Efficient computation of separation-compliant speed advisories for air traffic arriving in terminal airspace

Alexander V. Sadovsky[*]

NASA Ames Research Center
Moffett Field, CA 94035

Damek Davis

UCLA, Department of Mathematics
Los Angeles, CA 90095-1555

Douglas R. Isaacson

NASA Ames Research Center
Moffett Field, CA 94035

ABSTRACT

A class of problems in air traffic management asks for a scheduling algorithm that supplies the air traffic services authority not only with a schedule of arrivals and departures, but also with speed advisories. Since advisories must be finite, a scheduling algorithm must ultimately produce a finite data set, hence must either start with a purely discrete model or involve a discretization of a continuous one. The former choice, often preferred for intuitive clarity, naturally leads to mixed-integer programs, hindering proofs of correctness and computational cost bounds (crucial for real-time operations). In this paper, a hybrid control system is used to model air traffic scheduling, capturing both the discrete and continuous aspects. This framework is applied to a class of problems, called the Fully Routed Nominal Problem. We prove a number of geometric results on feasible schedules and use these results to formulate an algorithm that attempts to compute a collective speed advisory, effectively piecewise linear with finitely many vertices, and has computational cost polynomial in the number of aircraft. This work is a first step toward optimization and models refined with more realistic detail.

1 Introduction

Management of air traffic which is confined to a network of nominal routes requires that each aircraft be scheduled to at least a few crucial waypoints and arrive at each waypoint as near as possible to the scheduled time. This requirement gives rise to an entire class of scheduling problems, which vary by airspace configuration and various flight restrictions, such as speed- or altitude ranges, aircraft type, and weather. Since the airspace is often modeled as a directed graph, and a schedule as a chart that estimates for each aircraft a time of arrival at each of the finitely many waypoints, models of centralized scheduling of arrivals are frequently also sought in this, discrete, setting [1–7].

In such a model, aircraft are scheduled only at specific points (the vertices of the graph), with no time stamps maintained or imputed of the aircraft's position or speed away from such a point. One weakness of such inherently discrete models is their principal inability to assure, without adding artificial constraints, pairwise separation between the aircraft continuously in time, as is operationally required [8]. Another weakness, specific to the models that lead to a *mixed-integer program (MIP)* [9], is the difficulty of finding an algorithm that performs sufficiently better than the *NP*-hard worst case of a general *mixed-integer linear program (MILP)* [10], although a number of the algorithms developed for solving a MILP modeling air traffic have been shown to perform with a running time polynomial in the number of aircraft (see, for example, [5, 11]). However, while sufficient constraints can be imposed in a discrete model to assure separation at all times, the authors are unaware of published discrete models that assure such separation and show computational performance suitable for aircraft traffic in a terminal airspace.

[*] Corresponding author. Email: *alexander.v.sadovsky@nasa.gov*

In this paper, we approach the scheduling of air traffic as a time-continuous problem. The *hybrid control system (HCS)* framework proposed in [12] is used for modeling air traffic motion continuously in time in a broad class of airspace regions represented as directed multigraphs. A schedule sought in this framework is, thus, a speed advisory, i.e. a continuous time parameterization of each aircraft's motion along a pre-determined path. The pairwise separation requirements are imposed also in continuous time.

This framework has two main merits. First, it is capable of capturing realistically the following aspects of *Air Traffic Management (ATM)*: separation requirements for every pair of aircraft at every instant in time, speed restrictions, off-nominal routes, and uncertainty in the actual times of an aircraft's arrival at a waypoint. Second, it rests on the apparatus of HCS, which is relatively well-understood theoretically [13–19] and furnished with general numerical methods (see, e.g., [20]). While no optimization is pursued in this paper, and only *feasible trajectories* are sought for a given HCS, the framework can serve as a setting for problems in optimal control [12].

The contribution of this paper is an algorithm for the class of *Fully Routed Nominal Problems* (section 6.1) in ATM, where the goal is to guide a given finite set of aircraft, each along its pre-determined path, out of the route network in a separation-compliant way. The algorithm, formulated in section 6.5, constructs, or reports failure to construct, a feasible (i.e., compliant with the separation and speed range constraints) speed advisory that guides *all* the aircraft as required. The computational cost is shown to be bounded by a low-degree polynomial in the number of aircraft.

2 Background

The inherently discrete models mentioned in section 1 have historic origins. Planning of air transportation, from the time it came into mass use and to the time of this writing, has in practice been based on two doctrines, *i)*, that scheduling and separation assurance are two separate tasks and, *ii)*, that a schedule need not synthesize for each aircraft a 4-D aircraft trajectory (i.e., a time-parameterized curve in 3-D space), but only specify an appropriate set of waypoints and the times at which they should be reached by the aircraft. (Current air transportation operations rely on *Air Traffic Control (ATC)* to devise and issue clearance instructions to each aircraft and to modify the schedule when necessary in order to maintain aircraft separation.) While automated trajectory synthesis has received some attention from researchers (see [21] and references within) and is finding its way into air traffic operations (see, e.g., [22,23]), doctrine *i)* remains a dominant practice: currently such synthesis does not meet, simultaneously, a given schedule and the time-continuous pairwise separation requirement.

As these conservative tendencies have persisted, increases in traffic demand have hampered ATC's mission–to promote "safe, orderly, and expeditious flow of traffic" [8]–in all aspects except safety. To address these newly arising difficulties of ATC, researchers have sought to minimize various performance indices. Examples of such indices are makespan [2, 5], fuel burn [24, 25], average or total delay [3, 24], excess separation [25], and deviation from preferred time of arrivals and departures [1]. With this broadened focus, researchers have approached various types of air traffic scheduling problems in terminal airspace using such frameworks as dynamic network flows, constraint-based scheduling, and scheduling combined with separation considerations; a detailed survey, references, and discussion of this past research can be found in [26].

Further work on, and wider use of, automated generation of 4-D trajectories, as envisioned in the future [27], will raise the question of how the responsibility for scheduling and for separation should be divided between the automation system and the human controller. Having 4-D trajectory generation handled by the automation, and separation assurance by the controller, would leave the controller with the full weight of separation responsibility while considerably restricting their freedom in the choice of aircraft maneuvers. Such a practice is likely to limit the benefits associated with the use of 4-D trajectories and to be unacceptable to Air Traffic Controllers. This consideration has led us to consider an investigation, undertaken herein, of algorithms for 4-D trajectory generation that aim, as a Controller does today, to maintain separation along the entire continuous trajectory for each aircraft.

3 An HCS model of an airspace

For the operational purposes of ATM, an airspace is regarded as a region with a set of marked points, called *waypoints*. Some pairs of waypoints are connected by route segments. (For example, a portion of the airspace surrounding the Los Angeles International Airport is shown in Fig. 1.) The same pair of points may be connected by more than one segment. In practice, a route segment directly connecting a pair of waypoints is rectilinear, but this assumption will made here only in section 6. This setting is naturally modeled by a directed *multigraph* [28]: the waypoints are the vertices, while the route segments are the edges. A sequence of route segments with every consecutive pair adjacent is called a *route* and is none other than the graph-theoretic *path* [28].

Although the application central to this paper is in Air Traffic Management, the following construct is, in principle, suitable for any finite set of moving agents whose movement is confined to a route network. Assume one is given the following data:

1. A directed multigraph $G = (V, E)$, each vertex $v \in V$ being a point in a Euclidean space $\mathbf{E}$ of dimension 2 or 3. If

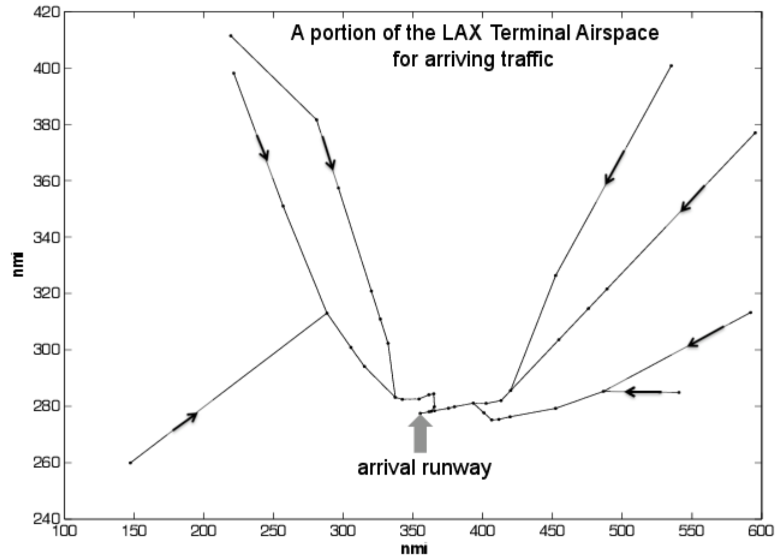

Fig. 1. A portion of the LAX terminal airspace for arriving traffic. The thin arrows indicate the traffic directions.

$e \in E$ is an edge from $v_1 \in V$ to $v_2 \in V$, then the nominal route segment from waypoint v_1 to waypoint v_2 is a curve in $\mathbf{E}$, connecting v_1 to v_2. All such curves will henceforth be assumed *rectifiable* [29, section 4.6-9] and capable of a parameterization which is continuous and piecewise continuously differentiable. A cusp in the curve can be traversed with the assumption (made throughout this paper) that inertia is neglected, and approximately smoothed if inertia is included. The outdegree and indegree of a vertex are allowed arbitrary values. To emphasize the aerospace context, the multigraph $G = (V, E)$ will be called a *route network*.

