11-28-2010

Joint Metering and Conflict Resolution in Air Traffic Control

Jerome Le Ny
University of Pennsylvania, jerome.le-ny@polymtl.ca

George J. Pappas
University of Pennsylvania, pappasg@seas.upenn.edu

Joint Metering and Conflict Resolution in Air Traffic Control

Abstract
This paper describes a novel optimization-based approach to conflict resolution in air traffic control, based on geometric programming. The main advantage of the approach is that Geometric Programs (GPs) can also capture various metering directives issued by the traffic flow management level, in contrast to most recent methods focusing purely on aircraft separation issues. GPs can also 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. Additional integer variables can be introduced to improve the quality of the obtained solutions and handle combinatorial choices, resulting in Mixed-Integer Geometric Programs (MIGPs). We present GPs and MIGPs to solve a variety of joint metering and separation scenarios, e.g. including miles-in-trail and minutes-in-trail restrictions through airspace fixes and boundaries. Simulation results demonstrate the efficiency of the approach.

Keywords
air traffic control, conflict resolution, metering, geometric programming, optimization, path planning

Disciplines
Multi-Vehicle Systems and Air Traffic Control

Joint Metering and Conflict Resolution in Air Traffic Control

Jerome Le Ny[1] and George J. Pappas[2]

University of Pennsylvania, Philadelphia, PA 19104, USA

This paper describes a novel optimization-based approach to conflict resolution in air traffic control, based on geometric programming. The main advantage of the approach is that Geometric Programs (GPs) can also capture various metering directives issued by the traffic flow management level, in contrast to most recent methods focusing purely on aircraft separation issues. GPs can also 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. Additional integer variables can be introduced to improve the quality of the obtained solutions and handle combinatorial choices, resulting in Mixed-Integer Geometric Programs (MIGPs). We present GPs and MIGPs to solve a variety of joint metering and separation scenarios, e.g. including miles-in-trail and minutes-in-trail restrictions through airspace fixes and boundaries. Simulation results demonstrate the efficiency of the approach.

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 any pair of aircraft, typically five nautical miles of horizontal separation and a thousand feet of vertical separation. Besides 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 [1, 2]. For example, handoffs between sectors [3] and traffic flows entering a terminal area or a flow constrained area during an airspace flow program [4] can all be limited. A typical way of enforcing such rate limitations is by using miles-in-trail or minutes-in-trail restrictions at certain airspace fixes [2, 5]. The resulting path-planning problems then lie at the interface between traffic flow management and separation assurance [6]. With the exception of the Stream-Option-Manager of Niedringhaus [7] however, there is little research that integrates in the same framework aircraft separation and metering constraints in order to support the ATCs' task. Hence in this paper we consider the problem of resolving conflicts and maintaining separation between airborne aircraft, in the presence of additional metering constraints. We assume that the configurations and velocities of all aircraft involved in a particular conflict are known at a central computation facility, where the resulting path-planning problem is solved. The environment is two-dimensional, as in most of the existing literature [7–10]. In fact, altitude changes are typically avoided for conflict resolution (except for imminent conflicts), in order to minimize passenger discomfort and maintain compatibility with the vertically layered structure of the airspace [2].

Numerous conflict resolution methods have been previously proposed, and many of them are discussed in the survey of Kuchar and Yang [11]. These methods can roughly be divided into rule-

---

[1] Postdoctoral Researcher, Department of Electrical Engineering. Levine Hall 465, 3330 Walnut Street, Philadelphia, PA 19104. AIAA Member.

[2] Professor, Department of Electrical Engineering. Levine Hall 460, 3330 Walnut Street, Philadelphia, PA 19104.

based approaches [12–14], force field methods [15], and optimization-based approaches [7, 8, 10, 16]. This paper concentrates on the last approach, and adopts many of the simplifying assumptions made in previous research. In particular, our aircraft models are kinematic models (see Section II A), and we focus on simple ATC directives involving speed and heading changes that are assumed to occur instantaneously. Some previous work uses more detailed dynamic models, at the expense of more complex computations [17, 18]. Our method is limited to small heading changes (with the allowed interval of heading changes a chosen parameter, e.g. $\pm 15^\circ$), which is also the case in some previous references [7, 8], and can be reasonably expected in usual conditions. The simulations in Pallottino et al. [10] for example all result in such small heading changes, even though no constraint is imposed a priori. Under assumptions similar to ours, Niedringhaus [7] proposes a general modeling framework for conflict resolution based on linear programming. Subsequent studies include optimization over possible crossing patterns, using genetic algorithms [16], semidefinite-programming based relaxations [8], or mixed-integer linear programming [10]. 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. [20] 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-legged trajectories. Hwang et al. [14] propose a protocol based conflict resolution scheme where conflict resolution maneuvers consist of two straight paths of equal length.

The research efforts described in the previous paragraph are only concerned with the aircraft separation and conflict resolution problem. As we noted previously, aircraft trajectories are also often constrained by traffic flow management directives [2], a problem that has received much less attention. Niedringhaus [7] includes a scenario involving merging aircraft into a metered stream, similar to the problem described in Section IV. Bilimoria and Lee [21] and Mueller et al. [6] extend the previously mentioned two-aircraft conflict resolution algorithm [19] to include arrival time and metering constraints. Dugail et al. [22] control aircraft velocities and path lengths to accommodate an output flow rate restriction in an en-route sector, in order to describe the dynamics of upstream propagation of sector flow constraints. In this paper, we show that geometric programming [23] is an efficient optimization framework to handle similar problems involving both separation and metering constraints for scenarios with multiple aircraft. Specifically, we show how these often nonlinear constraints can be expressed or conservatively approximated by posynomial constraints [23] in the control variables (velocity and heading). 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 [24, 25]. As in some previous work [10], we can also model combinatorial choices, for example to optimize over the set of crossing patterns [8, 10] or arrival orderings at a fix, by including integer variables and solving a Mixed-Integer Geometric Program (MIGP). MIGP solvers are less mature than Mixed-Integer Linear Program (MILP) solvers however, and so one cannot currently obtain the same short computational times for MIGP as those obtained using MILP [10]. On the other hand, even for standard conflict resolution problems, the nonlinearities allowed in a GP let us jointly optimize over headings and velocities, whereas the linear programming approach in Pallottino et al. [10] is restricted to optimizing over either velocities 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 presents the problem formulation, and describes the constraints of the optimization problem that result from the safety conditions (aircraft separation) and metering specifications. In Section III we provide some background material on posynomials and geometric programming, and we explain how to reformulate or approximate the constraints as posynomial constraints. Section IV discusses a specific scenario involving merging aircraft at a metered intersection, and the use of mixed-integer geometric programming to optimize over the aircraft ordering at the fix as well. In Section V, we present a fast but local optimization heuristic based on geometric programming. Simulation results show that complex problem instances involving many aircraft and metered boundaries can be solved very efficiently. Finally, we conclude in Section VI.

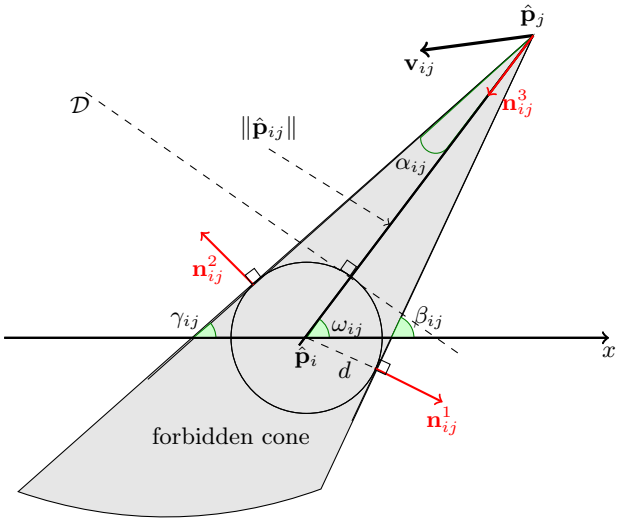

Fig. 1 Geometric representation of the forbidden cone of velocities.

