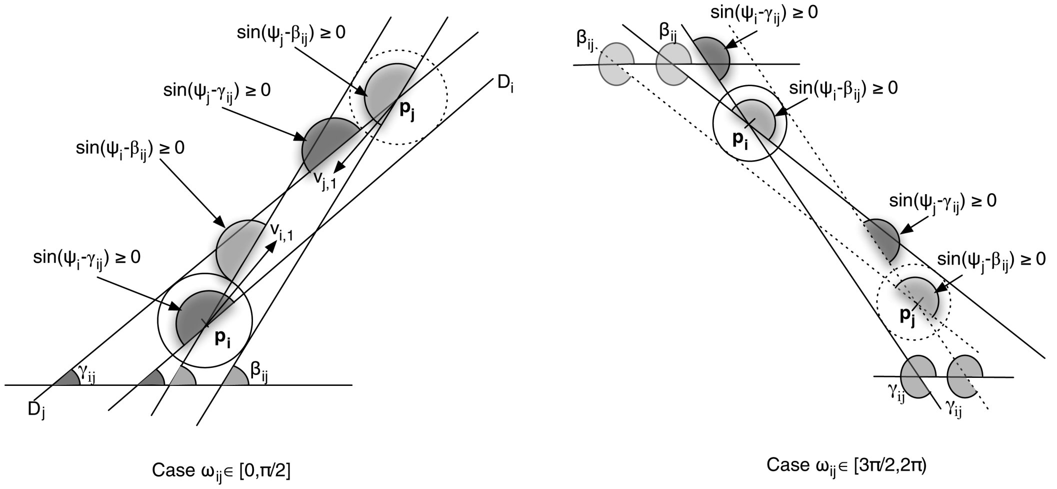

Fig. 2 Geometry of the constraints (2) and (3). The circles around each aircraft position have radius d equal to the mandatory separation distance. The definition of the set $\mathcal{C}_{ij}^0$ is illustrated in the left figure for $\mathbf{v}_i = \mathbf{v}_{i,1}$ and $\mathbf{v}_j = \mathbf{v}_{j,1}$.

following proposition gives necessary conditions for two aircraft to be in conflict.

Proposition 1: Suppose that aircraft *i and j* are initially in conflict. Then the following facts must all be true:

1) If $s_{i,0}^{\beta_{ij}} \geq 0$, then $s_{j,0}^{\beta_{ij}} > 0$. Otherwise, Eq. (2) holds for all positive values v_i, v_j.

2) If $s_{j,0}^{\beta_{ij}} \leq 0$, then $s_{i,0}^{\beta_{ij}} < 0$. Otherwise, Eq. (2) holds for all positive values v_i, v_j.

3) If $s_{i,0}^{\gamma_{ij}} \leq 0$, then $s_{j,0}^{\gamma_{ij}} < 0$. Otherwise, Eq. (3) holds for all positive values v_i, v_j.

4) If $s_{j,0}^{\gamma_{ij}} \geq 0$, then $s_{i,0}^{\gamma_{ij}} > 0$. Otherwise, Eq. (3) holds for all positive values v_i, v_j.

For reference, let us also record the initial heading values for which neither Eq. (2) nor Eq. (3) can be enforced by changing the speeds v_i, v_j only, i.e., for which both constraints remain infeasible for all (positive) speed values. Define

$$\mathcal{C}_{ij}^0 = \{(\psi_{i,0}, \psi_{j,0}) | [(s_{i,0}^{\beta_{ij}} < 0 \wedge s_{j,0}^{\beta_{ij}} \geq 0) \vee (s_{i,0}^{\beta_{ij}} = 0 \wedge s_{j,0}^{\beta_{ij}} > 0)] \wedge [(s_{j,0}^{\gamma_{ij}} < 0 \wedge s_{i,0}^{\gamma_{ij}} \geq 0) \vee (s_{j,0}^{\gamma_{ij}} = 0 \wedge s_{i,0}^{\gamma_{ij}} > 0)]\} \qquad (4)$$

Then if the initial headings of aircraft i and j belong to $\mathcal{C}_{ij}^0$, a heading change is necessary for one or both aircraft to resolve the conflict. The region $\mathcal{C}_{ij}^0$ corresponds to *head-on conflicts*, where the two aircraft are initially moving directly toward each other (see Fig. 2).

2. *Finite-Horizon Separation Constraints*

Condition (1) guarantees safety over an *infinite horizon* in the case where the velocity vectors are never changed after $t = 0$. In particular, it does not allow an aircraft in a stream to catch up with the preceding aircraft, because eventually a conflict would arise. This condition can be too conservative for certain finite-horizon problems considered in this paper. In the merging and metering scenario discussed in Sec. IV, separation between two aircraft must be ensured only until one of them reaches an airspace fix. It is assumed that safety is guaranteed beyond the fix, typically by a new ATC command. In this case, two aircraft following each other should be allowed to reduce their separation, potentially up to the minimum imposed by the metering restriction. Hence, the rest of this subsection presents safety constraints that guarantee separation until one of the aircraft reaches a specific waypoint.

Assume that aircraft i travels with fixed heading $\psi_i = \psi_{i,0}$ toward a waypoint at distance d_i from the initial position $\hat{\mathbf{p}}_i$. Moreover, suppose that for some aircraft $j \neq i$, the relative velocity vector $\mathbf{v}_{ij}$ belongs to the forbidden cone, i.e., condition (1) does not hold. The following additional sufficient condition ensures that separation is maintained until aircraft i reaches its waypoint, even if $\mathbf{v}_{ij}$ remains in the forbidden cone. Denote by $t_i = d_i/v_i$ the time it takes for aircraft i to reach the waypoint. Referring to Fig. 1, safety is maintained if aircraft i reaches the waypoint before j enters the circle of radius d around i. Note in Fig. 1 the dashed line $\mathcal{D}$ perpendicular to $\hat{\mathbf{p}}_{ij}$ and tangent to the circle around aircraft i, which separates the plane into two half-planes, each containing one aircraft. The sufficient condition considered consists in allowing cases where $\mathbf{v}_{ij}$ belongs to the forbidden cone, as long as aircraft j remains on the side of the line that does not contain i until time t_i. It is only a sufficient safety guarantee because safety would be maintained even if j were going past this line into the side regions of the forbidden cone around the circle. Since $\hat{\mathbf{p}}_{ij}$ is perpendicular to the line $\mathcal{D}$ and this line goes through the point $(d\hat{\mathbf{p}}_{ij})/\parallel \hat{\mathbf{p}}_{ij} \parallel$, the safety condition can be written

$$\left\langle \hat{\mathbf{p}}_{ij} + \frac{d_i}{v_i}\mathbf{v}_{ij}, \frac{\hat{\mathbf{p}}_{ij}}{\parallel \hat{\mathbf{p}}_{ij} \parallel} \right\rangle \geq d$$

Let us define the unit vector

$$\mathbf{n}_{ij}^{3} = -\frac{\hat{\mathbf{p}}_{ij}}{\parallel \hat{\mathbf{p}}_{ij} \parallel} = [-\cos\omega_{ij}, -\sin\omega_{ij}]^T$$

(see Fig. 1). This expression can then be rewritten

$$\frac{d_i}{v_i}\langle \mathbf{v}_{ij}, \mathbf{n}_{ij}^3 \rangle \leq \parallel \hat{\mathbf{p}}_{ij} \parallel - d \qquad (5)$$

This constraint can be again transformed into a linear constraint on the variables v_i, v_j, which in coordinates yields

$$\begin{aligned} &\frac{d_i}{v_i}[-(v_j\cos\psi_j - v_i\cos\psi_{i,0})\cos\omega_{ij} \\ &\quad - (v_j\sin\psi_j - v_i\sin\psi_{i,0})\sin\omega_{ij}] \leq \parallel \hat{\mathbf{p}}_{ij} \parallel - d \\ &v_i(\parallel \hat{\mathbf{p}}_{ij} \parallel - d - d_i\cos\psi_{i,0}\cos\omega_{ij} - d_i\sin\psi_{i,0}\sin\omega_{ij}) \\ &\quad + v_j d_i(\cos\psi_j\cos\omega_{ij} + \sin\psi_j\sin\omega_{ij}) \geq 0 \end{aligned}$$

i.e.,

$$v_i(\parallel \hat{\mathbf{p}}_{ij} \parallel - d - d_i\cos(\psi_{i,0} - \omega_{ij})) + v_j d_i\cos(\psi_j - \omega_{ij}) \geq 0 \quad (6)$$

The symmetric condition guaranteeing safety until j reaches a waypoint at distance d_j in direction $\psi_j = \psi_{j,0}$ gives the constraint

$$
\begin{aligned}
&\frac{d_j}{v_j}[-(v_j \cos\psi_{j,0} - v_i \cos\psi_i)\cos\omega_{ij} \\
&\quad - (v_j \sin\psi_{j,0} - v_i \sin\psi_i)\sin\omega_{ij}] \leq \|\, \hat{\mathbf{p}}_{ij} \,\| - d \\
&\quad - v_i d_j(\cos\psi_i \cos\omega_{ij} + \sin\psi_i \sin\omega_{ij}) + v_j(\|\, \hat{\mathbf{p}}_{ij} \,\| - d \\
&\quad + d_j \cos\psi_{j,0} \cos\omega_{ij} + d_j \sin\psi_{j,0} \sin\omega_{ij}) \geq 0
\end{aligned}
$$

i.e.,

$$
-v_i d_j \cos(\psi_i - \omega_{ij}) + v_j(\|\, \hat{\mathbf{p}}_{ij} \,\| - d + d_j \cos(\psi_{j,0} - \omega_{ij})) \geq 0 \tag{7}
$$

C. Metering Constraints

In addition to the separation requirements, metering directives at certain points (airspace fixes) or boundaries of the airspace also impose constraints on aircraft trajectories. These constraints are called *metering constraints* throughout the paper.