2. A finite set $\mathcal{A} = \{1, \ldots, A\}$. of *moving agents* $\alpha \in \mathcal{A}$ in G. If agent α is moving along edge e, the agent's position is specified by the arc length coordinate y_e^α along a path containing e; the choice of arc length coordinates below will be clarified as the model is developed, and *will always be chosen to increase along the direction of the edge*.
3. For each agent $\alpha \in A$, a specification of the agent's initial position, which is a point in G specified, for example, by an edge in G and a fractional distance along that edge.
4. The inertia-free *dynamical law* [13] (henceforth the dot denotes differentiation with respect to physical time t) $\dot{x}_e^\alpha = s_e^\alpha$, where the s_e^α's are the corresponding speeds, describing the motion of those agents α that have not yet exited the system. In what follows, and with the details provided below, the coordinates x_e^α will play the role of *state variables*; the speeds s_e^α, of the *control variables*.
5. *State constraints:* the separation requirement for each pair of agents. This requirement is described mathematically, in terms of the coordinates x^α, in section 4. A state in which at least one pair of agents violates the separation requirement will be called a *conflicting state*.
6. *Control constraints:* bounds on the speeds s_e^α.

One choice of a *hybrid control system (HCS)* [13] fit for modeling coordinated multi-agent motion in a route network G was introduced in [12]. In this HCS, one defines each *discrete mode* (in [13], *location*) as a mapping $\mu : \mathcal{A} \to E$, which specifies for each agent α the edge $\mu(\alpha)$ currently occupied by that agent. Examples of three discrete modes on a 6-edge route network with 3 moving agents are shown in Fig. 2. Denoting the arc length parameterization of edge $\mu(\alpha)$ by x_μ^α, the dynamical law corresponding to discrete mode μ is given by the system $\dot{x}_\mu^\alpha = s_\mu^\alpha, \quad \alpha \in A$, of differential control equations. Suppose each x_μ^α ranges from 0 to the length $l(\mu(\alpha))$ of edge $\mu(\alpha)$. The set X_μ of all continuous states $\mathbf{x}_\mu = (x_\mu^\alpha)_{\alpha \in \mathcal{A}} = \left(x_\mu^1, x_\mu^2, \ldots, x_\mu^A\right)$ in discrete mode μ is then obtained by taking the A-dimensional parallelotope given by the Cartesian product $\prod_\alpha [0, l(\mu(\alpha))]$ and removing from it all states that violate the separation constraint for at least one pair of agents (*conflicting states*). (The set X_μ is an example of a *roadmap coordination space*, defined in [30, section 1.2].) For example, Fig. 3A shows the X_{μ_1} for discrete mode μ_1 from Fig. 2A; the conflicting states are not shown in Fig. 3.

One must then specify, for each pair of states $\mathbf{x}_\mu \in X_\mu, \quad \mathbf{x}_{\mu'} \in X_{\mu'}$, whether it is possible for the system to go from

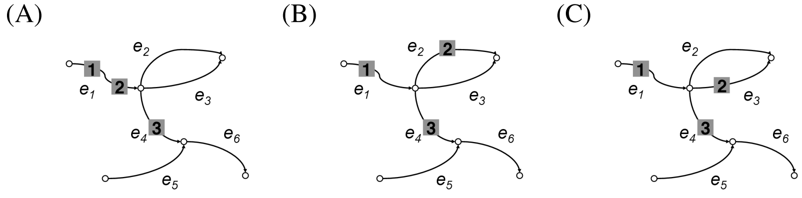

Fig. 2. Examples of three discrete modes with edge set $E = \{e_1, e_2, \ldots, e_6\}$ and moving agent set $\mathcal{A} = \{1, 2, 3\}$ ($A = 3$). The agents are shown as numbered gray squares. The discrete modes shown are: (A) $\mu_1 \, : \, \mu_1(1) = e_1$, $\mu_1(2) = e_1$, $\mu_1(3) = e_4$; (B) $\mu_2 \, : \, \mu_2(1) = e_1$, $\mu_2(2) = e_2$, $\mu_2(3) = e_4$; (C) $\mu_3 \, : \, \mu_3(1) = e_1$, $\mu_3(2) = e_3$, $\mu_3(3) = e_4$.

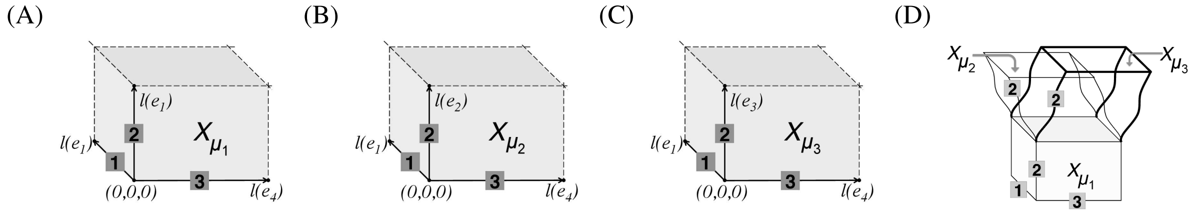

Fig. 3. (A-C) The continuous state spaces $X_{\mu_1}, X_{\mu_2}, X_{\mu_3}$ corresponding to discrete modes μ_1, μ_2, μ_3 from Fig. 2. Among the discrete modes the system may enter from μ_1 are μ_2 and μ_3, accordingly as agent 2 enters e_2 or e_3. Conflicting states not shown. (D) A gluing of X_{μ_1} to X_{μ_2} and to X_{μ_3} (see Fig. 3). Glued, all three state spaces share a face, but no two have any other points in common. Conflicting states not shown.

one to the other, thus switching from one discrete mode to another, instantly. The set, denoted $S_{\mu,\mu'}$, of all such state pairs $(\mathbf{x}_\mu, \mathbf{x}_{\mu'})$ for a given pair of discrete modes μ, μ' is called a *switching set* [13]. For this HCS, the switching sets are completely determined by the following condition: $(\mathbf{x}_\mu, \mathbf{x}_{\mu'}) \in S_{\mu,\mu'}$ *if and only if there is at least one agent, $\overline{\alpha}$, that in discrete mode μ occupies a vertex $v \in V$ with coordinate $x_\mu^{\overline{\alpha}} = l(\mu(\overline{\alpha}))$, and in discrete mode μ' occupies that same vertex with coordinate $x_{\mu'}^{\overline{\alpha}} = 0$.* Thus, as the system switches from μ to μ', agent $\overline{\alpha}$ exits edge $\mu(\overline{\alpha})$ and enters edge $\mu'(\overline{\alpha})$. Geometrically, the switching sets are described by a topological *gluing* [31] (similar to that used in [17, section 3]) between the corresponding continuous state spaces X_μ and $X_{\mu'}$: the right endpoint of $[0, l(\mu(\overline{\alpha}))]$ is identified with the left endpoint of $[0, l(\mu'(\overline{\alpha}))]$. If the transition from μ to μ' involves a change of edge for more than one agent, then each of these agents effects a gluing. Thus, the face $x_{\mu(\overline{\alpha})}^{\overline{\alpha}} = l(\mu(\overline{\alpha}))$ of X_μ is to be glued to the face $x_{\mu'(\overline{\alpha})}^{\overline{\alpha}} = 0$ of $X_{\mu'}$ for each discrete mode μ' that allows a direct transition from μ. (For example, in Fig. 3(A-C), the "top" (in the sense of the Figure) face of X_{μ_1} is glued to the bottom face of X_{μ_2} and of X_{μ_3}; see Fig. 3D.) The state space obtained by carrying out all such gluings, and denoted X, is called the *global continuous state space*. Denoting the gluing operation by the non-standard symbol $\overset{\text{glue}}{\cup}$, one has $X = \overset{\text{glue}}{\cup}_{\mu} X_\mu$. Since the set of the conflicting states in each X_μ is *closed* [29], as it will be defined below using sharp inequalities, two states $\mathbf{x}_\mu, \mathbf{x}_{\mu'}$ identified as a result of the above gluing are either both conflicting or both separation-compliant. It follows from the above that every global continuous state space is a *chain of dimension A on* $\mathbf{R}^A$ [32, section 35E] with the conflicting states removed, but generally fails to be a *differentiable manifold* [32].

The only discrete control variable appearing in this HCS would thus be the one indicating, for an agent reaching a vertex of outdegree ≥ 2, which edge the agent is to take (or whether the agent should exit the system) when leaving that vertex.