II. Problem Formulation

A. Air Traffic Model

We 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 $v_{i,0} \in \mathbb{R}_{>0}, i = 1, \ldots, N$, where $\psi_{i,0}$ denotes the initial heading of aircraft i and $\hat{\mathbf{p}}_i = (x_{i,0}, y_{i,0})$ its initial position. We adopt a kinematic aircraft model, as in much of the previous work on conflict resolution [7, 8, 10, 20]. Hence aircraft can change their heading and speeds instantaneously. The mandatory separation distance to maintain between any two aircraft at all times is denoted d (currently d is set to 5 nmi for en-route traffic). We consider the problem of specifying new headings and speeds ψ_i and v_i so that no conflict, i.e. loss of separation, occurs if all aircraft follow the resulting straight line trajectories. Hence a single maneuver is executed at time $t = 0$ by all aircraft in order to enforce separation, as in e.g. references [8, 10]. If the computation of these maneuvers is fast enough however, it can be executed at regular intervals to obtain a closed-loop control policy [8, 10] and the resulting trajectories are then piecewise linear, see Section V C. We modify the decision variables from $(\psi_{i,0}, v_{i,0})$ to $(\psi_i, v_i) \in \mathcal{S}^1 \times \mathbb{R}_{>0}$ in order to minimize a given objective function. For example, we might want to minimize deviations with respect to the initial aircraft trajectories. In addition to enforcing separation between aircraft, the ATC might also need to impose additional metering constraints on the trajectories crossing certain boundaries and waypoints, as discussed in section 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 translates into constraints imposed on the allowed velocities. Consider two aircraft i, j with initial positions $\hat{\mathbf{p}}_i, \hat{\mathbf{p}}_j$, current positions $\mathbf{p}_i, \mathbf{p}_j$, and velocity vectors set to $\mathbf{v}_i, \mathbf{v}_j$, see Fig. 1. Note that throughout the paper, we denote vector quantities in boldface. We study the problem in the mobile frame centered at $\mathbf{p}_i$. In this frame, aircraft j has relative velocity $\mathbf{v}_{ij} = \mathbf{v}_j - \mathbf{v}_i$ and aircraft i is immobile. We first impose a safety condition considered in previous work [8, 10], 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 we have the option of modifying these parameters again at a later time.

No conflict arises if the distance from $\hat{\mathbf{p}}_i$ to the half-line $\hat{\mathbf{p}}_j + \mathbb{R}_+ \mathbf{v}_{ij}$ describing the trajectory of j in the moving frame is at least d. Geometrically, this sufficient conditon 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 disc centered at $\hat{\mathbf{p}}_i$ and of radius d, see Fig. 1. This constraint has been handled in previous work

in at least two ways. Frazzoli et al. [8] represent it by a non-convex quadratic inequality on $\|\mathbf{v}_{ij}\|$. This non-convex constraint is then relaxed to obtain a semidefinite program, the solution of which serves to design a feasible solution using randomized rounding. We follow an alternative approach, as in e.g. Pallottino et al. [10], which consists in representing the forbidden cone as the union of two half-spaces defined by two normal vectors $\mathbf{n}_{ij}^1$ and $\mathbf{n}_{ij}^2$ such that the admissible relative velocities satisfy

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

where $\vee$ denotes a disjunction (logical "or"). We now write these separation constraints (1) in coordinates. Let $\alpha_{ij} = \arcsin(d/\|\hat{\mathbf{p}}_{ij}\|)$ (then $\alpha_{ij} \in (0, \pi/2]$), and define with respect to a global coordinate system $\omega_{ij} = \arg(\hat{\mathbf{p}}_{ij})$, $\beta_{ij} = \omega_{ij} + \alpha_{ij}$ and $\gamma_{ij} = \omega_{ij} - \alpha_{ij}$, see Fig. 1. Then we have

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

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

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

For the second constraint in (1), we get similarly

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

After choosing a coordinate system (Oxy) on the plane, we order the aircraft by increasing x coordinate. Then for any two aircraft i and 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 on Fig. 2. For simplicity, we now adopt the following notation. We let $s_i^{\beta_{ij}} = \sin(\psi_i - \beta_{ij})$, $s_{i,0}^{\beta_{ij}} = \sin(\psi_{i,0} - \beta_{ij})$, and define similarly $s_i^{\gamma_{ij}}, s_{i,0}^{\gamma_{ij}}, s_j^{\beta_{ij}}, s_{j,0}^{\beta_{ij}}, s_j^{\gamma_{ij}}, s_{j,0}^{\gamma_{ij}}$. If either of the constraints (2) or (3) is satisfied , aircraft i and j are not in conflict. For aircraft that are in conflict, we have the following proposition.

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 (2) holds for all v_i, v_j.*
2. *If $s_{j,0}^{\beta_{ij}} \leq 0$, then $s_{i,0}^{\beta_{ij}} < 0$. Otherwise (2) holds for all v_i, v_j.*
3. *If $s_{i,0}^{\gamma_{ij}} \leq 0$, then $s_{j,0}^{\gamma_{ij}} < 0$. Otherwise (3) holds for all v_i, v_j.*
4. *If $s_{j,0}^{\gamma_{ij}} \geq 0$, then $s_{i,0}^{\gamma_{ij}} > 0$. Otherwise (3) holds for all v_i, v_j.*

For reference, we also record the initial heading values for which we cannot enforce (2) or (3) by changing v_i, v_j only, i.e. for which both constraints remain infeasible for all positive values of v_i, v_j. Define

$$\begin{aligned} \mathcal{C}_{ij}^0 = \Big\{(\psi_{i,0}, \psi_{j,0}) \Big| \quad & \Big[(s_{i,0}^{\beta_{ij}} < 0 \wedge s_{j,0}^{\beta_{ij}} \geq 0) \vee (s_{i,0}^{\beta_{ij}} = 0 \wedge s_{j,0}^{\beta_{ij}} > 0)\Big] \\ & \wedge \Big[(s_{j,0}^{\gamma_{ij}} < 0 \wedge s_{i,0}^{\gamma_{ij}} \geq 0) \vee (s_{j,0}^{\gamma_{ij}} = 0 \wedge s_{i,0}^{\gamma_{ij}} > 0)\Big]\Big\}. \end{aligned} \tag{4}$$

Then if the initial headings of aircraft i, 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.

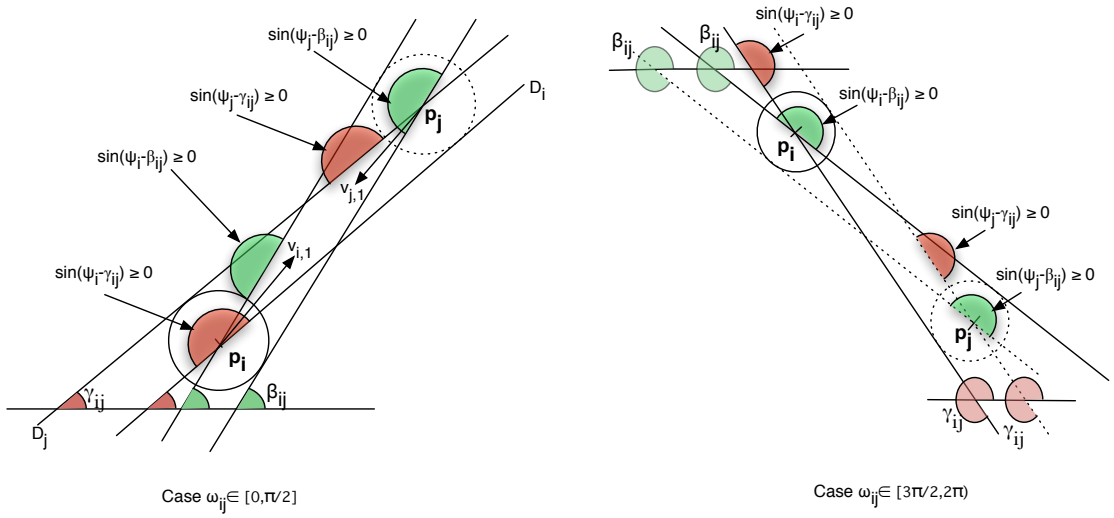

Fig. 2 Geometry of the constraints (2), (3). The circles around each aircraft position have radius d equal to the mandatory separation distance. The definition of the set $\mathcal{C}_{ij}^0$ is illustrated on the left figure for $v_i = v_{i,1}$ and $v_j = v_{j,1}$.