1. Distance Metering

Miles-in-trail restrictions constitute a commonly enforced metering constraint, where the ATC must maintain a specified distance, denoted henceforth by MIT, between aircraft in a stream. Consider the situation depicted in Fig. 3, with several aircraft converging toward a metering fix where a miles-in-trail restriction MIT must be enforced in the downstream flow. Suppose that aircraft j is to be inserted after aircraft i in this downstream flow. The speeds of aircraft i and j before reaching the fix are v_i and v_j, respectively, and the common speed of the aircraft in the downstream flow is set to v_0. If at time $t = 0$ aircraft i and j are at distance d_i and d_j away from the fix and heading toward it, the miles-in-trail restriction imposes the constraint

$$
\frac{d_i}{v_i} + \frac{\text{MIT}}{v_0} \leq \frac{d_j}{v_j} \tag{8}
$$

If $v_i = v_0$ or $v_j = v_0$ then this constraint can be rewritten as a linear constraint in the variables v_i, v_j, but in the general case it is nonlinear in the speed variables.

2. Time Metering

Instead of specifying a separation between successive aircraft in a stream by a distance MIT, the ATC can alternatively enforce a minutes-in-trail restriction, specifying a minimum time MINIT separating successive aircraft. Such constraints constitute a particularly natural interface with the traffic flow management level, which can restrict traffic flow rates through certain airspace boundaries or resources. For example, flow rate constraints can be imposed at the boundaries of flow-constrained areas during airspace flow programs or at arrival fixes in the vicinity of airports. A rate of at most x aircraft per hour through an air traffic control resource can be enforced using a minutes-in-trail restriction MINIT $= 60/x$. The notation MINIT is used in the following to specify separation times between two successive aircraft through any airspace resource, e.g., a boundary, not necessarily at a fix nor for aircraft forming a one-dimensional stream. For two successive aircraft i and j initially at distances d_i and d_j of their respective waypoints and subject to a minutes-in-trail restriction on the times at which they reach these waypoints, the following inequality must hold

$$
\frac{d_i}{v_i} + \text{MINIT} \leq \frac{d_j}{v_j} \tag{9}
$$

This constraint is again nonlinear in the decision variables v_i, v_j.

For the case of a flow-constrained boundary, the ATC can modify the heading of an aircraft passing through that boundary. Let us assume that the boundary is a line segment with orientation $\delta \in [0, \pi)$ with respect to the global fixed coordinate system (see Fig. 4). Then if aircraft i, initially at distance D_i from the line, is to cross this boundary, it must be the case that $\delta < \psi_i < \delta + \pi[\text{mod}\, 2\pi]$ or $\delta + \pi < \psi_i < \delta + 2\pi[\text{mod}\, 2\pi]$. It follows that in the first case, the inequality $\sin(\psi_i - \delta) > 0$ holds and

$$
d_i = \frac{D_i}{\sin(\psi_i - \delta)} \tag{10}
$$

whereas in the second case, the inequality $\sin(\psi_i - \delta) < 0$ holds and

$$
d_i = \frac{D_i}{\sin(\psi_i - (\delta + \pi))} = \frac{D_i}{\sin(\delta - \psi_i)} \tag{11}
$$

Suppose now that aircraft i and j, initially on the same side of a boundary so that Eq. (10) holds, are to cross the boundary in this order. Then Eq. (9) gives the constraint

$$
\frac{D_i}{v_i \sin(\psi_i - \delta)} + \text{MINIT} \leq \frac{D_j}{v_j \sin(\psi_j - \delta)}
$$

or

$$
\frac{D_i v_j \sin(\psi_j - \delta)}{D_j v_i \sin(\psi_i - \delta)} + \text{MINIT}\frac{v_j \sin(\psi_j - \delta)}{D_j} \leq 1 \tag{12}
$$

If Eq. (11) holds instead for both aircraft, the resulting constraint is

$$
\frac{D_i v_j \sin(\psi_j - (\delta + \pi))}{D_j v_i \sin(\psi_i - (\delta + \pi))} + \text{MINIT}\frac{v_j \sin(\psi_j - (\delta + \pi))}{D_j} \leq 1 \tag{13}
$$

Note that the requirement that both aircraft pass through the boundary imposes additional bounds on ψ_i and ψ_j, defined by the angles to the endpoints of the line segment representing the boundary.

D. Optimization-Based Conflict Resolution

The previous subsections described separation and metering constraints that restrict the set of admissible speeds and headings of the aircraft. The rest of the paper presents an optimization-based approach to aircraft trajectory planning, where a cost function depending on the $2N$ variables v_i and ψ_i ($i = 1, \ldots, N$) is minimized, subject to the separation and metering constraints. An example of such an optimization problem for metering aircraft at a fix is presented in Eqs. (25–28) below. The cost function is introduced to keep the trajectories close to the preferred ones while enforcing the various constraints. For example, the ATC might wish to minimize trajectory deviations when resolving conflicts. Such an approach,

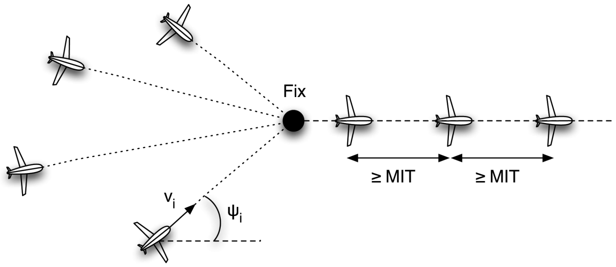

Fig. 3 Merging at a metered fix.

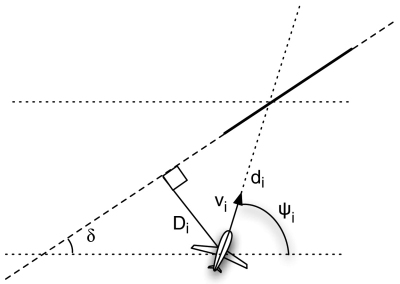

Fig. 4 Metered boundary. Case $\delta < \psi < \delta + \pi$.

however, requires an efficient optimization framework to handle the nonlinearities present in the constraints. Using standard nonlinear programming tools results in long computation times that are not compatible with the real-time aspect of the conflict resolution problems. Instead, it is shown in the following section that the constraints can be handled directly or approximately by fast geometric programming solvers.

III. Geometric Programming

After a brief introduction to geometric programming, this section develops posynomial approximations for some of the nonlinearities appearing in the separation and metering constraints of Sec. II. Conservative approximations are provided in order to guarantee safety.

A. Preliminaries

The separation constraints (2), (3), (6), and (7) are linear in the speed variables but nonlinear in the heading variables, and the metering constraints (8), (9), (12), and (13) are nonlinear in both the speed and heading variables. The ability to solve optimization problems subject to these constraints is very limited if standard nonlinear programming solvers are used. Moreover, if additional integer variables are introduced, e.g., to optimize over the aircraft arrival sequences in the scheduling scenario of Sec. IV, it is important to be able to solve the continuous optimization problems efficiently in order to implement a reasonably fast branch-and-bound method. It turns out that geometric programming [25], a class of efficiently solvable optimization problems, can handle exactly the nonlinearities in the speed variables and approximately the nonlinearities in the heading variables. The basic terminology of geometric programming is introduced in this subsection. See [25,27] for more details.

A *monomial* is a function $f\colon \mathbb{R}^n_{>0} \to \mathbb{R}$ of the form

$$f(x) = c x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$$

where $c > 0$ and $a_i \in \mathbb{R}$, for $1 \leq i \leq n$. A function f which is the sum of one or more monomials

$$f(x) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}}$$

is called a *posynomial*. Optimization problems involving the minimization of a posynomial function subject to constraints of the form $g(x) \leq 1$, where g is a posynomial, as well as $h(x) = 1$, where h is a monomial, are called *geometric programs* and can be solved almost as efficiently as linear programs by interior-point methods [26,27]. Indeed, the change of variable

$$x_i = e^{y_i} \tag{14}$$

which is valid, since the variables x_i are required to be positive, converts a GP to a convex optimization problem in the new variables y_i. Next, a *generalized posynomial* is a function that can be formed from posynomials by addition, multiplication, taking the maximum of several posynomials and taking a positive power of a posynomial [27]. Replacing posynomials by generalized posynomials in the definition of a GP gives a *generalized geometric program*, which can be solved efficiently as well by converting it to a GP.

In the rest of this subsection the headings of the aircraft are assumed fixed and the only decision variables are the aircraft speeds. These variables are positive, as requested by the form of geometric programs. Note that the metering constraint (9) can be rewritten

$$\frac{\text{MINIT}}{d_j} v_j + \frac{d_i}{d_j} v_j v_i^{-1} \leq 1 \tag{15}$$

and hence is of the form $g(v_i, v_j) \leq 1$, where g is a posynomial. Similarly, for fixed headings, the metering constraints (8), (12), and (13) are posynomial constraints in the speed variables. This is true for Eqs. (12) and (13) by noting that for the range of relevant headings the coefficients involving sine functions are positive [see Eqs. (10) and (11)]. The following lemma shows that the separation constraints (2), (3), (6), and (7) can be rewritten as posynomial constraints as well.