4 The geometry of a minimal separation requirement

In some transportation types, including aircraft and trains, every pair of moving agents must be be separated by a distance no smaller than a pre-determined *minimal separation*. Such requirements can be asymmetric and anisotropic; for a detailed discussion, see [12, section II]. Here we will characterize states that violate the separation requirement using conservative approximations, addressing only the following asymmetry: if two moving agents are in-trail (i.e., one is directly following the other along a route segment which is not necessarily in a horizontal plane), then the minimal separation can depend on the weight class of the leading and trailing agent. To capture this potential asymmetry, for each pair α_1, α_2 of agents with the first one leading, we introduce the minimal separation r_{α_1,α_2}. If the asymmetry takes place, it can be written

$$r_{\alpha_1,\alpha_2} \neq r_{\alpha_2,\alpha_1} \tag{1}$$

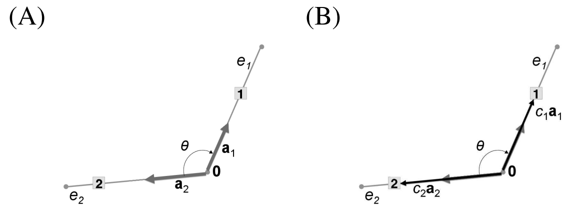

Fig. 4. Agents $1,2$ on their respective rectilinear edges e_1,e_2, which share a common vertex, taken as the origin $\mathbf{0}$ in $\mathbf{R}^2$. The orientation of the edges is not specified. (A) The unit vectors $\mathbf{a}_1,\mathbf{a}_2$ are collinear with the respective edges, but their directions do not necessary agree with the edges' orientations. (B) With suitably chosen scalars coefficients c_1,c_2, the vectors $c_1\mathbf{a}_1$ and $c_2\mathbf{a}_2$ are the respective position vectors of the two agents.

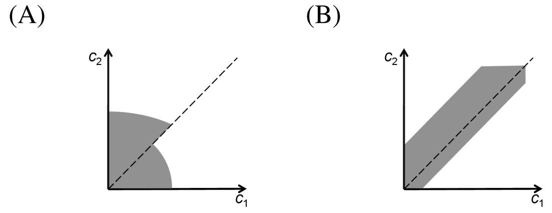

Fig. 5. (A) An example of two elliptical sectors in the c_1c_2-plane corresponding to conflicting states. (B) An example of two stripes in the c_1c_2-plane corresponding to conflicting states of two agents on the same edge.

We now calculate the set of all conflicting states in a discrete mode μ of an HCS described above; recall that these are the states where at least two agents violate the separation requirement. The scenario shown in Fig. 4A has two agents on two different rectilinear edges, which need not lie in a horizontal plane, with a common vertex and no specified orientation. (If the edges are curvilinear with low curvature near a common vertex or intersection, these portions can be approximated by linear segments; otherwise, the analysis becomes considerably more complicated.)

Remark 4.1. *Since edge orientation is not specified, Fig. 4 describes 4 cases: both agents are moving toward the common vertex, both moving away from the common vertex, and two more cases in which one agent moves toward, and the other away from, the common vertex.*

We will use the Euclidean *inner product* [29] $\langle\cdot,\cdot\rangle$ and the corresponding *norm* [29] $||\cdot||$) in the 2-D space containing the two edges. Pick the common vertex as the origin and the unit vectors $\mathbf{a}_1,\mathbf{a}_2$ as the basis vectors that, *regardless of the edge orientations*, point from the origin toward the respective agents. With suitable scalars c_1,c_2, the vectors $c_1\mathbf{a}_1$ and $c_2\mathbf{a}_2$ are the agents' respective position edges. The squared distance between the two agents, denoted by D^2, is

$$D^2 = ||c_1\mathbf{a}_1 - c_2\mathbf{a}_2||^2 = (c_1)^2 + (c_2)^2 - 2c_1c_2\langle\mathbf{a}_1,\mathbf{a}_2\rangle$$

Equating D^2 to the squared minimal separation, say, $r_{1,2}^2$, we obtain the equation $D^2 = r_{1,2}^2$ of an ellipse in the c_1c_2-plane. The corresponding set of conflicting sets is described by the elliptical sector obtained by intersecting the open octant $c_2 > c_1 > 0$ with the ellipse-bound region $D^2 < r_{1,2}^2$ (respectively, $D < r_{2,1}^2$) when agent 1 (respectively, agent 2) is the one closer to the origin. The role of the angle θ between the edges e_1,e_2 in both sectors is the equality $\langle\mathbf{a}_1,\mathbf{a}_2\rangle = \cos(\theta)$. An example of two such sectors is shown in Fig. 5A. The asymmetry of the gray-shaded region about the dashed diagonal is the asymmetry (1).

In each of the four cases listed in Remark 4.1, the respective continuous state coordinates x_μ^1, x_μ^2 of agents $1,2$ in discrete mode μ map to the coefficients c_1,c_2, as follows:

1. If both agents are moving toward the common vertex, then $x_\mu^\alpha = l(e_\alpha) - c_\alpha$ for $\alpha = 1,2$.
2. If both agents are moving away the common vertex, then $x_\mu^\alpha = c_\alpha$ for $\alpha = 1,2$.
3. If agent 1 is approaching, and agent 2 going away from, the common vertex, then $x_\mu^1 = l(e_1) - c_1, x_\mu^2 = c_2$.
4. If agent 2 is approaching, and agent 1 going away from, the common vertex, then $x_\mu^1 = c_1, x_\mu^2 = l(e_2) - c_2$.

If $\theta \geq 90^\circ$, then in the last two cases μ allows only one in-trail sequence, so the minimal separations used for the two sectors in Fig. 5A are equal. If the two agents $1,2$ are on the same edge, then the set in the c_1c_2-plane of the conflicting states appears as in Fig. 5B and is, in accordance with (1), asymmetric about the dashed diagonal. The mapping from the continuous state coordinates x_μ^1, x_μ^2 to the coefficients c_1,c_2 is constructed analogously to the above four cases.

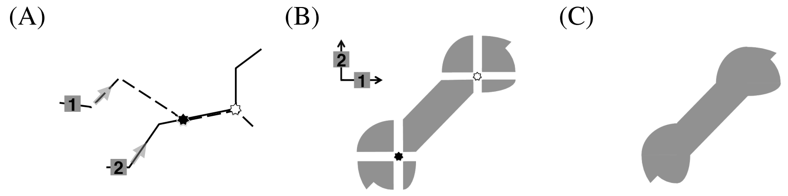

Fig. 6. An example of two agents whose paths overlap. The black star shows the beginning of the overlap in (A) and the corresponding state (both agents being at that point) in (B); the white star, the end of the overlap in (A) and the corresponding state (both agents being at that point) in (B). The system, shown in (A), has 7 discrete modes with both agents in the transportation network. Each mode's set of separation-violating states, shown in (B) as a *connected* [31] gray region, is "glued" to some of the others. The result of the gluing is the connected region shown in (C).

The above calculation is illustrated, for an example of two moving agents, in Fig. 6. Each discrete mode's set of conflicting states is shown as a *connected* [31] gray region. For dimension A above 2, one must compute for each pair of agents the set of states violating the separation requirements. Each such set is a cylinder, or union of cylinders, with the base shaped as shown in Fig. 6C, in the total state space $\cup_\mu X_\mu$. We note that the set of all separation-violating states in $\cup_\mu X_\mu$ is *cylindrical* in the sense of [33, Definition 2.2], the latter definition a key requirement for the applicability of a number of theoretical results of [33].

5 An alternative: path-based transportation HCS

The HCS model formulated in section 3 is sufficiently general to address problems in which an assignment of flight routes to aircraft ("routing") is not provided, but is to be carried out as part of solving the problem; see, e.g., Ref. [34] and references therein. The focus of this paper, however, is a narrower subclass of such problems. This subclass is defined in section 6.1 as the *Fully Routed Nominal Arrival problem.* In a Fully Routed Nominal Arrival problem, one of the underlying assumptions is (as the name suggests) that the routes to be flown by the aircraft are given.

To approach solving Fully Routed Nominal Arrival problems, it seems more convenient to operate, not with the individual edges of the graph G, as is done in section 3, but with entire flight paths in G. The content of this section, thus, is the latter, path-oriented view of the HCS model of section 3. The formulation of section 3 was included with the following two expository goals, unattainable (in the authors' opinion) by presenting only the path-oriented formulation: to illustrate clearly *(i)* the geometry of the separation loss region shown in Fig. 6C as arising from gluing together the regions shown in Fig. 6B, and *(ii)* the complexity of routing, which would amount to searching for suitable continuous paths in a complicated topological object, obtained by topological gluings such as that shown notionally in Fig. 3.

Assume the data 1)-6), listed in section 3, are given.

1. For each agent $\alpha \in \mathcal{A}$, let $e^{INIT;\alpha}$ denote the edge occupied initially by agent α. (Two or more agents can occupy the same edge.)
2. Let $\mathcal{P}\left(e^{INIT;\alpha}\right)$ be the set of all paths in the multigraph that begin with the edge $e^{INIT;\alpha}$ and end with an edge that contains an acceptable destination for α. Denote the length of a path $p \in \mathcal{P}\left(e^{INIT;\alpha}\right)$ by $l(p)$.
3. Define each discrete mode as a mapping that assigns each agent α to a path in $\mathcal{P}\left(e^{INIT;\alpha}\right)$. In more detail, ν is a mapping from the set $\mathcal{A}$ of moving agents to the union $\cup_\alpha \mathcal{P}\left(e^{INIT;\alpha}\right)$ such that $\nu(\alpha) \in \mathcal{P}\left(e^{INIT;\alpha}\right)$ for each $\alpha \in \mathcal{A}$.
4. In each ν, have the arc length coordinate y^α_ν evolve according to the dynamical law

$$\dot{y}^\alpha_\nu = s^\alpha_\nu, \quad \alpha \in \mathcal{A}, \tag{2}$$

where s^α_ν is the control variable corresponding to the agent's speed of motion along the path.