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 following another aircraft and moving in the same direction to catch up with the preceding aircraft, because eventually a conflict would arise. This condition is too conservative for certain finite-horizon problems considered in this paper. In the merging and metering scenario discussed in Section IV, separation between two aircraft must be ensured only until one them reaches an airspace fix, since we assume that safety is guaranteed beyond it, typically by a new ATC command. In this case, we wish to allow for two aircraft following each other to reduce their separation, potentially up to the minimum imposed by the metering restriction. Hence we consider now 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. We describe a *sufficient* condition ensuring that separation is maintained until aircraft i reaches its waypoint. The time t_i it takes for aircraft i to reach the waypoint is $t_i = d_i/v_i$. Referring to Fig. 1, safety is maintained if aircraft i reaches the waypoint before j enters the circle of radius d around i. Note on Fig. 1 the dashed line $\mathcal{D}$ perpendicular to $\hat{\mathbf{p}}_{ij} = \hat{\mathbf{p}}_j - \hat{\mathbf{p}}_i$ tangent to the circle around aircraft i, which separates the plane into two half-planes, each containing one aircraft. The sufficient condition that we consider 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 it ignores the fact that safety could also 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 $(d\,\hat{\mathbf{p}}_{ij})/\|\hat{\mathbf{p}}_{ij}\|$, the safety condition can be written

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

Let us define the unit vector $\mathbf{n}_{ij}^3 = -\frac{\hat{\mathbf{p}}_{ij}}{\|\hat{\mathbf{p}}_{ij}\|} = [-\cos\omega_{ij}, -\sin\omega_{ij}]^T$ (see Fig. 1). This expression can then be rewritten

$$\frac{d_i}{v_i}\left\langle \mathbf{v}_{ij}, \mathbf{n}_{ij}^3 \right\rangle \leq \|\hat{\mathbf{p}}_{ij}\| - d. \tag{5}$$

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

$$\frac{d_i}{v_i}[-(v_j\cos\psi_j - v_i\cos\psi_{i,0})\cos\omega_{ij} - (v_j\sin\psi_j - v_i\sin\psi_{i,0})\sin\omega_{ij}] \le \|\hat{\mathbf{p}}_{ij}\| - d$$
$$v_i(\|\hat{\mathbf{p}}_{ij}\| - d - d_i\cos\psi_{i,0}\cos\omega_{ij} - d_i\sin\psi_{i,0}\sin\omega_{ij}) + v_j d_i(\cos\psi_j\cos\omega_{ij} + \sin\psi_j\sin\omega_{ij}) \ge 0$$
i.e.,
$$v_i(\|\hat{\mathbf{p}}_{ij}\| - d - d_i\cos(\psi_{i,0} - \omega_{ij})) + v_j d_i\cos(\psi_j - \omega_{ij}) \ge 0. \quad (6)$$

If instead we were considering safety until j reaches a waypoint at distance d_j in direction $\psi_j = \psi_{j,0}$, then the constraint would be

$$\frac{d_j}{v_j}[-(v_j\cos\psi_{j,0} - v_i\cos\psi_i)\cos\omega_{ij} - (v_j\sin\psi_{j,0} - v_i\sin\psi_i)\sin\omega_{ij}] \le \|\hat{\mathbf{p}}_{ij}\| - d$$
$$- v_i d_j(\cos\psi_i\cos\omega_{ij} + \sin\psi_i\sin\omega_{ij}) + v_j(\|\hat{\mathbf{p}}_{ij}\| - d + d_j\cos\psi_{j,0}\cos\omega_{ij} + d_j\sin\psi_{j,0}\sin\omega_{ij}) \ge 0$$
i.e.,
$$- v_i d_j\cos(\psi_i - \omega_{ij}) + v_j(\|\hat{\mathbf{p}}_{ij}\| - d + d_j\cos(\psi_{j,0} - \omega_{ij})) \ge 0. \quad (7)$$

C. Metering Constraints

Besides the separation constraints, constraints on aircraft trajectories often arise from metering at certain points (airspace fixes) or boundaries of the airspace. We cal these constraints *metering constraints*.

1. Distance Metering

Miles-in-trail restrictions are a commonly enforced metering constraint, where the ATC must maintain a specified distance, denoted hereafter MIT, between aircraft in a single stream. Consider the situation depicted on Fig. 3, with several aircraft converging towards a metering fix where a miles-in-trail restriction MIT must be enforced in the downstream flow. Suppose aircraft j is to be inserted after aircraft i in the 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. Then if at $t = 0$ 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

$$\frac{d_i}{v_i} + \frac{\mathsf{MIT}}{v_0} \le \frac{d_j}{v_j}. \quad (8)$$

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 general it is nonlinear.

2. Time Metering

Instead of specifying a separation between successive aircraft in a stream by a distance MIT, we 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 $\mathsf{MINIT} = 60/x$. In the following we use the notation MINIT 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, d_j of their respective waypoints and subject to a minutes-in-trail restriction on the times at which they reach these waypoints, we must have

$$\frac{d_i}{v_i} + \mathsf{MINIT} \le \frac{d_j}{v_j}. \quad (9)$$

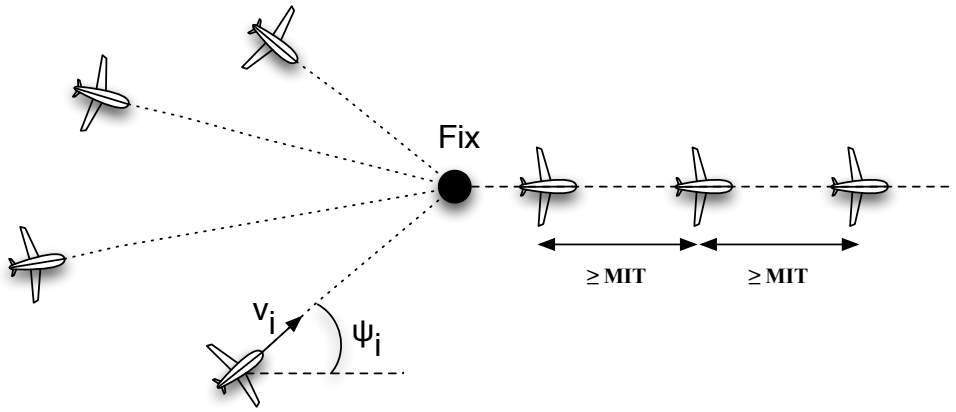

Fig. 3 Merging and metered traffic flows.

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

For the case of a flow constrained boundary, we 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 fixed coordinate system, see Fig. 4. Aircraft i is at distance D_i from this line. Then if aircraft i is to cross this boundary, we must have $\delta < \psi_i < \delta + \pi [\text{mod } 2\pi]$ or $\delta + \pi < \psi_i < \delta + 2\pi [\text{mod } 2\pi]$. In the first case, we have $\sin(\psi_i - \delta) > 0$ and

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

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

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

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

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

or

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

If equation (11) holds instead for both aircraft, we get similarly the constraint

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

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

D. Optimization-Based Conflict Resolution

In the previous sections, we have described separation and metering constraints that restrict the set of allowed velocities and headings of the aircraft. In the rest of the paper, we adopt an optimization-based approach to aircraft trajectory planning, where we wish to minimize a cost function depending on the $2N$ variables $v_i, \psi_i, i = 1, \ldots, N$, subject to the separation and metering constraints. See (25)-(28) below for an example of such an optimization problem for metering aircraft

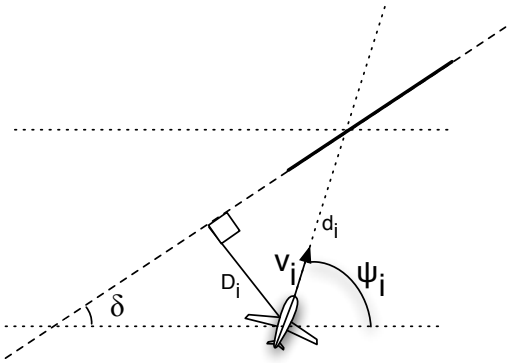

Fig. 4 Flow constrained boundary. Case $\delta < \psi < \delta + \pi$.

at a fix. The cost function is introduced to try to keep the trajectories close to the preferred ones while satisfying the various constraints, for example to minimize deviation with respect to the initial trajectories. We need however an efficient optimization framework to handle the nonlinearities in the constraint. One potential approach is to use standard nonlinear programming tools, but this results in long computation times that are not compatible with the real-time aspect of the conflict resolution problem. Instead, we show in the following section that the constraints can be handled directly or approximately by fast geometric programming solvers.

III. Geometric Programming

In this section we first provide some background material on geometric programming, a class of efficiently solvable optimization problems that we rely on to design conflict-free trajectories. Then, we develop in Subsection III B certain posynomial approximations for the nonlinearities appearing in the separation and metering constraints of Section II. We provide conservative approximations in order to guarantee safety.