Lemma 2: Any linear constraint in the decision variables $x > 0$ and $y > 0$ of the form

$$\alpha x + \beta y \geq 0 \tag{16}$$

is either trivial (if $\alpha, \beta \geq 0$), infeasible (if $\alpha, \beta \leq 0$ and $\alpha + \beta \neq 0$), or can be rewritten as $g(x, y) \leq 1$, where g is a monomial.

Proof: In the last case the parameters α and β have opposite signs and are both nonzero. If $\alpha > 0$, $\beta < 0$, then the inequality can be rewritten as

$$\frac{|\beta|}{\alpha} \frac{y}{x} \leq 1$$

If $\alpha < 0$, $\beta > 0$, the constraint becomes

$$\frac{|\alpha|}{\beta} \frac{x}{y} \leq 1$$

Hence, after a simple preprocessing step, linear inequalities of the form in Eq. (16) can be handled by geometric programming. With the headings fixed, all the separation constraints (2), (3), (6), and (7) are constraints on the speed variables of the form of Eq. (16).

GP modeling also allows us to work directly with time to arrivals or delays, which are important metrics for ATCs ([1], chapter 10), in the objective function or in the constraints of the optimization problem. For example, suppose that aircraft i is cleared from the considered airspace when it reaches a waypoint at distance d_i. An ATC might want to optimize, subject to the separation and metering constraints, the total clearing time for all aircraft:

$$\min_{v_1,\ldots,v_N} \sum_{i=1}^{N} \frac{d_i}{v_i} \quad \text{or} \quad \min_{v_1,\ldots,v_N} \max_{i=1,\ldots,N} \left\{ \frac{d_i}{v_i} \right\} \tag{17}$$

which are a posynomial and generalized posynomial, respectively. These cost functions increase in priority the velocity of slower aircraft that are farther from their destination. If instead of clearing the airspace in minimum time the goal is to minimize the deviations with respect to the initial aircraft speeds, then an objective function such as the following can be used:

$$\min_{v_1,\ldots,v_N} \sum_{i=1}^{N} \max\left\{ \frac{v_{i,0}}{v_i}, \frac{v_i}{v_{i,0}} \right\} \tag{18}$$

which is a generalized posynomial in the decision variables $v_1, \ldots, v_N$, with minimum in the absence of constraints at $v_i = v_{i,0}$ for $i = 1, \ldots, N$.

B. Posynomial Approximations

When the aircraft headings are included as decision variables, the resulting optimization problems become complex nonlinear programs, with possibly many local minima. Instead of solving such nonlinear programs directly, this paper proposes to develop *conservative* approximations of the problems using local posynomial approximations of the nonlinearities. These approximate problem formulations can then be very efficiently and reliably solved by a GP solver using interior-point methods, and the obtained solutions are *always guaranteed to satisfy the original nonlinear constraints*. However, the approximations are satisfying only in the neighborhood of a given heading for each aircraft, e.g., ±15° around the initial headings $\{\psi_{i,0}\}_{1 \leq i \leq N}$ if the objective is to minimize trajectory deviations. Although the user can choose the amplitude of the allowed heading deviations, the approximations become overly conservative for large intervals.

Approximating the separation constraints (2) and (3) and metering constraints (12) and (13) requires posynomial approximations of the functions $\psi \to \sin(\psi - \beta)$ and $\psi \to 1/\sin(\psi - \beta)$ for $\psi \in (\beta, \beta + \pi)$. To work with positive variables, which is required in GPs, all angles are expressed as positive values, shifting them by a multiple of 2π if necessary. Hence, an interval such as $-\frac{\pi}{3} \le \psi \le \frac{2\pi}{3}$, is expressed instead as $\frac{5\pi}{3} \le \psi \le \frac{8\pi}{3}$. A given heading is involved in several constraints and the 2π shifts have to be chosen appropriately so that the resulting bounds remain compatible. Note that approximations of the finite-horizon separation constraints (6) and (7) are not presented in this paper, although a similar approach could be used in this case as well. These constraints (6) and (7) are only used in the merging and metering scenario described in Sec. IV, where optimization is performed over the speed variables only.

The most difficult of the two functions sin and $1/\sin$ to approximate is the sine function. The following simple monomial approximation [27] can be used in a neighborhood of $\psi = \psi_0$ and turns out to an upper bound over the relevant interval due to the concavity property of sin:

$$\text{for } \psi, \psi_0 \in (\beta, \beta + \pi), \qquad \sin(\psi - \beta) \le c\left(\frac{\psi}{\psi_0}\right)^a \tag{19}$$

where

$$c = \sin(\psi_0 - \beta), \qquad a = \psi_0 \cot(\psi_0 - \beta)$$

Boyd et al. [27] provide a general discussion of techniques for developing such monomial approximations. Figure 5 shows the behavior of the approximation, for angular deviations of ±10° around the value ψ_0. The approximation is the poorest for ψ_0 close to β or $\beta + \pi$, which results in missing feasible solutions in the approximation of the nonlinear program.

Let us now turn our attention to the function $1 \mapsto 1/\sin(\psi - \beta)$, for $\beta < \psi < \beta + \pi$. For ψ_0 away from the boundaries, say $\psi_0 - \beta > G$ and $\beta + \pi - \psi_0 > G$ for some $G > 0$, a similar local monomial approximation is satisfying, shifted upwards in order to obtain an upper bound on an interval $[\psi_0 - l, \psi_0 + r] \subset (\beta, \beta + \pi)$

$$\frac{1}{\sin(\psi - \beta)} \le c_0 + c_1\left(\frac{\psi}{\psi_0}\right)^a \tag{20}$$

where

$$c_1 = \frac{1}{\sin(\psi_0 - \beta)}, \qquad a = -\psi_0 \cot(\psi_0 - \beta)$$
$$c_0 = \max\left\{\frac{1}{\sin(\psi_0 - l - \beta)} - c_1\left(\frac{\psi_0 - l}{\psi_0}\right)^a, \frac{1}{\sin(\psi_0 + r - \beta)} - c_1\left(\frac{\psi_0 + r}{\psi_0}\right)^a\right\}$$

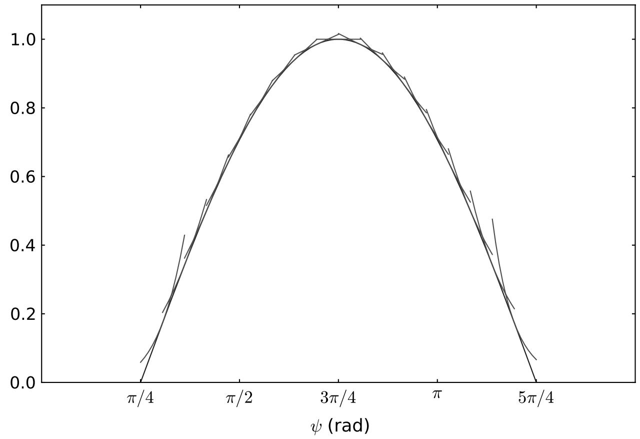

Fig. 5 Local monomial overapproximations of $\psi \mapsto \sin(\psi - \pi/4)$, for $\pi/4 < \psi < \pi/4 + \pi$. For $\psi_0 = \pi/4 + 10°$, and $\pi/4 + 20°, \ldots$, the approximation (19) is plotted for $\psi \in [\psi_0 - 10°, \psi_0 + 10°]$. Clearly, the quality of the approximation is worst for ψ_0 close to the boundaries of the interval $(\pi/4, 5\pi/4)$.

The monomial approximation alone under-approximates the function, but the addition of the constant c_0 results in an upper bound. For the remaining cases, first when $\beta < \psi_0 \le \beta + G$, the following upper bound holds on $(\beta, \psi_0 + r]$

$$\frac{1}{\sin(\psi - \beta)} \le c_0' + \frac{1}{(\psi - \beta)^a} \tag{21}$$

where

$$a = \frac{\log(\sin(0.4))}{\log(0.4)} \approx 1.029$$
$$c_0' = \max\left\{\frac{1}{\sin(\psi_0 + r - \beta)} - \frac{1}{(\psi_0 + r - \beta)^a}, 0\right\}$$

Symmetrically, when $\beta + \pi - G \le \psi_0 < \beta + \pi$, the following bound holds on $[\psi_0 - l, \beta + \pi)$

$$\frac{1}{\sin(\psi - \beta)} \le c_0'' + \frac{1}{(\beta + \pi - \psi)^a} \tag{22}$$

where

$$a = \frac{\log(\sin(0.4))}{\log(0.4)} \approx 1.029$$
$$c_0'' = \max\left\{\frac{1}{\sin(\psi_0 - l - \beta)} - \frac{1}{(\beta + \pi - \psi_0 + l)^a}, 0\right\}$$

The constants c_0' and c_0'' are again chosen in order to obtain an upper bound on the relevant interval. The constant a is chosen here to get an upper bound even without the constant term c_0' for Eq. (21) on $\psi \in (\beta, \beta + 0.4]$ and without c_0'' for Eq. (22) on $\psi \in [\beta + \pi - 0.4, \beta + \pi)$. This can be seen by noting that a is solution to the equation

$$\sin(0.4) = 0.4^a$$

from which it follows that $\sin(x) \ge x^a$ on [0,0.4], resulting in an upper bound for the inverses. The value $G = 0.4$ is chosen somewhat arbitrarily. Figure 6 shows the behavior of the approximation with $G = 0.6$, which is of similar quality.