5. For each agent α and each discrete mode μ, specify the speed constraint

$$S^{\alpha;\min}_\mu \le s^\alpha \le S^{\alpha;\max}_\mu, \tag{3}$$

where $[S^{\alpha;\min}_\mu, S^{\alpha;\max}_\mu]$ is the speed range allowed for agent α in discrete mode μ.

Definition 5.1. *The HCS defined in this section will be called a* path-based transportation HCS.

In a path-based transportation HCS, every discrete mode ν corresponds to a possible routing of the agents in $\mathcal{A}$, and every execution occurs in only one discrete mode. It follows that the discrete controls in a path-based transportation HCS are, effectively, identified with the discrete modes, and, in a Fully Routed Nominal Arrival problem, play but a trivial role.

6 A feasible trajectory algorithm for the Fully Routed Nominal Arrival Problem

6.1 Problem definition and geometric properties

In this section, we describe and analyze theoretically a subclass of the HCS defined above, the analysis being a foundation for the algorithm in section 6.5. This subclass represents a most commonly modeled scenario of air traffic, which is arriving in merging and, possibly, diverging flows, with no multiple routing options: each arriving aircraft follows without deviation a prescribed path in the route network. Consequently, the resulting path-based transportation HCS (definition 5.1) has only one discrete mode. Henceforth, the moving agents are aircraft. The subscript ν will be dropped for brevity.

Remark 6.1. *The assumption that each aircraft is assigned a route may not always hold. In cases when such assignment (known in Air Traffic Operations as* routing*) is not given, but is sought as part of the problem, the corresponding HCS has multiple discrete modes. The theory and algorithm developed in this section can then be used for each of those discrete modes, and the result computed in one of the modes selected later as final. See [12] for an analysis of computational costs and possibility of parallel computation of routing.*

The problem consists in computing a control strategy $\mathbf{s}(t) = (s^\alpha(t))_{\alpha\in\mathcal{A}}$ that will, in accordance with dynamical law (2), take each aircraft to the end of its prescribed path. To ATM operations, such a control strategy would serve as a speed advisory. This problem will be called *the Fully Routed Nominal Arrival problem*. It consists in finding feasible trajectories. No optimization is pursued.

Section 6.2 gives a list of assumptions, made to simplify the computations. As is shown below, these assumptions allow a solution by an algorithm that is non-iterative and proven to halt in polynomial time. Because of the urgency that can accompany ATM decisions, this bound on computation time is a highly desired feature of every solution candidate to be deployed as an operational tool in the field, especially for automated tools responsible for both separation assurance and scheduling.

6.2 Simplifying assumptions and their implications

Assumption 6.1. *All control strategies considered before section 7 will be assumed executable with certainty, i.e. without execution error that may be caused by wind, transportation performance, or human factors. In particular, under this assumption, the aircraft is able to follow its prescribed route exactly, without deviation.*

An approach to including execution uncertainty in the model is discussed in section 7.

Assumption 6.2. *All aircraft have the same cruising speed range,* $[S^{\min}, S^{\max}]$.

Assumption 6.3. *All aircraft pairs have the same required minimal separation distance, henceforth denoted* r.

Assumption 6.4. *During the execution of the algorithm described in section 6.5, below, no new aircraft enter the system.*

6.3 Geometry of the global continuous state space

Since the system is fully routed, each aircraft α is assigned a uniquely determined path, whose length will henceforth be denoted M^α, $\alpha = 1,\ldots,A$. Accordingly, the global state space Y for the problem can be obtained by removing from the parallelotope $\prod_{\alpha\in\mathcal{A}}[0,M^\alpha]$ all the conflicting states. We choose the coordinate system, for the whole of Y, so that the state variable y^α indicates the arc length coordinate of aircraft α on its path and ranges from 0 to M^α. Unless said otherwise, all state variables are regarded as coordinates referred to the standard basis $\mathbf{e}_1 = (1,0,\ldots,0,0),\ldots,\mathbf{e}_A = (0,0,\ldots,0,1)$.

The target set, the distal boundary, and protrusions

The vertex $\mathbf{M} = (M^1,\ldots,M^A)$ of Y will be called *the distal vertex* of Y. The faces of Y containing the distal vertex will be called *distal faces*. Distal faces of dimension 1 will be called *distal edges*. Since the distal edges correspond, one-to-one, to states in which all but one aircraft have exited the airspace, we define the target set of the fully routed nominal problem as the union of all distal edges. Fig. 7A illustrates the distal edges for 3 aircraft.

The union of all the $(A-1)$-dimensional faces adjacent to the distal vertex will be called *the distal boundary*. An illustration of the distal boundary for the case of 3 aircraft is given in Fig. 7B.

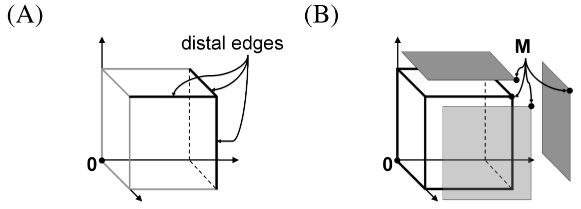

Fig. 7. The case of $A = 3$ aircraft: (A) The distal edges and the target set, and (B) an illustration of the distal boundary.

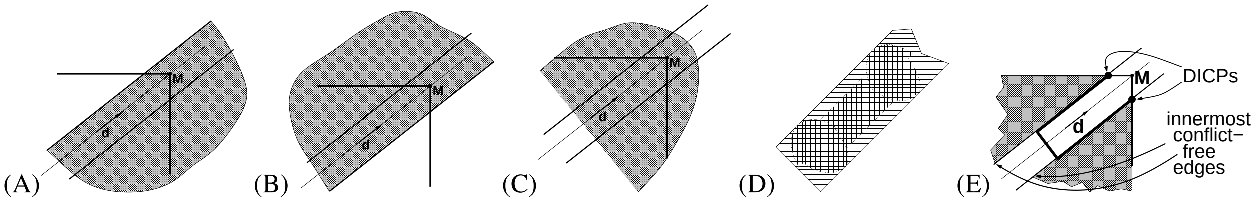

Fig. 8. The shaded regions are the orthogonal projections of the half-spaces (A) $H^{\alpha_1,\alpha_2;-1}$, (B) $H^{\alpha_1,\alpha_2;+1}$, and (C) F^{α_1,α_2} onto the $y^{\alpha_1}y^{\alpha_2}$-plane. Their intersection is the polygonal approximation, shown in Fig. 8D, of the conflict zone for α_1, α_2. (D) An approximation of a pairwise conflict zone (Fig. 6C) by a polygonal region equal to the intersection of three half-planes. (E) The geometry of the safe wedges (shaded) and DICPs.

6.4 Polygonally approximated conflict zones: geometric properties used in the algorithm

Each pairwise conflict zone will be approximated here by a polygon which is the intersection of three half-planes. This approximation achieves considerable computational advantage, but likely incurs a cost if an optimization were pursued: by enlarging the conflict zone, one restricts the class of considered candidate solutions. This loss would be most severe if at least two of the agents' path were to cross (the relevant operational concepts are *same courses*, *crossing courses*, and *opposite courses*; see [8, section 1-2-2].) In the framework of the above model, such a crossing can be viewed as a short overlap (compare to Fig. 6). To analyze the geometry of the conflicting state sets that result from this approximation in the global state space X, it will be convenient to introduce the technical terms of *protrusion*, *safe wedge*, *innermost conflict edge*, and *distal innermost conflict-free point*.

In the A-dimensional Euclidean space $\mathbf{R}^A$, define the unit vector $\mathbf{d} = \frac{1}{\sqrt{A}}(1, 1, \ldots, 1)$.

Let $H^{\alpha_1,\alpha_2;-1}$ and $H^{\alpha_1,\alpha_2;+1}$ denote, respectively, the half-spaces consisting of all states $\mathbf{y} = (y^1, \ldots, y^A)$ satisfying $y^{\alpha_1} - y^{\alpha_2} > -r^{\alpha_1,\alpha_2}$ and $y^{\alpha_1} - y^{\alpha_2} < r^{\alpha_1,\alpha_2}$, where r^{α_1,α_2} is the sum of the required minimal separation distance between aircraft α_1 and α_2 and a "buffer," which is a value sufficiently large to make sure that the obtained polygonal approximation contains all of the exact pairwise conflict zone depicted in Fig. 6C.

Define the half-space F^{α_1,α_2} to consist of all states $\mathbf{y}$ satisfying $\langle \mathbf{y} - \mathbf{M}, -\mathbf{d} \rangle < f^{\alpha_1,\alpha_2}$, where f^{α_1,α_2} is the orthogonal distance from $\mathbf{M}$ to the bounding hyperplane of F^{α_1,α_2} (Fig. 8). The pairwise conflict zone for the aircraft pair α_1, α_2 is, then, approximated by the *open* [29] polyhedral region

$$Z^{\alpha_1,\alpha_2} = H^{\alpha_1,\alpha_2;-1} \cap H^{\alpha_1,\alpha_2;+1} \cap F^{\alpha_1,\alpha_2}, \tag{4}$$

depicted in Fig. 8D. The distance f^{α_1,α_2} will be called *the protrusion* of the conflict zone (4).