A. Background Material

The separation constraints (2), (3), (6), (7) are linear in the speed variables but nonlinear in the heading variables, and the metering constraints (8), (9), (12), (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 ordering as in the scheduling scenario of Section 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. In this paper, we use geometric programming [23] to capture exactly the nonlinearities in the speed variables, and approximately the nonlinearities in the heading variables. First, we review the basic terminology of geometric programming [23, 25].

A *monomial* is a function $f : \mathbb{R}^n_{>0} \to \mathbb{R}$ 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 $g(x) \leq 1$ with g a posynomial, as well as $h(x) = 1$ with h a

monomial, are called *geometric programs* (GP), and can be solved almost as efficiently as linear programs by interior-point methods [24, 25]. Indeed the change of variable

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

which is valid since we require that the variables x_i be positive, converts the geometric program 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 [25]. Replacing posynomials by generalized posynomials in the definition of a geometric program, we obtain a *generalized geometric program*, which can be solved efficiently as well by converting it to a GP.

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

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

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), (13) are posynomial constraints in the speed variables. This is true for (12), (13) by noting that for the range of relevant headings the coefficients involving sine functions are positive, see (10), (11). In order to rewrite the separation constraints (2), (3), (6), (7) as posynomial constraints, we use the following lemma.

Lemma 2 *Any linear constraint in the decision variables $x > 0, y > 0$, 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 The last case is when α and β have opposite signs and are both non-zero. If $\alpha > 0, \beta < 0$, then we can rewrite the inequality as $\frac{|\beta|}{\alpha} \frac{y}{x} \leq 1$. If $\alpha < 0, \beta > 0$, we rewrite the constraint as $\frac{|\alpha|}{\beta} \frac{x}{y} \leq 1$.

Hence we see that by a simple preprocessing of the linear inequalities of the form (16), we can handle such inequalities by geometric programming. With the headings fixed, all separation constraints (2), (3), (6), (7) are constraint on the speed variables of the form (16).

GP modeling also allows us to work directly with time to arrivals (or delays) in the objective function, which are important metrics for ATCs [2]. For example, suppose aircraft i is cleared from the considered airspace when it reaches a waypoint at distance d_i. We might want to optimize, subject to the separation and metering constraints, the total clearing time for all aircraft

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

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 we would like to minimize the deviation with respect to the initial 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 $v_i = v_{i,0}, i = 1, \ldots, N$.

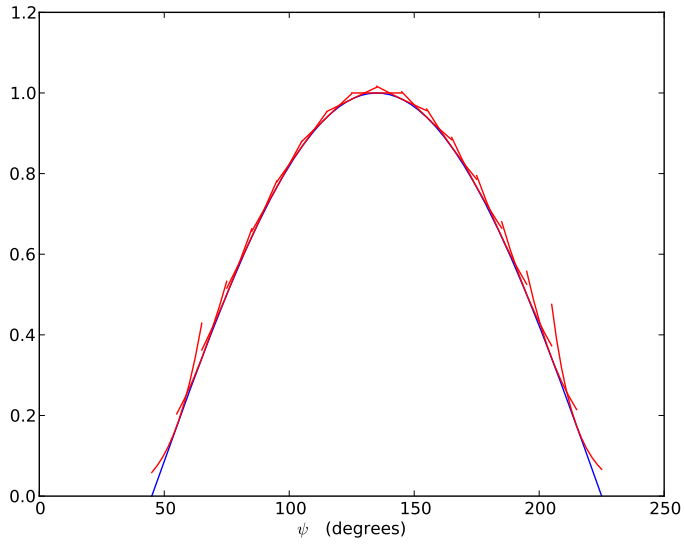

Fig. 5 Local monomial overapproximations (in red) of $\psi \mapsto \sin(\psi - \pi/4)$, for $\pi/4 < \psi < \pi/4 + \pi$ (in blue). For $\psi_0 = \pi/4 + 10^\circ, \pi/4 + 20^\circ, \ldots$, the approximation (19) is ploted for $\psi \in [\psi_0 - 10^\circ, \psi_0 + 10^\circ]$. Clearly, the quality of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

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, we 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 developed in the neighborhood of a given heading for each aircraft, e.g. $\pm 15^\circ$ around the initial heading $\psi_{i,0}$ if the objective is to minimize trajectory deviations. Although one can choose the amplitude of the allowed heading deviations, the approximations become overly conservative for large intervals.

Approximating the separation constraints (2), (3) and metering constraints (12), (13) requires that we develop posynomial approximations of the functions $\psi \to \sin(\psi - \beta)$ and $\psi \to 1/\sin(\psi - \beta)$ for $\psi \in (\beta, \beta + \pi)$. In order to work with positive variables, which is required in GPs, we express all angles as positive values, shifting them by a multiple of 2π if necessary. Hence an interval such as $-\frac{\pi}{3} \leq \psi \leq \frac{2\pi}{3}$, is expressed instead as $\frac{5\pi}{3} \leq \psi \leq \frac{8\pi}{3}$. 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 we do not present approximations of the finite-horizon separation constraints (6), (7) in this paper, although a similar approach could be used in this case as well. These constraints (6), (7) are only used in the merging and metering scenario described in section IV, where we optimize over the speed variables only.

The most difficult of the two functions sin and $1/\sin$ to approximate is the sine function. We use the following simple monomial approximation [25] around $\psi = \psi_0$, which turns out to an upper bound due to the concavity property of sin over the relevant interval:

$$\text{for } \psi, \psi_0 \in (\beta, \beta + \pi), \;\; \sin(\psi - \beta) \leq c \left(\frac{\psi}{\psi_0} \right)^a, \;\; \text{with } c = \sin(\psi_0 - \beta), \; a = \psi_0 \cot(\psi_0 - \beta). \quad (19)$$

We refer the reader to Boyd et al. [25] for techniques for developing such monomial approximations. Fig. 5 shows the behavior of the approximation, for angular deviations of $\pm 10^\circ$ 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.

We 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 $G > 0$, we use again a simple local monomial approximation [25], shifted upwards in order to obtain an upper bound on an interval $[\psi_0 - l, \psi_0 + r] \subset (\beta, \beta + \pi)$

$$\frac{1}{\sin(\psi - \beta)} \leq c_0 + c_1 \left(\frac{\psi}{\psi_0}\right)^a, \text{ with } c_1 = \frac{1}{\sin(\psi_0 - \beta)},\ a = -\psi_0 \cot(\psi_0 - \beta), \tag{20}$$
$$c_0 = \max\left\{\frac{1}{\sin(\psi_0 - l - \beta)} - c_1 \left(\frac{\psi_0 - l}{\psi_0}\right)^a, \frac{1}{\sin(\psi_0 + r - \beta)} - c_1 \left(\frac{\psi_0 + r}{\psi_0}\right)^a\right\}.$$

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

$$\frac{1}{\sin(\psi - \beta)} \leq c_0' + \frac{1}{(\psi - \beta)^a}, \text{ with } a = \frac{\log(\sin(0.4))}{\log(0.4)} \approx 1.029, \tag{21}$$
$$c_0' = \max\left\{\frac{1}{\sin(\psi_0 + r - \beta)} - \frac{1}{(\psi_0 + r - \beta)^a}, 0\right\}.$$

When $\beta + \pi - G \leq \psi_0 < \beta + \pi$, we take symmetrically on $[\psi_0 - l, \beta + \pi)$

$$\frac{1}{\sin(\psi - \beta)} \leq c_0'' + \frac{1}{(\beta + \pi - \psi)^a}, \text{ with } a = \frac{\log(\sin(0.4))}{\log(0.4)} \approx 1.029, \tag{22}$$
$$c_0'' = \max\left\{\frac{1}{\sin(\psi_0 - l - \beta)} - \frac{1}{(\beta + \pi - \psi_0 + l)^a}, 0\right\}.$$

The constants c_0', c_0'' are again chosen in order to obtain an upper bound. The constant a is chosen here to get an upper bound even without the constant term c_0' for (21) on $\psi \in (\beta, \beta + 0.4]$ and without c_0'' for (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 $\sin(x) \geq x^a$ on $[0, 0.4]$, resulting in an upper bound for the inverses. The value 0.4 is chosen somewhat arbitrarily. Fig. 6 shows the behavior of the approximation with $G = 0.6$, which is quite good.