Note, however, that the bounds on the right-hand side of Eqs. (21) and (22) are not a posynomials! Nonetheless, these expressions can be handled using geometric programming, as follows. The bound (21) is used to approximate the separation and metering constraints by stronger constraints of the form

$$\frac{p(v, \psi)}{(\psi_i - \beta)^a} + q(v, \psi) \le 1 \tag{23}$$

where p, q are posynomials in the speed and heading variables and $a > 0$. Such a constraint is then equivalent to the pair of constraints:

$$t + q(v, \psi) \le 1, \qquad \frac{p(v, \psi)}{(\psi_i - \beta)^a} \le t \tag{24}$$

where $t > 0$ is a new variable, and the second constraint in Eq. (24) can be rewritten as the generalized posynomial constraint:

$$\frac{p(v, \psi)^{1/a}}{\psi_i t^{1/a}} + \frac{\beta}{\psi_i} \le 1$$

The fact that Eq. (23) is equivalent to Eq. (24) can be seen as follows. If Eq. (23) is satisfied, then so is Eq. (24) with the choice $t = p(v, \psi)/(\psi_i - \beta)^a$. Conversely, if Eq. (24) is satisfied for some t, then t can be decreased until the second inequality in Eq. (24) becomes tight, i.e., Eq. (23) is satisfied. Expressions involving Eq. (22) are treated in a similar way.

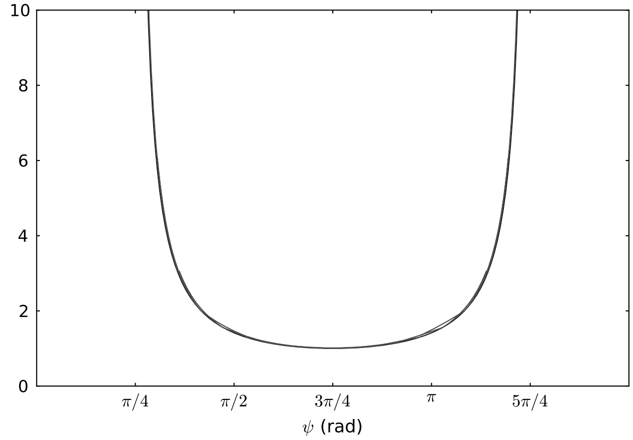

Fig. 6 Local overapproximations of $\psi \mapsto 1/\sin(\psi - \pi/4)$, for $\pi/4 < \psi < \pi/4 + \pi$. For $\psi_0 = \pi/4 + 10°$ and $\pi/4 + 20°, \ldots$, the approximations (19), (21), and (22) are plotted for $\psi \in [\psi_0 - 10°, \psi_0 + 10°]$. G is set to 0.6 here.

IV. Metering and Scheduling Using Mixed-Integer Geometric Programming

The rest of the paper presents specific scenarios involving conflict resolution and metering, and illustrates the application of the previously developed optimization framework based on geometric programming. In addition, this section discusses how disjunctions of separation and metering constraints can be modeled using mixed-integer geometric programming, in order to optimize over the combinatorial choices of possible crossing patterns and orderings of aircraft. The methodology is used to determine the optimal arrival order of aircraft at a metered fix in a stream-merging scenario.

A. Handling Disjunctions in Geometric Programs

Recall that our conflict resolution methodology consists in minimizing a posynomial cost function such as Eq. (17) or Eq. (18) subject to the safety and metering constraints discussed in Sec. II. Now consider the following joint scheduling and path-planning scenario at an airspace fix subject to a minutes-in-trail restriction MINIT. The goal is to merge a number of planes heading toward the fix in a single stream down the fix, while satisfying the metering constraint specifying that two successive aircraft reaching the fix must be separated by at least MINIT minutes. This scenario is represented in Fig. 3, for the similar problem involving a miles-in-trail restriction MIT. The optimal aircraft order at the fix is to be determined as well. The only continuous decision variables available here are the speeds v_i, $i = 1, \ldots, N$. The headings are fixed to their initial values $\psi_{i,0}$, with the aircraft directed toward the fix, so that each aircraft can reach the metering fix using a straight-line trajectory. The optimization problem to solve is then

$$\min\ f(v_1, \ldots, v_N) \tag{25}$$

$$\text{subject to}\ \ v_{i,\min} \leq v_i \leq v_{i,\max}, \quad i = 1, \ldots, N \tag{26}$$

$$\begin{gathered}\text{Eq. (2)} \vee \text{Eq. (3)} \vee \text{Eq. (6)} \vee \text{Eq. (7)}\\ 1 \leq i \leq N-1, i+1 \leq j \leq N\end{gathered} \tag{27}$$

$$\begin{gathered}\left[\frac{d_i v_j}{d_j v_i} + \text{MINIT}\frac{v_j}{d_j} \leq 1\right] \vee \left[\frac{d_j v_i}{d_i v_j} + \text{MINIT}\frac{v_i}{d_i} \leq 1\right]\\ 1 \leq i \leq N-1, \qquad i+1 \leq j \leq N\end{gathered} \tag{28}$$

The objective function f in Eq. (25) is chosen to be a generalized posynomial. For a pair (i, j), satisfying one of the constraints in the disjunction (27) is sufficient to ensure separation of the aircraft until one of them reaches the fix. The metering constraints (28) (see Sec. II.C.2) also involve a disjunction, where the first (respectively, second) literal is selected if aircraft i (respectively, j) reaches the fix first. Note again that instead of fixing the aircraft ordering at the fix a priori, solving this program directly provides the best aircraft ordering.

The disjunctions in the optimization problem can be modeled by introducing additional integer variables. Recall [14,28] the standard "big-M" formulation used to model a disjunction of linear constraints

$$a_1^T x \leq b_1 \vee a_2^T x \leq b_2$$

via mixed-integer linear programming. Assuming that bounds on the variables x are available, the disjunction above can be rewritten as the conjunction

$$a_1^T x \leq b_1 + cM, \qquad a_2^T x \leq b_2 + (1-c)M, \qquad c \in \{0, 1\}$$

for some sufficiently large M, where c is a new binary variable. A mixed-integer linear program (MILP) solver implementing a branch-and-bound procedure, for example, obtains lower bounds by relaxing the binary constraint to $0 \leq c \leq 1$. This idea does not work directly with geometric programs, however. Indeed, the equivalent of the first constraint would be $g(x) \leq 1 + cM$, with g a posynomial, which is *not* a posynomial constraint when c is relaxed to $0 \leq c \leq 1$, since $\tilde{g}(x, c) = g(x) - cM$ is not a posynomial. However, the method can be modified as follows. Consider the disjunction of posynomial constraints

$$g(x) \leq 1 \quad \vee \quad h(x) \leq 1 \tag{29}$$

and the conjunction

$$\begin{gathered}g(x) + 2M/c \leq 1 + 2M, \qquad h(x) + cM \leq 1 + 2M\\ c \in \{1, 2\}\end{gathered} \tag{30}$$

for M sufficiently large (assuming that bounds $0 < \underline{b} < x < \bar{b}$ are available, which is the case for our decision variables $v_1, \ldots, v_N$). Then for $c = 1$, the first constraint in Eq. (29) is enforced, and for $c = 2$, the second constraint is enforced. When the constraint on c is relaxed to $1 \leq c \leq 2$, Eq. (30) can be handled by a GP solver, since the left-hand sides of the inequalities consist of posynomials.

More generally, for a disjunction of n posynomial constraints

$$g_1(x) \leq 1 \vee, \ldots, \vee g_n(x) \leq 1$$

one can introduce n integer variables $b_1, \ldots, b_n \in \{1, 2\}$ and consider the conjunction of constraints

$$g_i(x) + b_i M \leq 1 + 2M, \qquad i = 1, \ldots, n$$

together with the posynomial constraint

$$2/(b_1, \ldots, b_n) \leq 1 \tag{31}$$

The inequality (31) forces at least one of the b_i variables to be 2 and forces the corresponding constraint in the disjunction to be enforced. Modeling the disjunctions in the separation and metering constraints (27) and (28) in this manner leads to a MIGP. In addition, to find the solution of the minimization problem, it is sufficient to consider the situation where exactly one of the variables b_i has value 2, enforcing just one constraint per clause, hence replacing Eq. (31) by

$$2/(b_1, \ldots, b_n) = 1 \tag{32}$$

Constraint (32) is again a valid geometric programming constraint once the variables b_i are relaxed to $1 \leq b_i \leq 2$, since the left-hand side is a monomial. This constraint is preferable to Eq. (31), since it reduces the search space, for example, by eliminating the variable b_n.

The above big-M method can be used to reformulate the program Eqs. (25–28), with f a posynomial, as an MIGP with N continuous variables $v_1, \ldots, v_N$ and $2N(N-1)$ binary variables. The number of binary variables is problematic, however, as the development of MIGP solvers is not as advanced as for MILP solvers. With the currently available open-source solver Bonmin,[‡] problems with five

[‡] Data available online at http://www.coin-or.org/Bonmin [retrieved 30 November 2010].