Safe wedges, innermost conflict edges, and DICPs

The intersection $H^{\alpha_1,\alpha_2;-1} \cap H^{\alpha_1,\alpha_2;+1}$ is a *conflicting slab* of the form $|y^{\alpha_1} - y^{\alpha_2}| < r$. By "removing" from the A-dimensional Euclidean space $\mathbf{R}^A$ all the conflicting slabs (i.e., the slabs corresponding to all pairs of aircraft), one obtains a pairwise disjoint collection of $A!$ polyhedral regions, called *safe wedges*. The intersection of a safe wedge with the global continuous state space is a conflict-free region and has a 1-dimensional face (a line) parallel to $\mathbf{d}$. This line will be called *an innermost conflict-free edge*. The intersection of this edge with a distal face of dimension $(A-1)$ is a point called *a distal innermost conflict-free point (DICP)*. Safety wedges and DICPs for the case of $A = 2$ aircraft are illustrated in Fig. 8E.

Consider "the positive A-dimensional quadrant" $\mathbf{R}^A_{\geq 0} = \{(y^1, y^2, \ldots, y^A) : y^\alpha \geq 0 \text{ for all } \alpha\}$ of the A-dimensional Euclidean space $\mathbf{R}^A$. For each permutation σ of the indices $1,\dots,A$, the set of all $\mathbf{y} = (y^1, y^2, \dots, y^A)$ such that

$$0 \le y^{\sigma(1)} < y^{\sigma(2)} < \dots < y^{\sigma(A)} \tag{5}$$

determines a subset Y_σ of the “quadrant” $\mathbf{R}^A_{\ge 0}$.

Definition 6.1. *The subset Y_σ (in fact, the interior of a polyhedral angle) will henceforth be called* the basic polyhedral angle corresponding to the permutation σ.

Lemma 6.1. *If the permutation σ is such that the safe wedge W consists of all $\mathbf{y} = (y^1, y^2, \dots, y^A)$ with*

$$y^{\sigma(\alpha)} \le r + y^{\sigma(\alpha+1)} \quad \text{for } \alpha = 1, 2, \dots, A-1, \tag{6}$$

then W is contained in Y_σ.

Proof. Condition (5) follows from (6).

Theorem 6.1. *If all the required minimal distances for in-trail separation have the same value, r, then the coordinate A-tuple of every DICP is the sum of $\mathbf{M}$ and a permutation of $0, -r, -2r, \dots, -(A-1)r$, and, conversely, every such permutation gives a DICP.*

Proof. These A-tuples are exactly those at which all constraints (6) are active.

The cone of attainability, feasible trajectories, and feasible solutions

The flow of the traffic consisting of A aircraft $\alpha = 1,\dots,A$ can be seen as a trajectory $(y^1(t), y^2(t), \dots, y^A(t))$ of the inertia-free dynamical law (2).

Remark 6.2. *Throughout this paper, the term* collective trajectory *(or, simply,* trajectory*) will mean a time-parameterized curve $\mathbf{y}(t) = (y^1(t), y^2(t), \dots, y^A(t))$ in the global continuous state space X that satisfies (2) for some choice of a control strategy $\mathbf{s}(t) = (s^1(t), \dots, s^A(t))$. This usage of the term* trajectory *differs from that commonly seen in Air Traffic Management.*

Recall assumption 6.2. A segment $\mathbf{y}_0\,\mathbf{y}$ in X is said to be *attainable* if all aircraft pairs (α_1, α_2) satisfy

$$\frac{S^{\min}}{S^{\max}}(y^{\alpha_2} - y_0^{\alpha_2}) \le y^{\alpha_1} - y_0^{\alpha_1} \le \frac{S^{\max}}{S^{\min}}(y^{\alpha_2} - y_0^{\alpha_2}) \tag{7}$$

A ray emanating from $\mathbf{y}_0$ is said to be *attainable* if for every point $\mathbf{y}$ on the ray the segment $\mathbf{y}_0\mathbf{y}$ is attainable. The union of all attainable rays emanating from $\mathbf{y}_0$ is a polyhedral cone, called *the cone of attainability at* $\mathbf{y}_0$ and denoted $C(\mathbf{y}_0)$. This cone is a pointed cone with vertex $\mathbf{y}_0$. A collective trajectory $\mathbf{y}(t)$ with initial state $\mathbf{y}(0) = \mathbf{y}_0$ is said to be *attainable* if it is piecewise differentiable and, at every point $\mathbf{y}$ of differentiability, the corresponding tangent vector lies in the cone of attainability at $\mathbf{y}$. A conflict-free attainable trajectory is said to be *feasible*. A 2-aircraft example is shown notionally in Fig. 9; a 3-aircraft example, in Fig. 10.

Definition 6.2. *A feasible trajectory whose final state is in the target set is called* a feasible solution.

The following lemma characterizes all the collective states from which it is safe to have all aircraft fly at the same speed.

Lemma 6.2. *If $\mathbf{y}_0$ is a DICP, then there exists a feasible solution with initial state $\mathbf{y}_0$. Consequently, a feasible trajectory with the final state a DICP can be extended to a feasible solution.*

Let $|\cdot|_\infty$ denote the *max* norm, defined for a vector $\xi = (\xi^1, \xi^2, \dots, \xi^A)$ in $\mathbf{R}^A$ by $|\xi|_\infty = \max_\alpha |\xi^\alpha|$.

Proof. The collective trajectory $\mathbf{y}(t) = \mathbf{y}_0 + tS^{\max}\frac{1}{|\mathbf{d}|_\infty}\mathbf{d}$, with the α-th component excluded once aircraft α exits the global continuous state space, is conflict-free since $\mathbf{y}_0$ lies in a safe wedge and since $\dot{\mathbf{y}}$ is co-directional with $\mathbf{d}$.

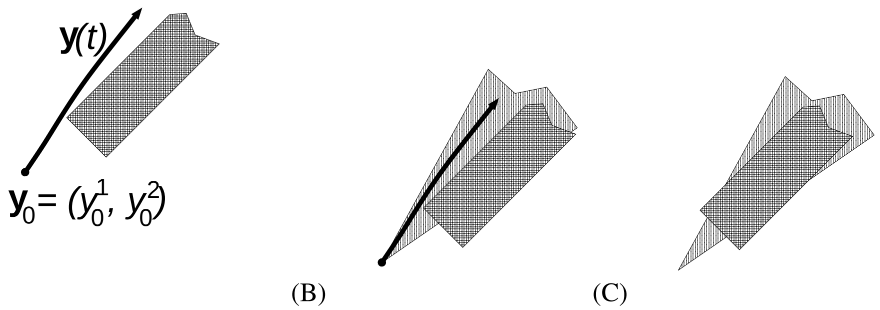

Fig. 9. An example with two aircraft. The conflict zone is as in Fig. 8D. (A) A collective state trajectory $\mathbf{y}(t)$, with initial state $\mathbf{y}(0) = \mathbf{y}_0$. (B) The cone of attainability $C(\mathbf{y}_0)$ (shown in the lighter shade) at $\mathbf{y}_0$. (C) The cone is positioned so that there are no feasible solutions.

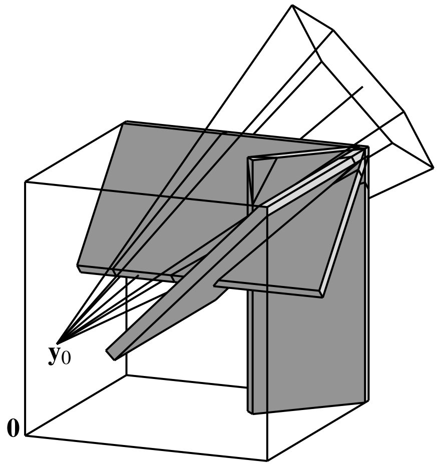

Fig. 10. An example with $A = 3$ aircraft (compare with Fig. 7). There are $\binom{A}{2} = 3$ conflict zones. The orthogonal projection of zone Z^{α_1,α_2} onto the $y^{\alpha_1}y^{\alpha_1}$-coordinate plane (onto the appropriate face of the parallelotope) has the form shown, as a shaded region, in Fig. 9A.

The geometry of FCFS: the closest DICP to y corresponds to the safe wedge containing y

The key result of this section, theorem 6.2, tells how to find (in time $O(A^2)$) a DICP that requires "minimal position shifting" to be performed on the current collective state.

Lemma 6.3. *If* $\mathbf{a} = (a^1, a^2)$ *and* $\mathbf{b} = (b^1, b^2)$ *are vectors in the first octant of the 2-dimensional Euclidean space* $\mathbf{R}^2$ *with Cartesian coordinates* y^1, y^2*, i.e. satisfy* $0 < a^1 < a^2$ *and* $0 < b^1 < b^2$*, then*

$$a^1b^1 + a^2b^2 > a^1b^2 + a^2b^1 \qquad (8)$$

Proof. Both $\mathbf{a}$ and $\mathbf{b}$ lie in the positive quadrant of the plane and below the diagonal $y^1 = y^2$. Therefore, the angle between $\mathbf{a}$ and $\mathbf{b}$ is smaller than that between $\mathbf{a}$ and $\mathbf{b}' = (b^2, b^1)$ (the latter point being above the diagonal). Since all angles are between 0 and $\pi/2$, one has $\langle \mathbf{a}, \mathbf{b} \rangle > \langle \mathbf{a}, \mathbf{b}' \rangle$, which is inequality (8).

