A Satisficing Approach to Aircraft Conflict Resolution

James K. Archibald, *Senior Member, IEEE*, Jared C. Hill, Nicholas A. Jepsen, Wynn C. Stirling, and Richard L. Frost

***Abstract*—Future generations of air traffic management systems may give appropriately equipped aircraft the freedom to change flight paths in real time. This would require a conflict avoidance and resolution scheme that is both decentralized and cooperative. We describe a multiagent solution to aircraft conflict resolution based on satisficing game theory. A key feature of the theory is that satisficing decision makers form their preferences by taking into consideration the preferences of others, unlike conventional game theory that models agents that maximize self-interest metrics. This makes possible situational altruism, a sophisticated form of unselfish behavior in which the preferences of another agent are accommodated provided that the other agent will actually take advantage of the sacrifice. This approach also makes possible the creation of groups in which every decision maker receives due consideration. We present simulation results from a variety of scenarios in which the aircraft are limited to constant-speed heading-change maneuvers to avoid conflicts. We show that the satisficing approach results in behavior that is attractive both in terms of safety and performance. The results underscore the applicability of satisficing game theory to multiagent problems in which self-interested participants are inclined to cooperation.**

***Index Terms*—Conflict resolution, decision making, distributed control, multiagent systems, satisficing games.**

I. Introduction

INEFFICIENCIES in the current air traffic control (ATC) system cost the airline industry billions of dollars annually in delays and wasted fuel [1], [2], and the burning of unnecessary fuel contributes to atmospheric pollution [3]. The desire to reduce operational expenses has motivated investigations into alternative ATC approaches. For example, in *free flight*, pilots would be allowed to modify their flight path in real time [4]. As a consequence, much of the responsibility for ensuring safety—maintaining adequate separation between aircraft—would shift from centralized ground control to pilots and onboard decision support systems. Such systems would utilize navigation aids, communication technologies, and computing infrastructure to detect and resolve projected conflicts before proximity violations can occur.

This paper presents a decentralized, multiagent approach to the resolution of conflicts in *enroute* airspace, the open airspace between airports. This space is an attractive candidate for automation because it does not require rigid scheduling and because aircraft density is lower than in the airspace immediately surrounding airports. As in preliminary versions of this research [5]–[7], we assume that each aircraft is aware of critical information (e.g., position, velocity, and destination) pertaining to each aircraft within a given communication radius. To allow comparison of our approach with previously published studies, we adopt a commonly employed 2-D model of airspace in which all aircraft fly at the same altitude and at constant speed. Conflicts can thus be avoided only through heading-change maneuvers. While maneuvers involving velocity and altitude changes would be critical in any complete ATC system, the difficulty of maintaining aircraft separation is *increased* if they are excluded. Thus, for high traffic densities, the model constitutes a formidable challenge for any conflict resolution technique.

As reviewed in Section V, many techniques have been proposed for resolving conflicts between aircraft, from heuristic path-planning algorithms to formalized optimal conflict resolution approaches. Although these approaches differ widely, they all seek the “best” solution, even if it can only be approximated. Conceptually, they rank-order the set of possible actions, and then select the action with the highest ranking, subject to appropriate constraints. The approach described in this paper is fundamentally different: each aircraft determines the set of *acceptable* avoidance maneuvers it can perform, obtained by eliminating from the full set of options as many bad choices as possible, based on safety and efficiency concerns. The remaining alternatives are deemed to be “good enough,” or *satisficing*. Essentially, a satisficer is a cautious optimizer who, rather than insisting on a single “best” solution, retains an enlarged view containing all reasonably acceptable solutions. We show that our satisficing approach compares favorably with conflict resolution techniques described in the literature.

There are several reasons why a satisficing decision mechanism is advantageous for distributed multiagent control in general, and aircraft conflict resolution in particular. First, there is no need to posit a single definition of “best,” which, in the absence of centralized control and global knowledge, varies from the perspective of each participant. Instead, satisficing permits agents—individual aircraft in this case—to extend their spheres of interest beyond the self. This enables complex social behavior such as altruism, where agent A defers to agent B because B’s immediate need is greater, even though it increases A’s personal expense. Second, satisficing permits group and individual interests to be reconciled in a single, coherent mathematical structure. Third, it provides a constructive way for the participants to negotiate and reach an acceptable compromise. Fourth, satisficing agents are not restricted to pairwise consideration of projected conflicts; many can be addressed in a single avoidance maneuver.

Manuscript received February 6, 2007; revised August 14, 2007. This paper was recommended by Editor V. Marik.