Table 1 Computation times for the MIGP solving the fix-scheduling scenario[a]

No. aircraft	No. binary variables	Median comp. time	Std dev	Max
2	4	0.12 s	0.14 s	1.44 s
3	12	0.44 s	0.29 s	1.8 s
4	24	1.33 s	1 s	4.4 s
5	40	8.5 s	10.7 s	51.1 s
6	60	61.4 s	44.3 s	208 s

[a] The problems were solved using the branch-and-bound algorithm implemented in Bonmin on a 3.06 GHz Intel Core 2 Duo processor with 4 GB of RAM. For each line of the table, 100 simulations were conducted with random initial positions for the aircraft. The number of binary variables shown accounts for the trivial elimination of one of the variables for each constraint (32).

aircraft can be solved reliably in a few seconds, and problems with six aircraft can be solved in about 1 min, as described in Table 1.

B. Simulation Results

The proposed approach is now illustrated in numerical experiments for particular instances of the scenario described in the previous paragraph. The aircraft initial positions are generated randomly in a 100 × 400 n mile rectangle, and the planes must all pass through an airspace fix situated 400 n mile away (see Fig. 7). A minutes-in-trail restriction of 10 min between successive aircraft is enforced at the fix, and the separation distance between aircraft is fixed to $d = 10$ n mile. The objective to minimize is the clearing time objective (17).

The resulting MIGPs (25–28) are solved by rewriting the relaxed geometric programs in convex form via the change of variable (14) and using the mixed-integer nonlinear programming (MINLP) solver Bonmin. A more satisfying solution would be to use a dedicated MIGP solver, but currently such solvers do not seem to be easily available. Table 1 presents the computation times for scenarios involving between two and six aircraft. Among the various algorithms implemented in Bonmin, the simple branch-and-bound algorithm was found to be the most reliable, although not always the fastest, and the simulation results presented relate to this algorithm. Bonmin is an exact solver for such mixed-integer convex programs, that is, it eventually returns the optimal solution if the problem is feasible.

MIGP and mixed-integer convex program solvers are not as mature as MILP solvers, and progress in this area would be very beneficial for our applications. The simulation results of Table 1 show that for six aircraft involved simultaneously in a conflict, half of the generated cases required more than 1 min of computation time. Two effects contribute to make instances with more aircraft more difficult to solve: the number of integer variables introduced, and the fact that with more aircraft it becomes harder to find feasible solutions. Figure 8 shows the distribution of the computation times for 100 simulations with five aircraft. Note the presence of a few outliers for which the computation time is much larger than for the typical case.

V. Optimization-Based Heuristic for Separation and Metering

The inclusion of integer variables in the optimization problem as in the previous section allows us to optimize over the aircraft positions in a stream or the crossing patterns of the trajectories. In particular, with this device it is possible to decide optimally for each pair of aircraft on which side of the forbidden cone the relative velocity vector should lie. However, for complex problems involving many aircraft, MIGP solvers are not fast enough for real-time applications. This section presents a fast heuristic based on geometric programming, which essentially chooses a priori which constraints in the disjunctions such as Eqs. (27) and (28) to enforce. In contrast to the scenario of Sec. IV, it is now assumed that the ATC can modify both aircraft speeds and headings within given bounds.

The proposed algorithm first resolves the head-on conflicts, i.e., those for which the initial aircraft headings are within the sets $\mathcal{C}^0_{ij}$ defined in Eq. (4), by a necessary heading modification. It then locally optimizes the velocities and headings while ensuring separation, assuming in this second step that only small heading changes are allowed. This local optimization step relies on the posynomial approximations of the trigonometric nonlinearities in the separation and metering constraints presented in Sec. III.B. The algorithm does not provide a global optimum for the original problem in general and,

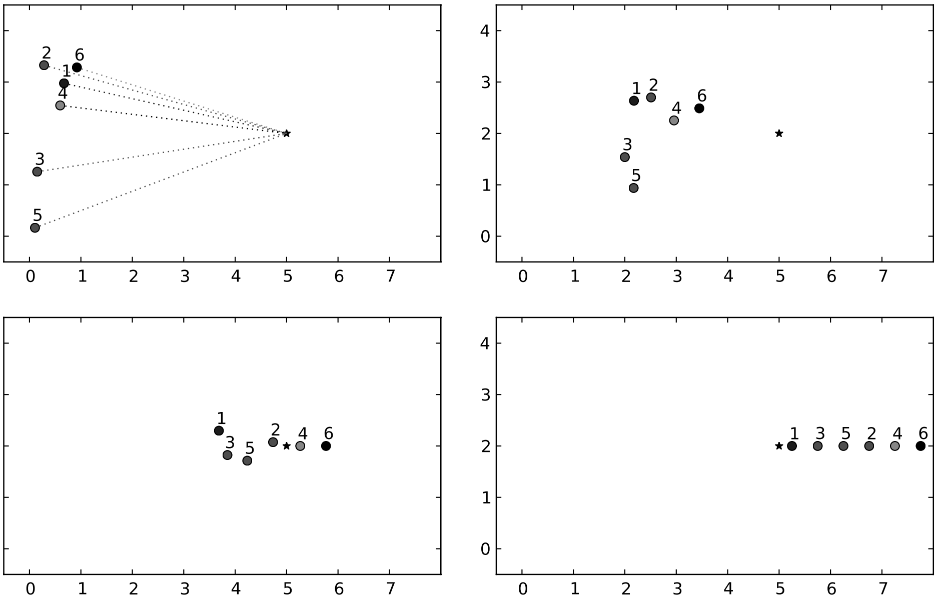

Fig. 7 Four snapshots with the positions of six aircraft approaching a merging point [at coordinates (5,2)], subject to a restriction MINIT = 10 min, and with mandatory separation distance equal to 10 n mile (the small circles of radius 5 n mile are not allowed to overlap). A unit on the figure corresponds to 100 n mile. The velocities are constrained to be between 200 kt and 400 kt for all aircraft. Once an aircraft reaches the fix, a new ATC command instructs it to follow the heading $\psi = 0°$ at a speed of 300 kt. The solution to the MIGP determines an optimal aircraft ordering for the first cost function in Eq. (17), i.e., minimizes the clearing time, while maintaining separation at all times. The first aircraft reaches the fix at $t = 64$ min and the subsequent aircraft cross the fix every 10 min after that time. The optimal velocities obtained are $v_1 = 233$ kt, $v_2 = 349$ kt, $v_3 = 283$ kt, $v_4 = 359$ kt, $v_5 = 333$ kt, and $v_6 = 400$ kt.

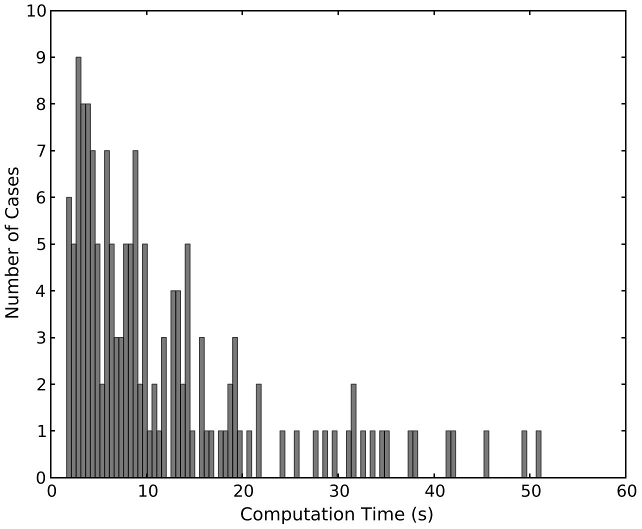

Fig. 8 Distribution of the computation times for the fix-scheduling problem with five aircraft, over 150 simulations.

in particular, it does not optimize over the possible crossing patterns. Because it only solves a convex optimization problem, however, it can be used in real time and in a periodic reoptimization strategy, as demonstrated in Sec. V.C. Moreover, a solution returned by this algorithm is guaranteed to be safe for our kinematic aircraft model. The two phases of the algorithm are described next in more details.

A. Phase I: Removing Configurations in $\mathcal{C}^0_{ij}$

As discussed in Sec. II.B.1, for two aircraft i and j with headings in $\mathcal{C}^0_{ij}$, a heading change for at least one of the aircraft is necessary to avoid loss of separation. Moreover, this required heading change is not necessarily small if the two aircraft are close to each other. In Phase I of the algorithm, the occurrences of such configurations between any two aircraft in the region under consideration are removed by modifying the headings only. More precisely, two aircraft $i < j$ with $(\psi_{i,0}, \psi_{j,0})$ in $\mathcal{C}^0_{ij}$ execute right-turn maneuvers, as described in Fig. 9. All aircraft turn in the same direction in order to resolve these head-on conflicts. Hence, to resolve conflicts using right turns, the headings $(\psi_{i,0}, \psi_{j,0}) \in \mathcal{C}^0_{ij}$ are reset to the new values $\psi_{i,0} = \gamma_{ij} - \epsilon[\mathrm{mod}\, 2\pi]$ and $\psi_{j,0} = \gamma_{ij} + \pi - \epsilon[\mathrm{mod}\, 2\pi]$. Here, ϵ is a small positive number used to simplify the analysis later on. This strategy is clearly not optimal, but is satisfying in the typical situations where only a small number of such head-on conflicts are expected to occur.