Lemma 6.4. *Suppose* $\mathbf{a} = (a^1, \ldots, a^A)$ *and* $\mathbf{b} = (b^1, \ldots, b^A)$ *are vectors, with all standard components distinct, in the A-dimensional Euclidean space* $\mathbf{R}^A$ *such that* $0 < a^1 < a^2 < \ldots < a^A$ *and* $0 < b^\alpha$ *for* $\alpha = 1, \ldots, A$*. For each permutation* σ *of* $\{1, 2, \ldots, A\}$*, let* $\sigma\mathbf{b} = (b^{\sigma(1)}, b^{\sigma(2)}, \ldots, b^{\sigma(A)})$*. Then, of the products* $\langle \mathbf{a}, \sigma\mathbf{b} \rangle$ *corresponding to all the different permutations* σ*, a largest one is delivered by that permutation* σ_* *for which* $b^{\sigma_*(1)} < b^{\sigma_*(2)} < \ldots < b^{\sigma_*(A)}$.

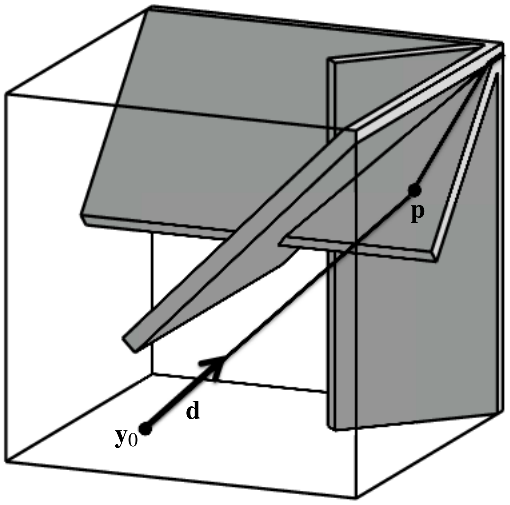

Fig. 11. An example illustrating lemma 6.5. There are $A = 3$ aircraft, and the initial state $\mathbf{y}_0$ lies in a safe wedge. By going from $\mathbf{y}_0$ in the direction $\mathbf{d}$, one reaches the distal boundary (in state $\mathbf{p}$) without running into a conflict zone.

Proof. Applying *BubbleSort* [35] to the numerical sequence

$$b^1, \ldots, b^A, \tag{9}$$

one obtains a sequence $\sigma_1, \sigma_2, \ldots, \sigma_{J-1}, \sigma_J = \sigma_*$ of permutations of $\{1, 2, \ldots, A\}$, every two consecutive ones differing by a transposition that puts two elements of (9) in increasing order. By lemma 6.3, $\langle \mathbf{a}, \sigma_{j-1}\mathbf{b}\rangle < \langle \mathbf{a}, \sigma_j \mathbf{b}\rangle$ for $j = 2, 3, \ldots, J$. Thus, the largest of the products $\langle \mathbf{a}, \sigma_j \mathbf{b}\rangle$ is given by $\sigma_J = \sigma_*$.

The following helps locate a DICP closest to a given $\mathbf{y}$.

Theorem 6.2. *If $\mathbf{y} = (y^1, \ldots, y^A)$ and a DICP $\mathbf{p} = (p^1, \ldots, p^A)$ lie in the same basic polyhedral angle, then $\mathbf{p}$ is a closest (in the sense of Euclidean distance) DICP to $\mathbf{y}$.*

Proof. For this proof, change coordinates so that $\mathbf{M}$ is now the origin, and $\mathbf{d}$ points along the positive diagonal. If $\mathbf{q}$ were a DICP closer to $\mathbf{y}$ than $\mathbf{p}$ is, then the permutation σ such that $\mathbf{q} = \sigma\mathbf{p}$ would satisfy $\langle \sigma\mathbf{p}, \mathbf{y}\rangle > \langle \mathbf{p}, \mathbf{y}\rangle$, contradicting lemma 6.4.

Existence and nonexistence of feasible solutions

The following lemma and its proof are based on the intuition that starting in a state $\mathbf{y}_0$ in a safe wedge and going in the direction $\mathbf{d}$, one will reach the distal boundary without running into any conflict zones; see Fig. 12.

Lemma 6.5. *(A sufficient condition for the existence of a feasible solution.) If $\mathbf{y}_0 = (y_0^1, \ldots, y_0^A)$ is in a safe wedge, then there exists a feasible solution with initial state $\mathbf{y}_0$.*

Proof. Consider the trajectory $\mathbf{y}(t) = \mathbf{y}_0 + t\bar{s}\frac{1}{|\mathbf{d}|_\infty}\mathbf{d}$, with t increasing to the value T for which the state $\mathbf{p} = \mathbf{y}(T)$ lies on the distal boundary. By construction, the trajectory is feasible, and $\mathbf{y}(T)$ lies in a safe wedge. The state $\mathbf{y}_T$, taken to be the initial state for a new, $(A-1)$-dimensional, problem, lies in a safe wedge of the new ambient $(A-1)$-dimensional space. (For, in the state $\mathbf{y}_T$, at least one aircraft, say α-th, has exited the airspace, so all the pairwise conflict zones involving α are removed.) Consequently, the lemma holds by induction on A.

Theorem 6.3. *(A sufficient condition for the absence of feasible solutions.) If a pairwise conflict zone (4) (not necessarily with a largest protrusion) is such that $C(\mathbf{y}_0) \cap \partial F^{\alpha_1,\alpha_2}$ is contained entirely in the conflicting slab $H^{\alpha_1,\alpha_2;-1} \cap H^{\alpha_1,\alpha_2;+1}$ (Fig. 9C), then there is no feasible solution with initial state $\mathbf{y}_0$.*

Proof. In a cone $C(\mathbf{y}_0)$ so positioned, every attainable collective trajectory reaches a conflicting state before reaching the target set.

Table 1. The output format of an advisory segment.

a/c 1 speed	a/c 2 speed	…	a/c A speed	TTF

6.5 A polynomial-time speed control algorithm for computing a feasible solution or reporting that none was found

Given: An initial state $\mathbf{y}_0 = \left(y^1, \ldots, y^A\right)$, the global continuous state space (completely determined by $(\mathbf{M} - \mathbf{y}_0)$), and the pairwise conflict zones (4).

	Steps and explanatory remarks	**Computational cost in the big-O notation [35]**
1.	Find all conflicting slabs that contain $\mathbf{y}_0$.	$\binom{A}{2} = O\left(A^2\right)$
2.	If $\mathbf{y}_0$ lies in no conflicting slab, i.e. lies in a safe wedge, exit the algorithm, returning the speed advisory constructed in lemma 6.5.	$O(A)$
3.	Test whether the collective state $\mathbf{y}_0$ is in any of the pairwise conflict zones (4). If it is, go to step 8.	$O\left(A^2\right)$
	(For steps 4, 5, 7, refer to Fig. 12).	
4.	Find a DICP $\mathbf{p}$ closest to $\mathbf{y}_0$.	By theorem 6.2, the computational cost of this step is $O(A \log_2 A)$
5.	Compute the state $\mathbf{y}_1 = \mathbf{p} + \alpha \mathbf{d}$ that lies on the boundary of $C(\mathbf{y_0})$.	$O(A)$
	Remark 6.3. *By construction, the state* $\mathbf{y}_1$ *lies in a safe wedge, hence the segment* $\mathbf{y}_1\mathbf{p}$ *is a feasible trajectory (lemma 6.5; see also Fig. 12). If segment* $\mathbf{y}_0\mathbf{y}_1$ *is feasible, then the trajectory* $\mathbf{y}_0\mathbf{y}_1\mathbf{p}$ *is feasible and, by lemma 6.2, extends to a feasible solution.*	
6.	If segment $\mathbf{y}_0\mathbf{y}_1$ is feasible, exit the algorithm, returning the speed advisory for the trajectory described in remark 6.3. Computations are required to check the feasibility of $\mathbf{y}_0\,\mathbf{y}_1$.	$O\left(A^2\right)$
7.	(In this step, let Γ denote the hyperplane orthogonal to $\mathbf{d}$ and passing through the origin. Orthogonal projection onto Γ will also be denoted by Γ. Refer to Fig. 12 for an illustration of a case with $A = 3$.) Segment $\mathbf{y}_0\,\mathbf{y}_1$ has been found infeasible. Use a fixed number of iterations of the Bisection Method to look for the parameter β that determines the point $\mathbf{q} = \beta\,\Gamma\mathbf{p}$ such that the point $\mathbf{y}_{1;*}$, defined as the intersection between the line through $\mathbf{q}$ parallel to $\mathbf{d}$ and the boundary of $C(\mathbf{y}_0)$, yields a feasible segment $\mathbf{y}_0\,\mathbf{y}_{1;*}$. If this segment is found, return a speed advisory for the trajectory $\mathbf{y}_0\,\mathbf{y}_{1;*}\mathbf{p}$ (analogously to step 6). Otherwise, go to step 8.	$O(A)$
8.	Exit the algorithm with the message, “Cannot produce a conflict-free speed advisory by speed control only.”	$O(1)$