Note however that the bounds on the right-hand side of (21), (22) are not a posynomials! Nonetheless, these expressions can be handled using geometric programming, as we discuss now. When using say approximation (21), we replace the separation and metering constraints by stronger constraints of the form

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

where p, q are posynomials in the speed and heading variables and $a > 0$. We can then rewrite such a constraint as the pair of constraints

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

where $t > 0$ is a new variable, and rewrite the second constraint in (24) as the generalized posynomial constraint

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

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

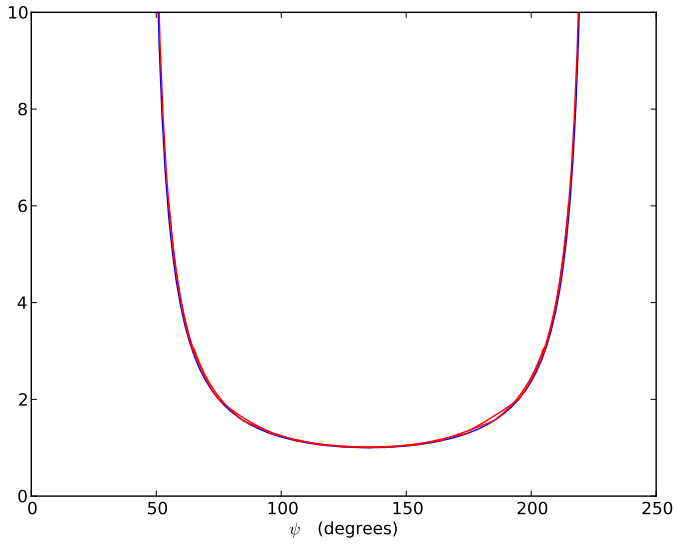

Fig. 6 Local overapproximations (in red) of $\psi \mapsto 1/\sin(\psi - \pi/4)$ (in blue), for $\pi/4 < \psi < \pi/4 + \pi$. For $\psi_0 = \pi/4 + 10^\circ, \pi/4 + 20^\circ, \ldots$, the approximations (19) or (21), (22) are plotted for $\psi \in [\psi_0 - 10^\circ, \psi_0 + 10^\circ]$. We chose a value $G = 0.6$ here.

IV. Metering and Scheduling using Mixed-Integer Geometric Programming

In the rest of the paper, we present specific scenarios involving conflict resolution and metering, and illustrate the application of the previously developed optimization framework based on geometric programming. In addition, in this section we discuss how disjunctions of separation and metering constraints can be modeled using mixed-integer geometric programming, to also optimize over the combinatorial space of possible crossing patterns and orderings of aircraft. As an illustration of the method, we solve here a problem involving the determination of the optimal aircraft ordering at a metered fix.

A. Handling Disjunctions in Geometric Programs

We discussed safety and metering constraints in Section II, and (17), (18) are some examples of posynomial cost functions we could optimize subject to these constraints being satisfied. 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 airspace 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 on Fig. 3, for the similar problem involving a miles-in-trail restriction MIT. The optimal ordering of the aircraft 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 we want to solve is then

$$\begin{aligned}
\min \quad & f(v_1, \ldots, v_N) && (25)\\
\text{subject to} \quad & v_{i,\min} \leq v_i \leq v_{i,\max}, \quad i = 1, \ldots, N && (26)\\
& (2) \vee (3) \vee (6) \vee (7), \quad 1 \leq i \leq N-1, i+1 \leq j \leq N && (27)\\
& \left[\frac{d_i v_j}{d_j v_i} + \mathsf{MINIT}\frac{v_j}{d_j} \leq 1\right] \vee \left[\frac{d_j v_i}{d_i v_j} + \mathsf{MINIT}\frac{v_i}{d_i} \leq 1\right], \quad 1 \leq i \leq N-1, i+1 \leq j \leq N. && (28)
\end{aligned}$$

In (25), the objective function f is chosen to be a generalized posynomial. The disjunction (27) for a pair (i, j) represents the fact that satisfying one of these constraints is sufficient to ensure separation

of the aircraft until one of them reaches the fix. The metering constraints (28) (see subsection II C 2) also involve a disjunction, where the first (resp. second) literal is selected if i (resp. j) reaches the fix first. Note that by solving this program we determine the best ordering of aircraft at the fix, rather than fixing this ordering a priori.

We now show how to model the disjunctions by introducing additional integer variables. Recall [10, 26] the standard "big-M" formulation used to model a disjunction of linear constraints

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

using mixed-integer linear programming. Assuming that we have bounds on the variables x, we can rewrite the disjunction above as the conjunction

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

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). Hence we modify the method as follows. Consider the disjunction of posynomial constraints

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

and the conjunction

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

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

More generally, for a disjunction of n posynomial constraints

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

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

$$g_i(x) + b_i M \leq 1 + 2M,$$

and in addition the posynomial constraint

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

which forces at least one of the b_i variables to be 2, and the corresponding constraint in the disjunction to be enforced. Modeling the disjunctions in the separation and metering constraints (27), (28) this way, we obtain a Mixed Integer Geometric Program (MIGP). In addition, to find the solution of the minimization problem, it is sufficient to consider the situation where exactly one b_i has value 2, enforcing just one constraint per clause, hence replacing (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 (31) since it reduces the search space, for example by eliminating variable b_n.

Using the above big-M method to reformulate the program (25)-(28), with say f a posynomial, we obtain an MIGP with N continuous variables $v_1, \ldots, v_N$, and $2N(N-1)$ binary variables. The number of binary variables is particularly problematic, as the development of MIGP solvers is not as advanced as for MILP solvers. With the currently available open-source solver Bonmin[27], we could solve reliably problems with 5 aircraft in a few seconds, and problems with 6 aircraft in about one minute, see Table 1.

Table 1 Computation times for the MIGP solving the fix scheduling scenario. The problems were solved using the branch-and-bound algorithm implemented in Bonmin [27], on a 3.06GHz Intel Core 2 Duo processor with 4GB 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	# 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

B. Simulation Results

For the scenario described in the previous paragraph, we solve the resulting MIGP (25)-(28) with the clearing time objective (17) by first rewriting the geometric programming part in convex form via the change of variable (14), and then using the mixed-integer nonlinear programming (MINLP) solver Bonmin [27]. 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 2 and 6 aircraft. The aircraft initial positions are generated randomly in a 100×400 nmi rectangle, and the planes must all pass through an airspace fix situated 400 nmi away, see Fig. 7. A Minutes-In-Trail restriction of 10 minutes between successive aircraft is enforced at the fix, and the separation distance between aircraft is fixed to 10 nmi. Among the various algorithms implemented in Bonmin [27], 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. For such mixed-integer convex programs, Bonmin is an exact solver, that is, it eventually returns the optimal solution if the problem is feasible.

MINLP and MIGP 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 6 aircraft involved simultaneously in a conflict, half of the generated cases required more than one minute 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. Fig. 8 shows the distribution of the computation times for 100 simulations with 5 aircraft. Note the presence of a few outliers for which the computation time is much larger than for the typical case.

V. An 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 ordering of aircraft as well as the crossing patterns of the trajectories, i.e., to decide for each pair of aircraft on which side of the forbidden cone the relative velocity vector should lie. For complex problems involving many aircraft, MIGP solvers are not fast enough for real-time applications, and thus in this section we present a fast heuristic based on geometric programming, which essentially chooses a priori which constraints in the disjunctions to enforce. In contrast to the scenario of Section IV, we consider here the situation where the ATC can modify both aircraft velocities as well as headings within given bounds.