Resolving one head-on conflict can create new ones with other aircraft. These conflicts are then resolved similarly one at a time, always with the same turning direction, until a configuration free from head-on conflicts is found. Note that this process is not theoretically guaranteed to terminate. Indeed, some artificial examples lead to head-on conflicts that cannot be resolved by such an iterative scheme, whereas under our assumptions neglecting aircraft dynamics, conflicts can always be trivially resolved, e.g., by simply setting the headings of all aircraft to the same value. However, this heuristic proved to be sufficient in all the simulations presented in Sec. V.C. Potentially much more computationally intensive procedures could also be used to handle head-on conflicts in this phase more rigorously, e.g., using mixed-integer linear programming [14].

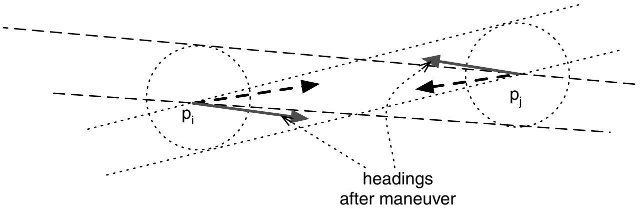

Fig. 9 Maneuver to remove configurations in $\mathcal{C}^0_{ij}$. The direction of rotation of the aircraft is fixed and the same for all planes (here, all planes turn right to resolve head-on conflicts). As can be seen on the figure even in the case of two planes, this rule is not necessarily optimal, since turning left would have resulted in a smaller heading change in this case, at least in the absence of other conflicts.

In addition, in this first phase of the algorithm any initial heading $\psi_{i,0}$ that is exactly equal to $\beta_{ij}, \gamma_{ij}, \beta_{ij} + \pi$, or $\gamma_{ij} + \pi$ for some $j > i$ is also perturbed by some small positive value ϵ, in a direction avoiding the creation of a new configuration in $\mathcal{C}^0_{ij}$, until all these limit cases are removed. This procedure simplifies the analysis of the second phase of the algorithm, presented next.

B. Phase II: Local Parameter Optimization via Geometric Programming

At the beginning of the second step of the algorithm, as a result of executing the first phase discussed above, no aircraft i has its heading $\psi_{i,0}$ exactly equal to $\beta_{ij}, \gamma_{ij}, \beta_{ij} + \pi$ or $\gamma_{ij} + \pi$ for some $j > i$, and no pair of aircraft $i < j$ has its headings $(\psi_{i,0}, \psi_{j,0})$ in the set $\mathcal{C}^0_{ij}$. Speeds and headings are now further adjusted to resolve any remaining conflict and optimize a cost objective. However, the heading variations are restricted to a prespecified small interval for each aircraft, such as ±15° around the values $\{\psi_{i,0}\}_{1 \leq i \leq N}$. Note that similar restrictions are made in some previous work [6,13], usually motivated by more physical concerns associated with large heading changes. In realistic scenarios where conflicts are detected long enough in advance, the necessary changes in headings (and velocities) are typically small. For example, in all simulation results reported in [14], heading changes were smaller than 12°, even though speed changes were not allowed.

1. *Heuristic Choice of Literal in the Separation Constraints*

To simplify the computations and avoid the introduction of combinatorial choices, one of the separation constraints (2) or (3) must be chosen a priori for each pair $i < j$ of aircraft. The finite-horizon constraints (6) and (7) are not used here but could be included to reduce conservativeness. For each pair $i < j$ of aircraft, the algorithm distinguishes between two cases

1. one of the implications 1–4 in Proposition 1 is not true, and thus one of the constraints (2) or (3) holds trivially for all v_i, v_j in $\mathbb{R}_{>0}$. This means that the aircraft i, j are initially not in conflict, and no conflict between them can arise by simply changing their speeds. The algorithm then chooses the trivial constraint (2) or (3) for this pair of aircraft, and imposes bounds on ψ_i, ψ_j ensuring that this constraint remains trivial, i.e., that the signs of the sine functions do not change. For example, if $\sin(\psi_{i,0} - \beta_{ij}) > 0$ and $\sin(\psi_{j,0} - \beta_{ij}) < 0$, the inequality (2) is valid for all positive values of v_i, v_j and the following constraints are added

$$\beta_{ij} < \psi_i < \beta_{ij} + \pi[\mathrm{mod}\, 2\pi],$$
$$\beta_{ij} + \pi < \psi_j < \beta_{ij} + 2\pi[\mathrm{mod}\, 2\pi]$$

With these bounds in place, the separation constraint for the pair i, j is removed, as any feasible choice of v_i, v_j, ψ_i, and ψ_j automatically guarantees separation.

2. all the implications 1–4 in Proposition 1 hold. In this case a conflict can arise for certain values of v_i, v_j and the initial headings. The algorithm enforces one of the constraints (2) or (3), while respecting the following rule: it must be possible to satisfy the enforced constraint for some positive values of v_i, v_j without changing the signs of the trigonometric terms in the constraint.

For an example illustrating case 2, if $\sin(\psi_{i,0} - \beta_{ij}) < 0$ and $\sin(\psi_{j,0} - \beta_{ij}) > 0$, then constraint (2) cannot be chosen, because it cannot lead to a valid inequality without changing the sign of $\sin(\psi_i - \beta_{ij})$ or $\sin(\psi_j - \beta_{ij})$. In this case, however, since the headings $(\psi_{i,0}, \psi_{j,0})$ are not in $\mathcal{C}^0_{ij}$, constraint (3) can be chosen. This is a consequence of the definition (4), which leads to the conclusion

$$(s^{\gamma_{ij}}_{j,0} \geq 0 \vee s^{\gamma_{ij}}_{i,0} < 0) \wedge (s^{\gamma_{ij}}_{j,0} \neq 0 \vee s^{\gamma_{ij}}_{i,0} \leq 0)$$

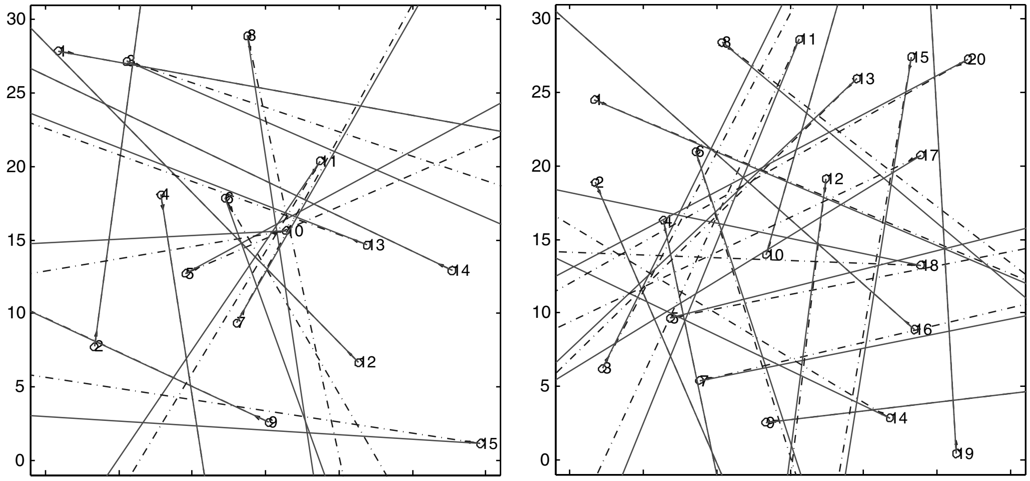

Fig. 10 Conflict resolution with 15 (left) and 20 (right) aircraft in a 300 × 300 n mile square. Each aircraft initial position is represented together with a circle of diameter 5 n mile around it (one unit on the figure represents 10 n mile). Dashed lines represent the initial trajectories, and solid lines are the adjusted trajectories after conflict resolution. The parameter ρ is set to 1 in Eq. (37). In the 15-aircraft example, the maximum heading change resulting from the optimization is 10.5° and the maximum speed change is 9 kt. In the 20-aircraft example, the maximum heading change is 9.5° and the maximum speed change is 12 kt.

It is not hard to verify that under this condition, inequality (3) is valid for some positive values of v_i, v_j. It is always the case that either constraint (2) or (3) can be chosen, and in some cases, both constraints could be chosen. The choice of which constraint to enforce in this last case can have a strong influence on the feasibility of the overall optimization problem. Referring to Fig. 1, note that satisfying constraint (2) means that $\mathbf{v}_{ij}$ is pushed outside of the forbidden cone in the direction of $\mathbf{n}^1_{ij}$. Similarly, satisfying constraint (3) means that $\mathbf{v}_{ij}$ is pushed in the direction of $\mathbf{n}^2_{ij}$. The algorithm chooses constraint (2) if the initial relative velocity vector $\hat{\mathbf{v}}_{ij}$ is on the side of the line $(\hat{\mathbf{p}}_i, \hat{\mathbf{p}}_j)$ that is in the direction of $\mathbf{n}^1_{ij}$ and chooses constraint (3) otherwise. In other words, it keeps the initial and final relative velocity vectors on the same side of the line $(\hat{\mathbf{p}}_i, \hat{\mathbf{p}}_j)$.