6.6 Sample numerical solutions

If a feasible speed advisory is found by the above algorithm successfully, it is output as one or more rectilinear segments, each called an *advisory segment* and conforming to the format shown in table 1. Here TTF stands for the *time to fly (with that speed)* and indicates the time duration in which the advisory segment is to be traversed. However, in the numerical

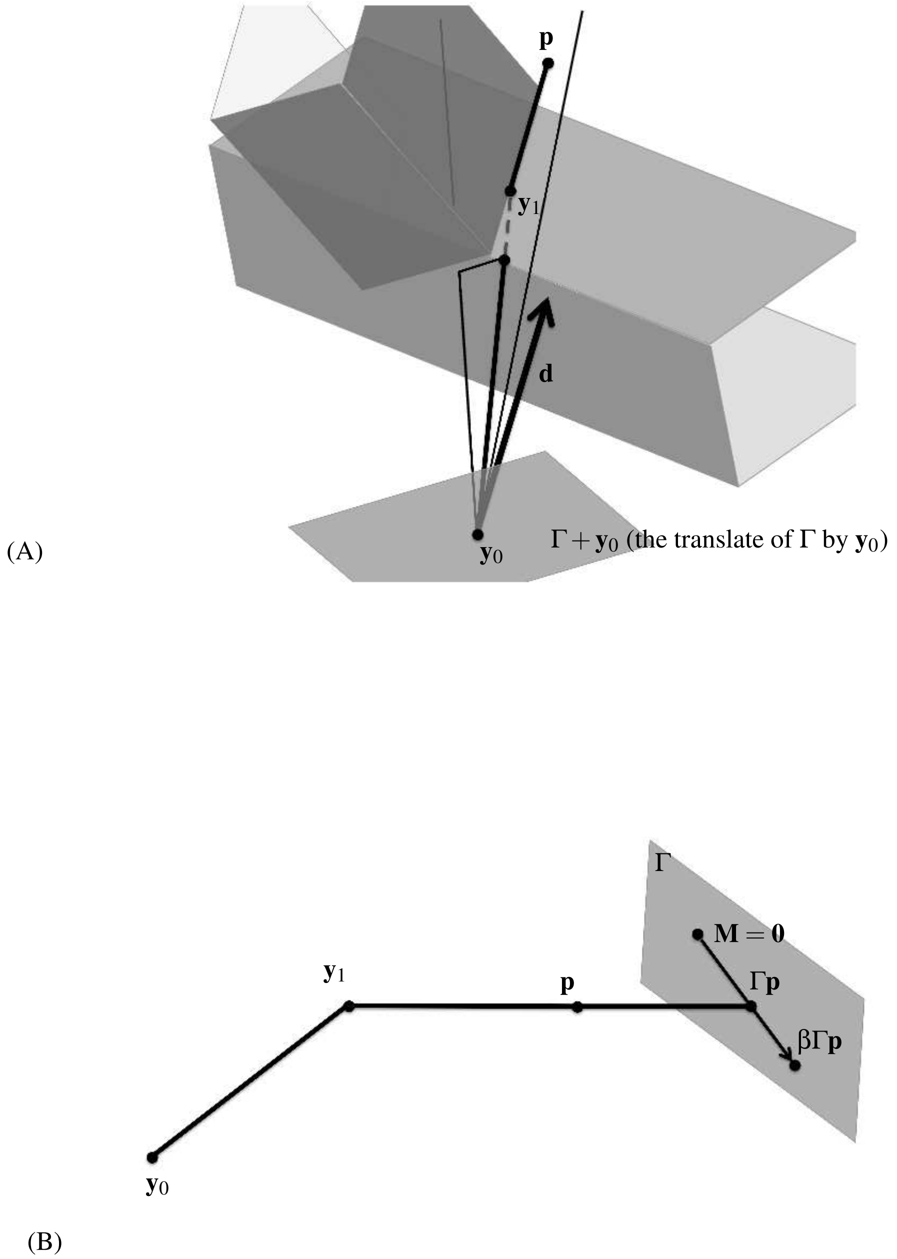

Fig. 12. An example, with $A = 3$ aircraft, illustrating the algorithm. In steps 4, 5, one finds a DICP $\mathbf{p}$ closest to $\mathbf{y}_0$ and proceeds from $\mathbf{p}$ along the direction of $-\mathbf{d}$ until reaching a state $\mathbf{y}_1$ on the boundary of $C(\mathbf{y}_0)$. The segment $\mathbf{y}_0\mathbf{y}_1$ may or may not be feasible. The latter case is illustrated in (A) (the infeasible part of $\mathbf{y}_0\mathbf{y}_1$ is shown dashed) and is handled by step 7, which is illustrated in (B). If $\mathbf{y}_0\mathbf{y}_1$ had turned out feasible, the piecewise linear trajectory $\mathbf{y}_0\mathbf{y}_1\mathbf{p}$ would have constituted a first portion of a solution, and this would have been detected and handled in step 6.

results presented below (table 2), instead of showing all the individual speeds, the following abbreviated format will be used:

min. of the A speeds	max. of the A speeds	TTF

(10)

The above algorithm was applied to a number of arriving traffic scenarios in the portion of the LAX airspace shown in Fig. 1. Although all the paths in this airspace portions converge to one runway, such convergence is not a requirement: the algorithm is applicable to all situations where the aircraft follow pre-determined paths. The solutions found for two of the scenarios, each with two advisory segments, are summarized in Table 2. The initial states are conflict-free. One of the two

Table 2. Each of the tables gives a 2-segment speed advisory and TTF for an arriving traffic scenarios. In columns 4-6 of each table, each line fits format (10).

# a/c	$\mathbf{y}_0$	advis. seg.	min. speed (kts)	max. speed (kts)	seg. TTF (hr)
23	Fig. 13	1	211.84	250.00	0.56
		2	250.00	250.00	0.41

# a/c	$\mathbf{y}_0$	advis. seg.	min. speed (kts)	max. speed (kts)	seg. TTF (hr)
24	not shown	1	199.11	250.00	0.59
		2	250.00	250.00	0.44

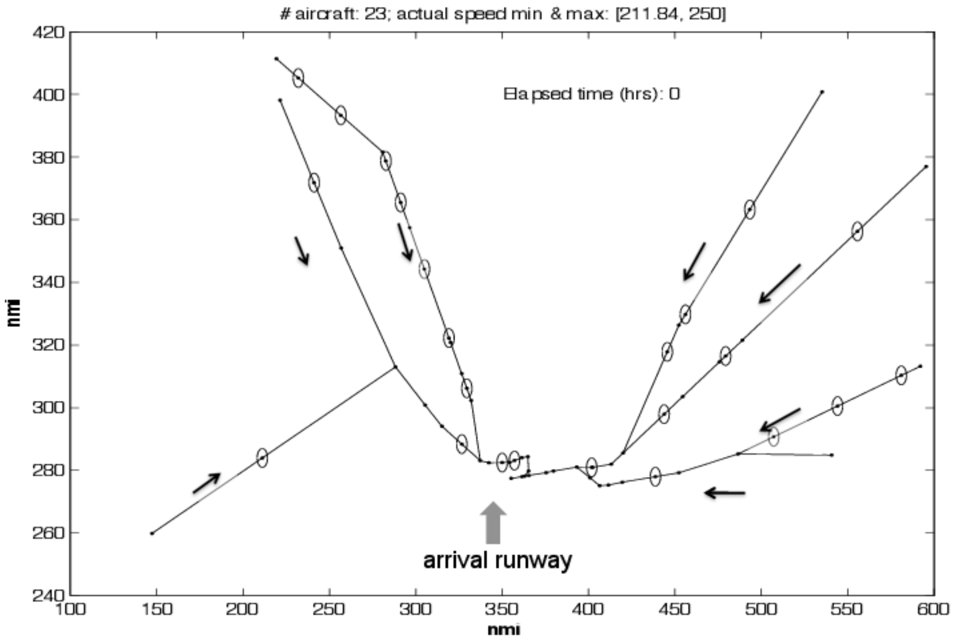

Fig. 13. The initial state of a traffic scenario for 23 aircraft. To capture airspace adequately, an aspect ratio different from 1 is used. The thin arrows indicate the directions of the traffic.

scenarios is depicted in Fig. 13, with circles of radius r centered at the aircraft.

7 Discussion

The above HCS framework addresses simultaneously the following issues in Air Traffic Management and adjacent fields: scheduling, speed advisory generation, conflicts (avoidance, detection, and recovery), computational efficiency, and algorithm correctness. The theory and algorithm given above are mainly a proof of concept, intended to demonstrate how HCS can be used to obtain provably correct solutions efficiently. The following are some directions for further research.

Control equations with inertia: A model more realistic than the dynamical law (2) should include inertia, thus making the agents's *acceleration* (and not speed) the quantity directly controllable.