Our heuristic first resolves the head-on conflicts, i.e., those for which the initial aircraft headings are within the set $\mathcal{C}_{ij}^0$ defined in (4), by a necessary heading change. It then locally optimizes the velocities and headings while ensuring separation, assuming in this second step that only small heading changes are allowed. The local optimization is based on posynomial approximations of the nonlinearities present in the separation and metering constraints, using the results presented in Section III B. This heuristic does not provide a global optimum in general, and in particular we do not optimize over the crossing patterns. Because it only solves a convex optimization problem however, it can be used in real-time and in a periodic re-optimization strategy (see Section V C). Moreover, if it finds a feasible solution, this solution is guaranteed to be safe for our kinematic

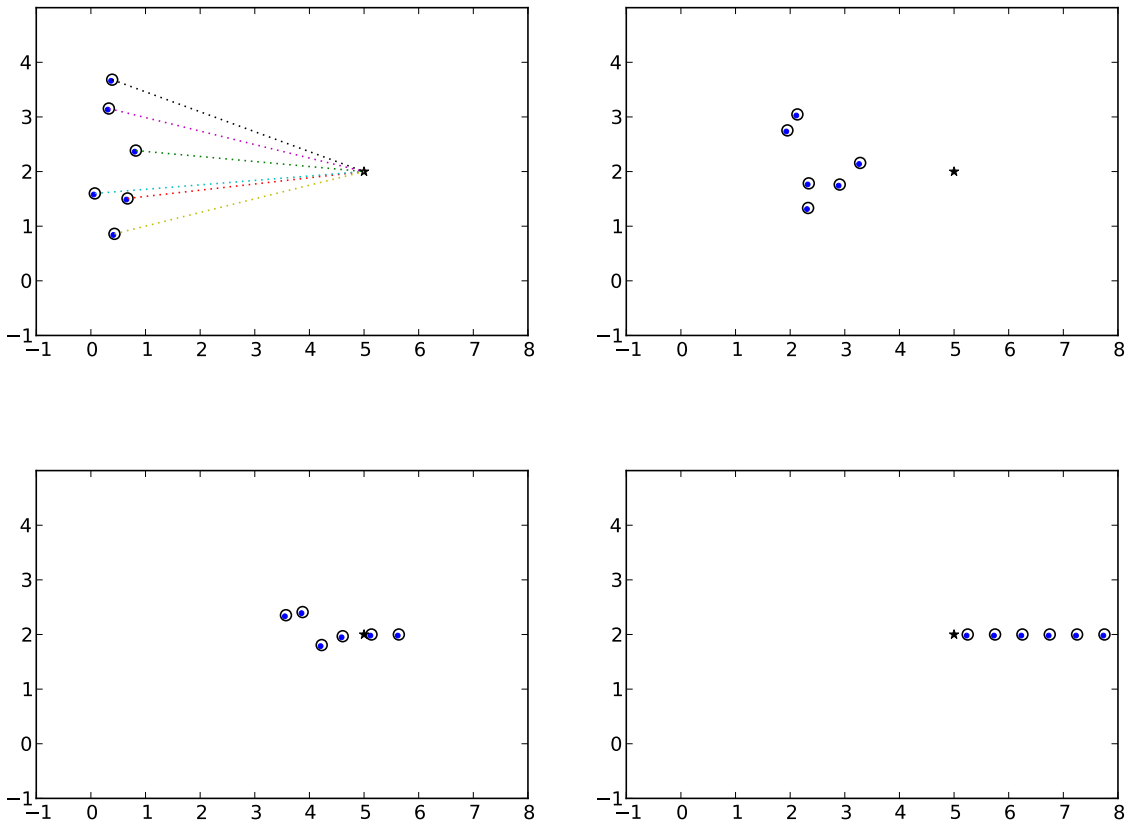

Fig. 7 Four snapshots with the positions of six aircraft approaching a merging point (at coordinates $(5, 2)$), subject to a $\mathsf{MINIT} = 10$ min restriction, and with mandatory separation distance equal to 10 nmi (represented by the small circles). A unit on the figure corresponds to 100 nmi. The velocities are constrained to be between 200 kn and 300 kn for three aircraft and between 250 kn and 350 kn for the other three. Once an aircraft reaches the fix, a new ATC command instructs it to follow the heading $\psi = 0^\circ$ at velocity 300 kn. The MIGP decides the optimal ordering of the aircraft based on a given cost function (here simply the total travel time), while maintaining separation at all times.

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 Section 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 in order to avoid loss of separation. Moreover, this cannot necessarily be accomplished with a small heading change if the two aircraft are close to each other. In Phase I of our algorithm, we remove the occurrences of such configurations between any two aircraft in the region under consideration, by modifying the headings only. More precisely if $i < j$ are two aircraft with $\psi_{i,0}, \psi_{j,0}$ in $\mathcal{C}^0_{ij}$, we move the headings out of the region $\mathcal{C}^0_{ij}$ by using right turn maneuvers as described on Fig. 9. We impose that all aircraft turn in the same direction in order to resolve these head-on conflicts. Hence to resolve conflicts using right turns, we reset $(\psi_{i,0}, \psi_{j,0}) \in \mathcal{C}^0_{ij}$ 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. Our strategy is clearly not optimal, but should be satisfying in the usual 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, until a head-on conflict free configuration is found. We do not claim any termination guarantee for this process. Indeed, some artificial examples lead to conflicts that cannot

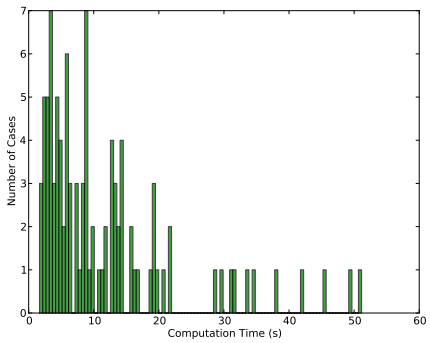

Fig. 8 Distribution of the computation times for the fix scheduling problem with 5 aircraft, over 100 simulations.

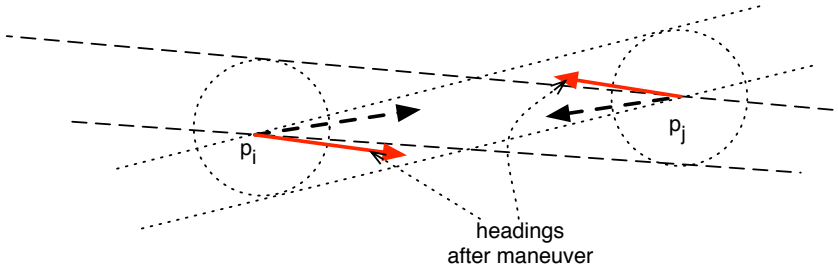

Fig. 9 Maneuver to remove configurations $\mathcal{C}^0_{ij}$. The direction of rotation of the aircraft is fixed and the same for all planes (here all planes turn right to resolve conflicts). As can be seen on the figure even in the case of two planes, this rule is not necessarily optimal since turning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

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, with reasonably large initial separation between aircraft, this heuristic proved to be sufficient in our simulations, see Section V C. Note that potentially much more computationally intensive procedures could be used to handle conflicts in this phase more rigorously, e.g. using mixed-integer linear programming [10].

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

B. Phase II: Local Parameter Optimization via Geometric Programming

In Phase II of the algorithm, as a result of running Phase I above, we assume that no aircraft has its heading $\psi_{i,0}$ exactly equal to $\beta_{i,j}, \gamma_{i,j}, \beta_{i,j}+\pi$ or $\gamma_{i,j}+\pi$, and no pair of aircraft $i<j$ has its headings $\psi_{i,0}, \psi_{j,0}$ in the set $\mathcal{C}^0_{ij}$. We then modify velocities and headings in order to resolve any remaining conflict and optimize a cost objective. If conflicts are detected long enough in advance, we can expect the necessary variations in headings (and velocities) to be small. For example, in all simulation results reported in Pallottino et al. [10], heading changes were smaller than 12°, even though speed changes were not allowed. Clearly, allowing speed changes can only help decreasing heading changes if the goal is to minimize deviations from the initial trajectory for example. In our heuristic, we restrict the variations of headings, e.g. to $\pm 15°$ from the values $\psi_{i,0}, i=1,\ldots,N$. Note that similar restrictions are also included in other previous work [7, 8], although there the a priori allowed heading variations are potentially larger.

1. Heuristic Choice of Literal in the Separation Constraints

In order to simplify the computations and avoid the introduction of combinatorial choices, we choose a priori for each pair $i < j$ of aircraft one of the constraints (2) or (3) that we want to enforce. The finite-horizon constraints (6), (7) are not used here, although they could be included to reduce conservativeness. For each pair $i < j$, we distinguish 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. This means that the aircraft i, j are initially not in conflict, and no conflict between them can arise by simply changing their speeds. We then choose to enforce the trivial constraint (2) or (3), and impose 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 $\sin(\psi_{j,0} - \beta_{ij}) < 0$, we add the constraints

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

   With these bounds added, we can then remove the separation constraint for the pair i, j, as any feasible choice of v_i, v_j, ψ_i, ψ_j automatically guarantees separation.