Once a constraint (2) or (3) to enforce has been chosen in case 2, bounds on ψ_i, ψ_j are added that prevent the trigonometric terms to change signs in the optimization procedure. For example, if constraint (3) is chosen, with say $\sin(\psi_{i,0} - \gamma_{ij}) < 0$ and $\sin(\psi_{j,0} - \gamma_{ij}) < 0$, then the following constraints are added:

$$\gamma_{ij} + \pi < \psi_i, \psi_j < \gamma_{ij} + 2\pi[\mathrm{mod}\, 2\pi]$$

Moreover, negative terms of the form $\sin(\psi - B)$ in the constraints (2) and (3), i.e., with $B + \pi < \psi < B + 2\pi$, are rewritten $-\sin(\psi - A)$, with $A = B + \pi$ and thus $A < \psi < A + \pi$. With these additional bounds on headings in place, the chosen constraint (2) or (3) is then rewritten in the form

$$\frac{v_k}{v_l}\frac{\sin(\psi_k - A)}{\sin(\psi_l - A)} \le 1 \tag{33}$$

where $(k, l) = (i, j)$ or (j, i), and $A = \beta_{ij}$, γ_{ij}, $\beta_{ij} + \pi$ or $\gamma_{ij} + \pi[\mathrm{mod}\, 2\pi]$. The case where $\sin(\psi_l - A) = 0$ in Eq. (33) need not be considered, since the preprocessing step in phase I of the algorithm eliminated the problematic initial conditions by introducing small perturbations.

2. *Optimization*

At this point of the procedure, a separation constraint to enforce has been chosen for each pair of aircraft and expressed in the form of Eq. (33). The next step is to perform posynomial approximations of these inequalities around the headings $\{\psi_{i,0}\}_{1\le i\le N}$ obtained at the end of phase I, using the conservative approximations developed in Sec. III.B. Explicit bounds are added to constrain the heading variables to remain in the intervals where these approximations are valid. The interval of heading variations cannot be increased too much because the resulting posynomial constraints become overly conservative. In the simulations presented in the next section, the heading variations are limited to ±15°. Note that the satisfaction of the posynomial inequalities guarantees that the separation constraints (33) are satisfied.

Hence, the upper bounds introduced in Sec. III.B lead to replacing Eq. (33) by more conservative constraints of the form

$$\frac{v_k}{v_l} c_k \left(\frac{\psi_k}{\psi_{k,0}}\right)^{a_k} \left(c_{l,0} + c_{l,1}\left(\frac{\psi_l}{\psi_{l,0}}\right)^{a_l}\right) \le 1 \tag{34}$$

by using Eqs. (19) and (20). If Eqs. (19) and (21) are used instead for $\psi_{l,0} \in (A, A + G]$, the constraint takes the form

$$\frac{v_k}{v_l} c_k \left(\frac{\psi_k}{\psi_{k,0}}\right)^{a_k} \left(c_{l,0} + \frac{1}{(\psi_l - A)^{a_l}}\right) \le 1 \tag{35}$$

Finally, if $\psi_{l,0} \in [A + \pi - G, A + \pi)$, the following constraint based on Eqs. (19) and (22) is used to replace Eq. (33):

$$\frac{v_k}{v_l} c_k \left(\frac{\psi_k}{\psi_{k,0}}\right)^{a_k} \left(c_{l,0} + \frac{1}{(A + \pi - \psi_l)^{a_l}}\right) \le 1 \tag{36}$$

Constraint (34) is a posynomial constraint. It was also explained at the end of Sec. III.B how to convert Eqs. (35) and (36) to equivalent posynomial constraints. Finally, metering constraints such as Eq. (12) can be similarly handled by geometric programming using the same approximations of the sine and inverse sine functions. With the cost function and all constraints expressed in the form of posynomials to approximate the original problem, the algorithm finally solves the GP.

C. Simulations

The performance of the algorithm described in the previous subsections is now illustrated in a number of numerical experiments. The intervals of admissible heading variations are set to ±15° in order to develop the posynomial approximations of Sec. III.B. GPs are solved in MATLAB using the convex modeling package CVX[§] [29].

Consider first a pure conflict resolution problem, i.e., with no metering constraints, where aircraft are initially randomly distributed in a 300×300 n mile square, with random initial headings. Figure 10 shows two instances of the problem, involving 15 and 20 aircraft, respectively. Any two aircraft are initially separated by at

[§] Data available online at http://cvxr.com/cvx [retrieved 30 November 2010].

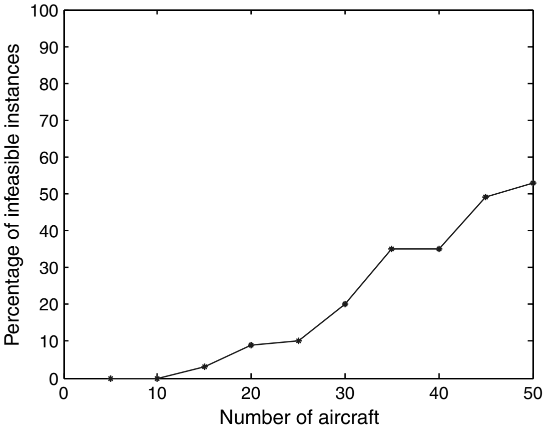

Fig. 11 Percentage of infeasible instances for the heuristic in the conflict resolution scenario of Fig. 10, as the number of aircraft increases in the fixed region; 100 instances were generated in each case.

least 30 n mile in the generated instances. The initial speeds are uniformly and randomly distributed over the interval [180 kt, 300 kt], and these speeds are also the minimum and maximum allowed speeds for all aircraft. The cost function optimized penalizes deviations with respect to the initial speeds and headings:

$$\min_{v_i,\psi_i,i=1,\ldots,N} \sum_{i=1}^{N} \max\left\{\frac{v_{i,0}}{v_i},\frac{v_i}{v_{i,0}}\right\} + \rho \max\left\{\frac{\psi_{i,0}}{\psi_i},\frac{\psi_i}{\psi_{i,0}}\right\} \qquad (37)$$

where the parameter ρ controls the penalization of heading changes with respect to speed changes. Initial and final trajectories (i.e., after conflict resolution) are represented in Fig. 10 for the two instances of the problem. Recall that for such a conflict resolution problem, there is always at least one feasible solution, e.g., the trivial one that simply aligns all the headings. However, the algorithm does not guarantee that it will find a feasible solution, since it only locally optimizes the headings and, moreover, does not optimize over the crossing patterns. The histogram in Fig. 11 shows empirically how the number of instances found to be infeasible in simulations grows with the density of planes considered.

Next let us consider problems involving both conflict resolution and metering. Figure 12 shows a scenario with 14 aircraft with crossing trajectories, initially involved in seven conflicts. In addition, half of the aircraft are heading toward a metered zone, with the constraint that two successive aircraft entering the zone must be separated by at least 6 min. The cost function minimized is again Eq. (37). The algorithm successfully determines how to slow down the aircraft of the metered flow while removing the conflicts between aircraft. Note that the ordering of the aircraft crossing the metered line is fixed a priori here. Optimizing over this ordering would require introducing integer variables as in Sec. IV.

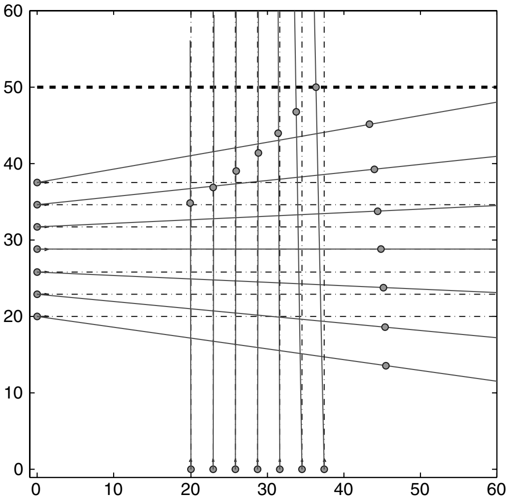

Fig. 12 Crossing flows with 14 aircraft and a metered boundary. One unit represents 10 n mile. The initial speed of all aircraft is the same, creating seven conflicts. Moreover, two successive aircraft crossing the thick dashed line must be separated by at least 6 min. The positions of the aircraft are shown at $t = 0$ and at the time where the first aircraft traveling north crosses the metered boundary.

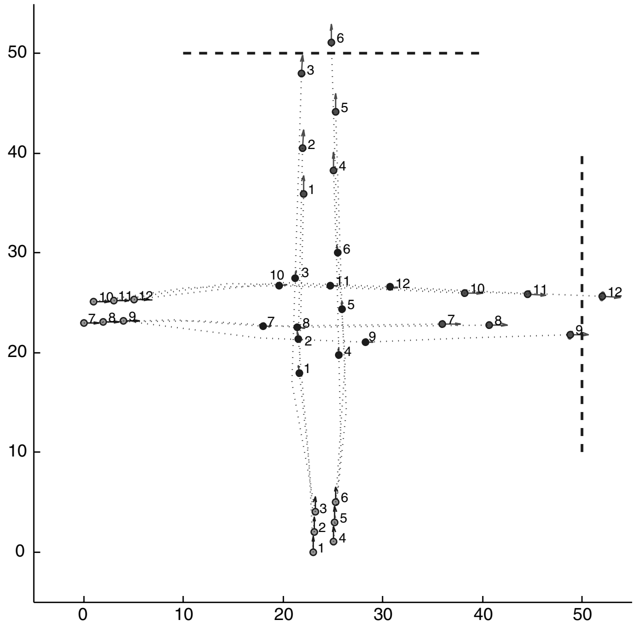

Fig. 13 Crossing flows with 12 aircraft and two metered boundaries. One unit represents 10 n mile. The initial configuration is symmetric with respect to the diagonal, producing six conflicts. The two dashed lines are metered boundaries, each with a minutes-in-trail restriction MINIT = 6 min between successive aircraft crossing it. The order in which the aircraft cross each boundary is fixed a priori. In contrast to Fig. 12, the headings and velocities are reoptimized every 30 min, resulting in piecewise-linear trajectories. The positions of the aircraft are shown for three snapshots, at $t = 0$, $t = 1$ h, and $t = 2$ h. The disks represent the safety regions around the aircraft and are not allowed to overlap. At each optimization step, the aircraft 1–6 optimize their headings locally around the headings directed toward the centered waypoint [23.5, 100], and symmetrically for aircraft 7–12 and the waypoint [100, 23.5]. This allows the aircraft to correct their trajectories and minimize deviations once the conflicts are resolved.