Allowing new aircraft to enter the airspace during the computation: A reasonable direction is a search for geometric insight into those entrances for which the original, A-dimensional, problem has a feasible solution but the newly obtained one does not. The problem above, together with the Scheduled Routing Problem formulated in [12], suggest that such insight would be valuable not only in air traffic management and in airspace design, but also in the general field of multi-agent coordination. In published literature, entrance of new aircraft into the system has been considered in the context of the *dynamic Aircraft Scheduling (Landing) Problem* (see [26]), which, however, does not consider separation except at the runway.

Conflict recovery: Recovering from a conflicting state $\mathbf{y}_c$ in the above model is equivalent to finding–or establishing the absence of–a shortest (or otherwise optimal) attainable trajectory from $\mathbf{y}_c$ to a nearest safe wedge. Theorem 6.2 is instrumental here, as a nearest safe wedge is with a nearest DICP, and the latter is computable in time polynomial in A.

Path control (in ATM operational terminology, Vectors): In operational ATM practice, an aircraft is often allowed, and even instructed, to deviate from a given route. Such deviations are known in Air Traffic operations as *path control*. This deformation of the given route results in a deformation of the global continuous state space and of the pairwise conflict zones. Insight into the relation between the two deformations can be instrumental in extending the model to include path control.

Modeling uncertainty of control execution: Control of aircraft is affected by such sources of uncertainty as human factors, aircraft performance, and weather. In the context of the above HCS, such uncertainty generally leads to stochastic dynamical law [36, chapter 14]. Among the most common sources of random perturbation to an aircraft's execution of control is wind. In the special case when each aircraft is cleared for a procedure that prescribes a specific route consisting of rectilinear segments, connected by fixed-radius turns (e.g., *RF-leg types* [37]), the aircraft's navigation system will keep the aircraft on the route even in the presence of wind. Therefore, the wind affects only those components of the aircraft's speed and acceleration tangential to the route. This suggests a stochastic dynamical law, for the Fully Routed Problem, that includes inertia (i.e., the control variable is the acceleration), governs the arc length coordinate y^α and the arc length speed s^α as stochastic processes serving as the state variables, and has the form

$$dy^\alpha = s^\alpha\, dt, \quad ds^\alpha = a^\alpha\, dt + a_w\, dt + dW^\alpha, \quad \alpha \in \mathcal{A}$$

where a_w is the (possibly time-dependent) mean of the tangential component of the acceleration due to wind, and W is a Wiener process that characterizes the fluctuations of the acceleration due to wind about the mean.

Acknowledgment

D. Davis's research was supported by NSF Grant DGE-0707424. The authors thank D. Denery (NASA ARC) for suggesting the use of hybrid systems for ATM, W. Haskell (USC) for editing early drafts of the paper, H. Swenson (NASA ARC) and J. Mitchell (SUNY Stony Brook) for helpful suggestions.

References

[1] Carr, G. C., Erzberger, H., and Neuman, F., 1999. "Delay exchanges in arrival sequencing and scheduling". *Journal of Aircraft,* **36**, pp. 785–791.

[2] Bayen, A., Tomlin, C., Ye, Y., and Zhang, J., 2003. "An approximation algorithm for scheduling aircraft with holding time". In IEEE Conference on Decision and Control (CDC).

[3] Roy, K., Bayen, A., and Tomlin, C., 2005. "Polynomial time algorithms for scheduling of arrival aircraft". In AIAA Guidance, Navigation, and Control Conference and Exhibit.

[4] Bianco, L., Dell'Olmo, P., and Giordani, S., 2006. "Scheduling models for air traffic control in terminal areas". *J. Scheduling,* **9**(3), pp. 223–253.

[5] Balakrishnan, H., and Chandran, B., 2006. "Scheduling aircraft landings under constrained position shifting". In AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO.

[6] Lee, H., and Balakrishnan, H., 2008. "A study of tradeoffs in scheduling terminal-area operations". *Proceedings of the IEEE,* **96**(12), dec., pp. 2081 –2095.

[7] Chipalkatty, R., Twu, P., Rahmani, A., and Egerstedt, M., 2010. "Distributed scheduling for air traffic throughput maximization during the terminal phase of flight". In Decision and Control (CDC), 2010 49th IEEE Conference on, pp. 1195 –1200.

[8] U.S.A., 2010. *Order JO 7110.65T, Air Traffic Control*. U.S. Dept. of Transportation, Federal Aviation Administration, Washington, D.C.

[9] Bradley, S., Hax, A., and Magnanti, T., 1977. *Applied Mathematical Programming*. Addison Wesley.

[10] Vazirani, V. V., 2004. *Approximation Algorithms*. Springer, Mar.

[11] Bayen, A. M., Zhang, J., Tomlin, C. J., and Ye, Y., 2003. MILP formulation and polynomial time algorithm for an aircraft scheduling problem.

[12] Sadovsky, A., Davis, D., and Isaacson, D., 2012. "Optimal routing and control of multiple agents moving in a transportation network and subject to an arrival schedule and separation constraints". *In review*.

[13] Sussmann, H. J., 1999. "A maximum principle for hybrid optimal control problems". *The 38th IEEE Conference on Decision and Control*, pp. 425–430.

[14] Sussmann, H. J., 1999. "A nonsmooth hybrid maximum principle". *The First European Community Nonlinear Control Network Workshop on 'Stability and Stabilization'*, pp. 325–354.

[15] Tomlin, C., Mitchell, I., Bayen, A. M., and Oishi, M., 2003. "Computational techniques for the verification of hybrid systems". *Proceedings of the IEEE,* **91**(7), pp. 986–1001.

[16] Branicky, M., 2005. *Introduction to hybrid systems*. Birkhauser.

[17] Simic, S., Johansson, H., Sastry, S., and Lygeros, J., 2005. "Towards a geometric theory of hybrid systems". *Dynamics of Continuous, Discrete, and Impulsive Systems, Series A*.

[18] Dmitruk, A., and Kaganovich, A., 2008. "The Hybrid Maximum Principle is a consequence of Pontryagin Maximum Principle". *Systems and Control Letters,* **57**(11), pp. 964 – 970.

[19] Passenberg, B., Caines, P. E., Sobotka, M., Stursberg, O., and Buss, M., 2010. "The minimum principle for hybrid systems with partitioned state space and unspecified discrete state sequence". In in Proc. of the 49th IEEE Conf. on Decision and Control.

[20] Sethian, J., and Vladimirsky, A., 2002. "Ordered upwinds methods for hybrid control". *Proceedings Fifth International Conference on Hybrid Systems and Control, LCNS 2289*.

[21] D. R. Isaacson and J. E. Robinson III and H. Swenson and D. Denery, 2010. "A concept for robust, high density terminal air traffic operations". In 10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Fort Worth, TX.

[22] Swenson, H. N., Hoang, T., Engelland, S., Vincent, D., Sanders, T., Sanford, B., and Heere, K., 1997. "Design and operational evaluation of the traffic management advisor at the fort worth". In 1st USA/Europe Air Traffic Management R&D Seminar, Saclay, France, pp. 17 – 19.

[23] Love, D., Horn, W., and Fellman, L., 2002. "An evaluation of the use of the URET prototype tool at indianapolis and memphis air route traffic control centers, using benefit metrics". In Digital Avionics Systems Conference, 2002. Proceedings. The 21st, Vol. 1, pp. 1A1–1 – 1A1–12 vol.1.

[24] Neuman, F., and Erzberger, H., 1991. *Analysis of Delay Reducing and Fuel Saving: Sequencing and Spacing Algorithms for Arrival Traffic*. NASA TM-103880.

[25] Sadovsky, A., Swenson, H., Haskell, W., and Rakas, J., 2011. "Optimal time advance in terminal area arrivals: Throughput vs. fuel savings". In 30th Digital Avionics Systems Conference (DASC), Seattle, WA.

[26] Isaacson, D., Sadovsky, A., and Davis, D., 2012. "Scheduling for future terminal air traffic operations: Problem statement and review of prior research". *To be submitted*.

[27] U.S.A., 2007. Concept of operations for the next generation air transportation system, version 2.0.

[28] Papadimitriou, C. H., and Steiglitz, K., 1998. *Combinatorial Optimization; Algorithms and Complexity*. Dover Publications.

[29] Korn, G., and Korn, T., 1961. *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review*. McGraw-Hill, New York.

[30] Ghrist, R., O'Kane, J. M., and LaValle, S. M., 2005. "Computing pareto optimal coordinations on roadmaps". *The International Journal of Robotics Research,* **24**(11), pp. 997–1010.

[31] Adams, C., and Franzosa, R., 2008. *Introduction to topology: pure and applied*. Pearson Prentice Hall.

[32] Arnol'd, V., 1997. *Mathematical Methods of Classical Mechanics*, 2nd ed. Springer.

[33] Jung, J. B., and Ghrist, R., 2008. "Pareto optimal multi-robot coordination with acceleration constraints". In Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, pp. 1942 –1947.

[34] Sadovsky, A., to appear. "Application of the shortest-path problem to routing terminal airspace air traffic". *AIAA Journal of Aerospace Information Systems*.

[35] Cormen, T., Leiserson, C., and Rivest, R., 1990. *Introduction to Algorithms*. The MIT Press, Cambridge, MA.

[36] Bryson, A. E., and Ho, Y. C., 1969. *Applied optimal control*. New York: Blaisdell.

[37] ARINC, T. C. G., 2011. 424-20 navigation system database.