J. K. Archibald, W. C. Stirling, and R. L. Frost are with the Department of Electrical and Computer Engineering, Brigham Young University, Provo UT 84602 USA (e-mail: jka@ee.byu.edu; wynn@ee.byu.edu; rickf@ee.byu.edu).

J. C. Hill is with Rincon Research Corporation, Tucson, AZ 85711 USA (e-mail: jayrodhill@gmail.com).

N. A. Jepsen is with the J. Reuben Clark Law School, Brigham Young University, Provo, UT 84602 USA (e-mail: nicholas.jepsen@gmail.com).

Digital Object Identifier 10.1109/TSMCC.2008.919162

1094-6977/$25.00 © 2008 IEEE

The next section describes the theoretical foundation for this paper. Section III describes two conflict resolution models constructed within the satisficing framework. An example conflict is discussed in Section IV to illustrate the operation of the resolution techniques. Section V summarizes previously published approaches to conflict resolution, and measures of system performance are discussed in Section VI. Section VII describes the simulator used and presents simulation results from a variety of conflict scenarios. Finally, Section VIII summarizes our findings and concludes the paper.

II. Theoretical Background

Any mathematical formalization of a decision problem requires two basic components. First, a concept of preference must be defined, and second, a solution concept must be applied to identify an acceptable solution. In conventional decision theory, preferences are defined via utility functions (also called cost functions or performance indices) that rank-order the possible choices in terms of their desirability. For single-agent decision problems, an obvious solution concept is to maximize the utility function. For multiagent decision problems in which the agents are exclusively self-interested, the game-theoretic Nash equilibria is the appropriate solution concept. Unfortunately, the Nash solution concept can lead to unduly pessimistic solutions for agents disposed to cooperate. For example, in the prisoner's dilemma, the single Nash equilibrium corresponds to the players betraying each other, but the payoffs are highest if both cooperate and remain loyal to each other. (Axelrod has addressed this problem in detail; see [8].)

Satisficing game theory [9] employs a new utility structure and a new solution concept, both of which easily accommodate cooperative agent communities, and are therefore, well matched to conflict resolution. Before describing the application of this new theory to ATC, we summarize the essential components of satisficing game theory.

A. Social Utility

Perhaps the most basic requirement for the formation of a coherent society from a collection of autonomous agents is some basic guarantee of equity: no agent should be required in all situations to subjugate its own interests to benefit the group. Thus, a reasonable condition for a society of cooperatively disposed agents is that it be *socially coherent*, meaning that categorical subjugation cannot occur. As established in [10] and [11], social coherence can be assured if and only if the preferences of a multiagent system are expressed by the mathematical syntax of multivariate probability theory. Preferences are represented using *social utilities*, each of which is a mass function $p_G : \mathbf{U} \rightarrow [0, 1]$ where $\mathbf{U}$ represents the set of possible actions. Social utilities must satisfy the following properties.

1) *Nonnegativity*: $p_G(\mathbf{u}) \geq 0 \; \forall \mathbf{u} \in \mathbf{U}$.

2) *Normalization*: $\sum_{\mathbf{u} \in \mathbf{U}} p_G(\mathbf{u}) = 1$.

Because social utilities are probability mass functions (but with altogether different semantics), they inherit the properties of conditioning, independence, and marginalization.

A unique feature of satisficing game theory is its use of *dual utilities*. Each agent X_i is assumed to be composed of two *selves*—a *selecting self* S_i and a *rejecting self* R_i. Associated with each S_i is a social utility p_{S_i} that orders each action available to X_i in terms of its effectiveness in avoiding failure, or conversely, in achieving success. Similarly, the social utility p_{R_i} associated with each R_i orders each action in terms of its inefficiency in conserving resources. Social utilities p_{S_i} and p_{R_i} are referred to as *selectability* and *rejectability functions*, respectively, or simply *selectability* and *rejectability*. In effect, these represent, respectively, the benefits and costs of choosing a particular action. The *individually satisficing set* is defined as

$$\Sigma_i = \{u_i \in U_i : \; p_{S_i}(u_i) \geq q_i p_{R_i}(u_i)\} \tag{1}$$

where U_i is X_i's (finite) set of possible actions and q_i is X_i's *negotiation index*. The satisficing set contains the collection of actions for which the degree of effectiveness is at least as great as the degree of inefficiency, modulated by the negotiation index. Simply put, the benefits of each action in the set outweigh its costs. Nominally $q_i = 1$, but its value may be reduced if desired in the process of negotiation. The Cartesian product of all individually satisficing sets, called the *satisficing rectangle*, is denoted as

$$\Re = \Sigma_q^1 \times \cdots \times \Sigma_q^n. \tag{2}$$

Each vector $(u_1, \ldots, u_n) \in \Re$ represents a collection of individual decisions that are each "good enough" for the individuals, in the sense that failure is avoided while conserving resources.