The simulations above only show the results of a single stage optimization. Note, however, that solving the GPs of moderate size arising in typical conflict resolution scenarios can be done in real time. Hence, the optimization procedure can be repeated at regular intervals to continuously adjust the trajectories based on the current aircraft positions, resulting in piecewise-linear trajectories. This approach also guarantees that the aircraft fly straight trajectories that are conflict-free if the algorithm stops updating the speed and heading variables. Figure 13 shows an example illustrating this periodic optimization procedure. At each optimization step, the headings $\{\psi_{i,0}\}_{1\leq i\leq N}$ used for the local optimization are the desired headings starting from the current location. A comparison of Fig. 12 and 13 shows that this procedure significantly reduces the deviations necessary to resolve conflicts.

VI. Conclusions

This paper introduced a novel conflict resolution method based on geometric programming. This method can be used to simultaneously optimize over the speeds and headings of many aircraft, assuming only small heading changes. A key feature of the approach is the ability to include metering constraints in the trajectory optimization problems. Hence, geometric programming is a useful tool to model and solve problems at the interface between traffic flow management and separation assurance. Solving these problems will allow a smoother integration of automated decision support tools in air traffic control.

Acknowledgment

This work was supported by the Office of Naval Research (ONR) Multidisciplinary University Research initiative (MURI) award N00014-08-1-0696.

References

[1] Nolan, M. S., *Fundamentals of Air Traffic Control*, 5th ed., Delmar Cengage Learning, Clifton Park, NY, 2010, Chaps. 3, 9, 10.

[2] Haraldsdottir, A., Schwab, R. W., Shakarian, A., Wood, G., and Krishnamachari, R. S., "ATM Operational Concepts and Technical Performance Requirements," *New Concepts and Methods in Air Traffic Management*, edited by L. Bianco, P. Dell'Olmo, and A. R. Odoni, Springer, New York, 2001, pp. 63–74.

[3] Robinson, M., DeLaura, R., Martin, B., Evans, J. E., and Weber, M. E., "Initial Studies of an Objective Model to Forecast Achievable Airspace Flow Program Throughput from Current and Forecast Weather Information," *Aviation, Range and Aerospace Meteorology Special Symposium on Weather-Air Traffic Management Integration*, Phoenix, AZ, Jan. 2009.

[4] Ostwald, P., Topiwala, T., and DeArmon, J., "The Miles-in-Trail Impact Assessment Capability," 6th AIAA Aviation Technology, Integration and Operations Conference (ATIO), AIAA Paper 2006-7780, Wichita, KS, Sept. 2006.

[5] Mueller, K. T., Schleicher, D. R., and Bilimoria, K. D., "Conflict Detection and Resolution with Traffic Flow Constraints," AIAA Guidance, Navigation, and Control Conference, AIAA Paper 2002-4445, Monterey, CA, Aug. 2002.

[6] Niedringhaus, W. P., "Stream Option Manager (SOM): Automated Integration of Aircraft Separation, Merging, Stream Management, and Other Air Traffic Control Functions," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No. 9, Sept. 1995, pp. 1269–1280. doi:10.1109/21.400505

[7] Kuchar, J. K., and Yang, L. C., "A Review of Conflict Detection and Resolution Modeling Methods," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, Dec. 2000, pp. 179–189. doi:10.1109/6979.898217

[8] Schild, R., "Rule Optimization for Airborne Aircraft Separation," Ph.D. Thesis, Department of Econometry, Operations Research and System Theory, Technical University of Vienna, Vienna, 1998.

[9] Hoekstra, J., van Gent, R., and Ruigrok, R., "Conceptual Design of Free Flight with Airborne Separation Assurance," AIAA Guidance, Navigation, and Control Conference, AIAA Paper 1998-4239, Boston, Aug. 1998, pp. 807–817.

[10] Hwang, I., Kim, J., and Tomlin, C., "Protocol-Based Conflict Resolution for Air Traffic Control," *Air Traffic Control Quarterly*, Vol. 15, No. 1, 2007, pp. 1–34.

[11] Zeghal, K., "A Review of Different Approaches Based on Force Fields for Airborne Conflict Resolution," AIAA Guidance, Navigation, and Control Conference, AIAA Paper 1998-4240, Boston, Aug. 1998, pp. 818–827.

[12] Durand, N., Alliot, J. M., and Chansou, O., "An Optimizing Conflict Solver for Air Traffic Control," *Air Traffic Control Quarterly*, Vol. 3, No. 3, 1995, pp. 139–161.

[13] Frazzoli, E., Mao, Z.-H., Oh, J.-H., and Feron, E., "Resolution of Conflicts Involving Many Aircraft via Semidefinite Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 1, Jan.–Feb. 2001, pp. 79–86. doi:10.2514/2.4678

[14] Pallottino, L., Feron, E. M., and Bicchi, A., "Conflict Resolution Problems for Air Traffic Management Systems Solved with Mixed Integer Programming," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 3, No. 1, March 2002, pp. 3–11. doi:10.1109/6979.994791

[15] Inalhan, G., Stipanovic, D., and Tomlin, C., "Decentralized Optimization, with application to Multiple Aircraft Coordination," *Proceedings of the 41st IEEE Conference on Decision and Control*, Vol. 1, IEEE, Piscataway, NJ, Dec. 2002, pp. 1147–1155. doi:10.1109/CDC.2002.1184667.

[16] Hu, J., Prandini, M., and Sastry, S., "Optimal Coordinated Maneuvers for Three Dimensional Aircraft Conflict Resolution," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 5, Sept.–Oct. 2002, pp. 888–900. doi:10.2514/2.4982

[17] Menon, P. K., and Sweriduk, G. D., "Optimal Strategies for Free-Flight Air Traffic Conflict Resolution," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 2, 1999, pp. 202–211. doi:10.2514/2.4384

[18] Raghunathan, A. U., Gopal, V., Subramanian, D., Biegler, L. T., and Samad, T., "Dynamic Optimization Strategies for Three-Dimensional Conflict Resolution of Multiple Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 4, July–Aug. 2004, pp. 586–594. doi:10.2514/1.11168

[19] Bilimoria, K. D., "A Geometric Optimization Approach to Aircraft Conflict Resolution," AIAA Guidance, Navigation, and Control Conference, AIAA Paper 2000-4265, Denver, CO, Aug. 2000.

[20] Tomlin, C., Pappas, G. J., and Sastry, S., "Conflict Resolution for Air Traffic Management: A Study in Multiagent Hybrid Systems," *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, April 1998, pp. 509–521. doi:10.1109/9.664154

[21] Schouwenaars, T., How, J., and Feron, E., "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees," AIAA Guidance, Navigation, and Control Conference, AIAA Paper 2004-5141, Providence, RI, Aug. 2004.

[22] Pallottino, L., Scordio, V. G., Frazzoli, E., and Bicchi, A., "Decentralized Cooperative Policy for Conflict Resolution in Multi-vehicle Systems," *IEEE Transactions on Robotics and Automation*, Vol. 23, No. 6, Dec. 2007, pp. 1170–1183. doi:10.1109/TRO.2007.909810.

[23] Bilimoria, K. D., and Lee, H. Q., "Aircraft Conflict Resolution with an Arrival Time Constraint," AIAA Guidance, Navigation, and Control Conference, AIAA Paper 2002-4444, Monterey, CA, Aug. 2002.

[24] Dugail, D., Feron, E., and Bilimoria, K. D., "Conflict-Free Conformance to En Route Flow Rate Constraints," AIAA Guidance, Navigation, and Control Conference, AIAA Paper 2002-5013, Monterey, CA, Aug. 2002.

[25] Duffin, R. J., Peterson, E. L., and Zener, C. M., *Geometric Programming*, Wiley, New York, 1967.

[26] Kortanek, K. O., Xu, X., and Ye, Y., "An Infeasible Interior-Point Algorithm for Solving Primal and Dual Geometric Programs," *Mathematical Programming*, Vol. 76, Nos. 1/2, Jan. 1997, pp. 155–182. doi:10.1007/BF02614382.

[27] Boyd, S., Kim, S.-J., Vandenberghe, L., and Hassibi, A., "A Tutorial on Geometric Programming," *Optimization and Engineering*, Vol. 8, No. 1, March 2007, pp. 67–127. doi:10.1007/s11081-007-9001-7

[28] Hillier, F. S., and Lieberman, G. J., *Introduction to Operations Research*, 7th ed., McGraw-Hill, New York, 2001, Chap. 12.

[29] Grant, M., and Boyd, S., "Graph Implementations for Nonsmooth Convex Programs," *Recent Advances in Learning and Control*, edited by V. Blondel, S. Boyd, and H. Kimura, Lecture Notes in Control and Information Sciences, Springer–Verlag, Berlin, 2008, pp. 95–110.