2. all the implications 1-4 of Proposition 1 hold. In this case for certain values of v_i, v_j, a conflict can arise even if i and j are initially not in conflict, and we can also modify v_i, v_j, ψ_i, ψ_j to resolve an initial conflict between aircraft. We choose one of the constraints (2) or (3) that we want to enforce, while respecting the following rule: it must be possible to satisfy the 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 we cannot choose constraint (2), 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, because we assumed that the initial headings are not in $\mathcal{C}_{ij}^0$, we can choose constraint (3). This is a consequence of the definition (4), which leads to the conclusion

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

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 an important influence on the feasibility of the overall optimization problem. Note that the meaning of constraint (2) is that we wish to push $\mathbf{v}_{ij}$ on Fig. 1 outside of the forbidden cone, in the direction of $\mathbf{n}_{ij}^1$. Similarly, satisfying constraint (3) means that $\mathbf{v}_{ij}$ is pushed in the direction of $\mathbf{n}_{ij}^2$. Our heuristic simply 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 $\mathbf{n}_{ij}^1$, and constraint (3) otherwise. In other words, we force the algorithm to keep $\hat{\mathbf{v}}_{ij}$ and $\mathbf{v}_{ij}$ on the same side of the line $(\hat{\mathbf{p}}_i, \hat{\mathbf{p}}_j)$.

Once a constraint to enforce has been chosen, we impose bounds on ψ_i, ψ_j 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 we add the constraints

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

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 $A < \psi < A + \pi$. Thus, the approximations discussed in section III B are sufficient. With these additional bounds on headings included, we then rewrite the chosen constraint (2) or (3) in the form

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

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

2. Optimization

At this point we have chosen a separation constraint to enforce for each pair of aircraft, set bounds on the heading variables that allow us to use posynomial approximations of the trigonometric terms, and expressed the selected separation constraints in the form (33). The next step is to perform a posynomial approximation of these inequalities around the headings $\{\psi_{i,0}\}_{1\leq i\leq N}$ (obtained at the end of phase I). For this purpose, we use the conservative approximations developed in Section III B. The satisfaction of the posynomial inequalities presented next is sufficient to guarantee that the separation constraints (33) are satisfied. In order to use the local approximations, we potentially further restrict the variations of the heading variables ψ_i. 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, we chose to limit heading variations to $\pm 15°$.

The upper bounds discussed in section III B lead to replacing (33) by a more conservative constraint of the form

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

by using (19), (20); or, if $\psi_{l,0} \in (A, A+G]$, of the form

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

by using (19), (21); or finally, if $\psi_{l,0} \in [A+\pi-G, A+\pi)$, of the form

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

by using (19), (22). Constraint (34) is a posynomial constraint. It was also explained at the end of subsection (III B) how to convert (35) and (36) to equivalent posynomial constraints. Finally, metering constraints such as (12) can be similarly handled by geometric programming using again the same approximations of the sine and inverse sine functions. With the cost function and all constraints expressed in the form of posynomials, we can finally solve the geometric program.

C. Simulations

We now present some simulations illustrating the results obtained with the algorithm discussed in the preceding subsections. In these simulations, we set the interval of allowed heading variations to $\pm 15°$ in order to develop the posynomial approximations of Subsection III B. Geometric programs are solved in Matlab using the convex modeling package CVX [28, 29].

We first consider a pure conflict resolution problem, i.e., with no metering constraints, where aircraft are initially randomly distributed in a 300 nmi $\times$300 nmi square, with random initial headings. Fig. 10 shows two instances of the problem, involving 15 and 20 aircraft respectively. Any two aircraft are initially separated by at least 30 nmi in the generated instances. The initial speeds are uniformly and randomly distributed over the interval [180 kn, 300 kn], 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

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

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 on 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, our heuristic 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 on Fig. 11 shows empirically how the number of instances found to be infeasible in simulations grows with the density of planes considered.

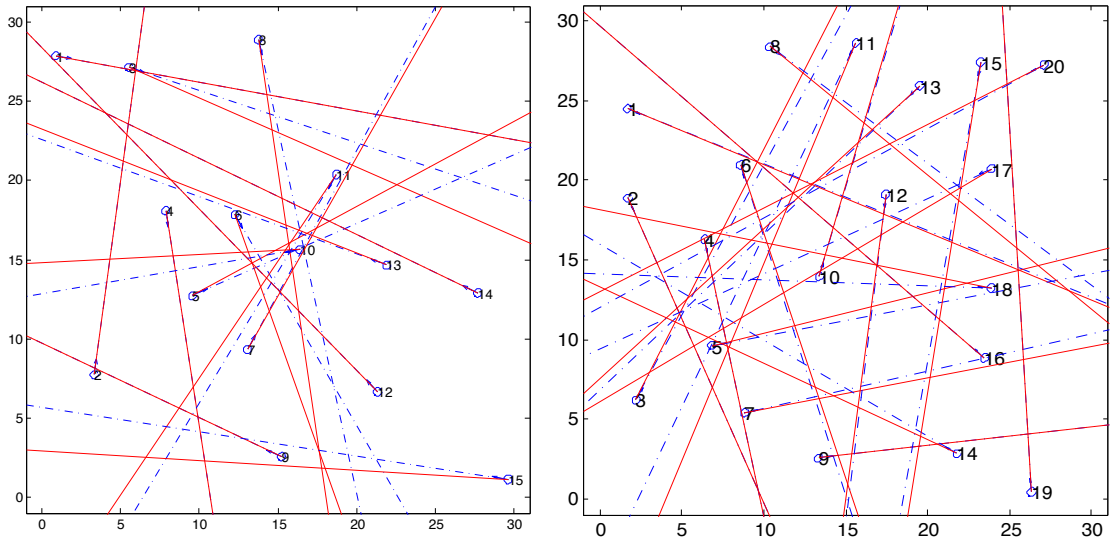

Fig. 10 Conflict resolution with 15 (left) and 20 (right) aircraft in a 300 nmi ×300 nmi square. Each aircraft initial position is represented together with a circle of diameter 5 nmi around it (one unit on the figure represents 10 nmi). Dashed lines represent the initial trajectories, and solid lines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

Next we consider problems involving both conflict resolution and metering. On Fig. 12 we show a scenario with 14 aircraft with crossing trajectories, initially involved in 7 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 minutes. The cost function minimized is again (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 Section IV.

The simulations above only show the result of a single stage optimization. Note however that solving the moderate size geometric programs arising in typical conflict resolution scenarios can be done in real-time. Hence the optimization can be repeated at regular intervals to continuously adjust the trajectories based on the current positions of the aircraft, resulting in piecewise linear trajectories. This approach also guarantees that if we stop updating the speed and heading variables, the aircraft fly straight trajectories that are conflict-free. Fig. 13 shows an example illustrating this periodic optimization procedure. At each optimization step we use as initial headings for the local optimization the desired headings starting from the current location. Comparing Fig. 12 and 13, we see that this procedure significantly reduces the deviations necessary to resolve conflicts.

VI. Conclusion

We have presented in this paper a novel conflict resolution method based on geometric programming. The method is able to handle many aircraft simultaneously and can be used to simultaneously optimize over speeds and headings, assuming only small heading changes. A key feature of this approach is that we can incorporate metering constraints in the path planning problem. Hence we believe that 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 permit a smoother integration of automated tools for decision support in air traffic control.

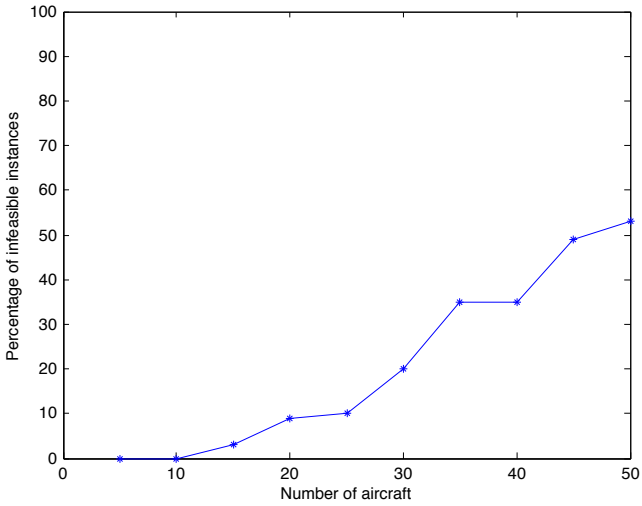

Fig. 11 Percentage of infeasible instances for the fast heuristic in the conflict resolution scenario of Fig. 10, as the number of aircraft increases. 100 instances were generated in each case.

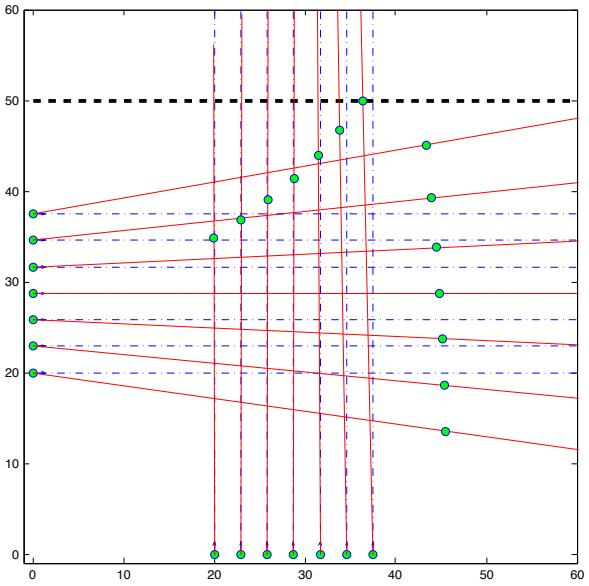

Fig. 12 Crossing of 14 aircraft with metering. One unit represents 10 nmi. The initial speed of all aircraft is the same, creating 7 conflicts. Moreover, we have the constraint that two successive aircraft crossing the thick dashed line must be separated by at least six minutes. We show the position of the aircraft at $t = 0$ and at the time where the first aircraft traveling North crosses the metered boundary.

Acknowledgments

This work was supported by the ONR-MURI award N00014-08-1-0696.

[1] A. Haraldsdottir, R. W. Schwab, A. Shakarian, G. Wood, and R. S. Krishnamachari, in *New Concepts and Methods in Air Traffic Management*, edited by L. Bianco, P. Dell'Olmo, and A. R. Odoni (Springer, 2001), pp. 63–74.

[2] M. S. Nolan, *Fundamentals of Air Traffic Control* (Brooks Cole, 2003), 4th ed.

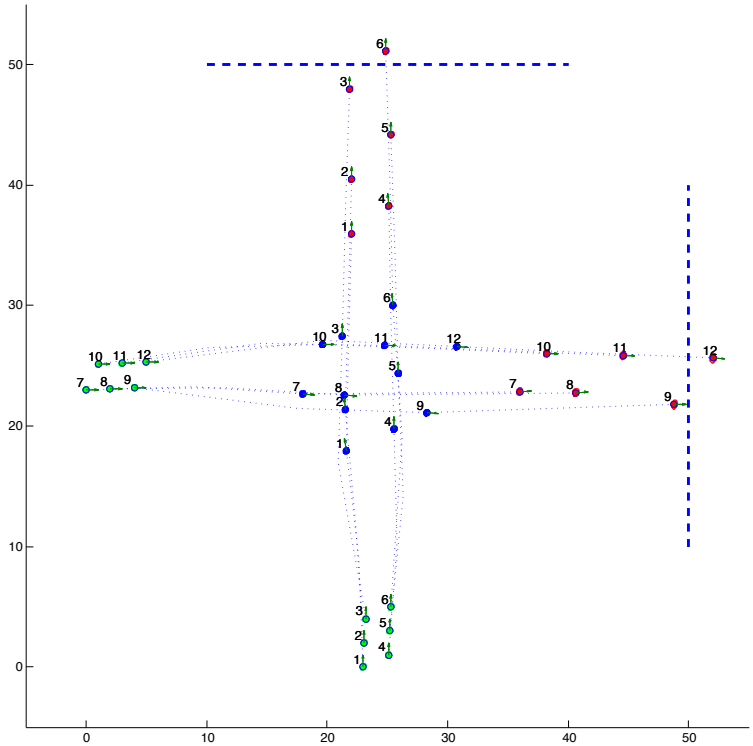

Fig. 13 Crossing of 12 aircraft with metering. One unit represents 10 nmi. The initial configuration is symmetric with respect to the diagonal, producing 6 conflicts. The two lines are metered boundaries, each with a minutes-in-trail restriction $MINIT = 6$ min between successive aircraft crossing it. For each boundary, the order in which the aircraft cross it is fixed. In contrast to Fig. 12, we re-optimize the headings and velocities every 30 min, resulting in piecewise linear trajectories. We show the positions of the aircraft for three snapshots, at $t = 0$, $t = 1$ hr, and $t = 2$ hr. The discs 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 directions starting from the current aircraft positions and directed toward the centered waypoint point $[23.5, 100]$, and symmetrically for aircraft 7-12. This allows the aircraft to correct their trajectories to minimize deviations once the conflicts are resolved.

[3] Federal Aviation Administration, *New York Area Standard Operating Procedures and Letters of Agreement*, Order 7110.1A (2005).

[4] M. Robinson, R. DeLaura, B. Martin, J. E. Evans, and M. E. Weber, in *Aviation, Range and Aerospace Meteorology Special Symposium on Weather-Air Traffic Management Integration, AMS Annual Meeting* (Phoenix, AZ, 2009).

[5] P. Ostwald, T. Topiwala, and J. DeArmon, in *6th AIAA Aviation Technology, Integration and Operations Conference (ATIO)* (2006).

[6] K. T. Mueller, D. R. Schleicher, and K. D. Bilimoria, in *Proceedings of the AIAA Guidance, Navigation and Control Conference* (Monterey, CA, 2002).

[7] W. P. Niedringhaus, IEEE Transactions on Systems, Man, and Cybernetics **25**, 1269 (1995).

[8] E. Frazzoli, Z.-H. Mao, J.-H. Oh, and E. Feron, Journal of Guidance, Control, and Dynamics **24** (2001).

[9] G. Inalhan, D. Stipanovic, and C. Tomlin, in *Proceedings of the 41st IEEE Conference on Decision and Control* (Las Vegas, 2002).

[10] L. Pallottino, E. M. Feron, and A. Bicchi, IEEE Transactions on Intelligent Transportation Systems **3**, 3 (2002).

[11] J. K. Kuchar and L. C. Yang, IEEE Transactions on Intelligent Transportation Systems **1**, 179 (2000).

[12] R. Schild, Ph.D. thesis, Department of Econometry, Operations Research and System Theory, Technical University of Vienna (1998).

[13] J. Hoekstra, R. van Gent, and R. Ruigrok, in *Proceedings of the AIAA Guidance, Navigation and Control Conference* (Boston, MA, 1998), pp. 807–817.

[14] I. Hwang, J. Kim, and C. Tomlin, Air Traffic Control Quarterly **15** (2007).

[15] K. Zeghal, in *Proceedings of the AIAA Guidance, Navigation and Control Conference* (Boston, 1998),

pp. 818–827.
[16] N. Durand, J. M. Alliot, and O. Chansou, Air Traffic Control Quarterly **3**, 139 (1995).
[17] P. K. Menon and G. D. Sweriduk, Journal of Guidance, Control, and Dynamics **22**, 202 (1999).
[18] A. U. Raghunathan, V. Gopal, D. Subramanian, L. T. Biegler, and T. Samad, Journal of Guidance, Control, and Dynamics **27**, 586 (2004).
[19] K. D. Bilimoria, in *Proceedings of the AIAA Guidance, Navigation and Control Conference* (Denver, CO, 2000).
[20] J. Hu, M. Prandini, and S. Sastry, AIAA Journal of Guidance, Control and Dynamics **25**, 888 (2002).
[21] K. D. Bilimoria and H. Q. Lee, in *Proceedings of the AIAA Guidance, Navigation and Control Conference* (Monterey, CA, 2002).
[22] D. Dugail, E. Feron, and K. D. Bilimoria, in *AIAA Guidance, Navigation and Control Conference* (Monterey, CA, 2002).
[23] R. J. Duffin, E. L. Peterson, and C. M. Zener, *Geometric Programming* (Wiley, 1967).
[24] K. O. Kortanek, X. Xu, and Y. Ye, Mathematical Programming **76**, 155 (1997).
[25] S. Boyd, S.-J. Kim, L. Vandenberghe, , and A. Hassibi, Optimization and Engineering **8**, 67 (2007).
[26] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research* (McGraw-Hill, New York, 1995), 6th ed.
[27] P. Bonami, J. J. Forrest, L. Ladanyi, J. Lee, F. Margot, and A. Wächter., *Bonmin: Basic Open-source Mixed INteger solver.*, http://www.coin-or.org/Bonmin.
[28] M. Grant and S. Boyd, *CVX: Matlab software for disciplined convex programming, version 1.21*, http://cvxr.com/cvx (2010).
[29] M. Grant and S. Boyd, in *Recent Advances in Learning and Control*, edited by V. Blondel, S. Boyd, and H. Kimura (Springer-Verlag, 2008), Lecture Notes in Control and Information Sciences, pp. 95–110.