The next issue to consider is the behavior of the group as a whole. Let $G = \{X_1, X_2, \ldots, X_n\}$ be a group of n autonomous agents. Then, the *group action set* is $\mathbf{U} = U_1 \times \cdots \times U_n$, the Cartesian product set. For any k-element subgroup $G^i = X_{i_1}, \ldots, X_{i_k}$, its corresponding action set is $\mathbf{U}_i = \{U_{i_1} \times \cdots \times U_{i_k}\}$. As with the individuals, we associate a social utility, the *group selectability function* $p_{S_1,\ldots,S_n}$, to characterize group-level effectiveness, and a *group rejectability function* $p_{R_1,\ldots,R_n}$ to characterize group-level inefficiency. The *group-level satisficing set* is defined as

$$\begin{aligned}\Sigma_G = \{&(u_1, \ldots, u_n) \in \mathbf{U} : \\ &p_{S_1 \cdots S_n}(u_1, \ldots, u_n) \geq q_G p_{R_1 \cdots R_n}(u_1, \ldots, u_n)\}\end{aligned} \tag{3}$$

where q_G is the *group negotiation index*.

III. Satisficing Conflict Resolution

In a decentralized approach, collisions must be avoided by the joint actions of individual aircraft, each using only local knowledge. We assume that each aircraft is equipped with a transponder that broadcasts information about its location and intentions to all other aircraft within a 50-nmi radius. This information includes current position and heading, destination, flight time, and delay (relative to an unobstructed straight line flight). Using this information, each aircraft will choose an action at each time step that considers both the positions and preferences of other aircraft with which it could conflict. To maintain a high level of accuracy, we use a time step of 1 s in our simulator. If aircraft are unable to exchange information this frequently, estimated positions obtained from predictive models may be used.

For this study, we assume that all aircraft fly at the same altitude and at the same constant speed. Once each second, each aircraft chooses one of five directional options, including flying straight, moderate turns (2.5°) left or right, and sharper turns (5°)) left or right. In our model, heading changes are instantaneous. Considering that the standard turn rate of conventional aircraft is about 3°/s, the moderate heading change option reflects a gentle turn, while the sharp turn corresponds to a more aggressive evasive action.

The first step in applying satisficing theory is to determine, in general terms, how selectability and rejectability are to be defined for the problem of interest. (Multiple formulations are typically possible within the satisficing framework.) Since selectability reflects goal achievement, and since the goal of each agent is to get to its destination, we base selectability on how directly the resulting heading leads to the destination. We use rejectability to reflect safety concerns; the number and immediacy of conflicts resulting from a given directional choice will determine the value of the rejectability function for that option. The precise details of these social utility functions are explained later.

The next step is to determine what information from other aircraft will be used in the local decision process for each agent. Relationships between agents in the system are represented by directed acyclic graphs depicting *influence flow*. Because the set of potentially conflicting aircraft is time varying, static influence flows cannot accurately represent the system. In our approach, situation-specific influence flows are created that cause each agent to consider the preferred headings of aircraft with higher rankings. Recomputed at each time step, the ranking is determined by delay, flight time, and proximity to destination. Each aircraft first partitions the set of *viewable* aircraft (those within 50 nmi) into two subsets: those within 5 nmi of their destination and all others. Aircraft in the first set have higher rank than those in the second set. Within each set, aircraft are ranked according to delay, with greater delay bringing a higher rank. Finally, aircraft in the same set with the same delay are ranked according to their current time in flight, with longer flight times resulting in a higher ranking. This paper assumes that the ranking mechanism results in a unique priority for each aircraft, and that rank orderings of conflicting aircraft are consistent from the point of view of all participants.

To determine the selectability of its options at every time step, each agent X_i creates an influence flow graph, as shown in Fig. 1, summarizing direct and indirect influences on S_i, X_i's selecting self. The graph includes an edge to S_i from every *parent* of X_i—viewable aircraft with higher ranking than X_i that could conflict with X_i for some allowed heading choice. The selecting selves of other aircraft are used because they reflect heading preferences. The graph also includes all parents of X_i's parents that lie within its viewable region, and S_i is the only leaf node in the graph. The set of X_i's parents constitute its *priority set*, denoted as P_i. Members of P_i are indexed 1 to $|P_i|$ and only members of this set directly influence p_{S_i}. In the scenario depicted in Fig. 1, three aircraft with higher rankings are within X_i's viewable area, two of which are in conflict with X_i, and hence, in its priority set with indices 1 and 2. A third aircraft, X_j, has conflicts with both members of X_i's priority set but not with X_i directly.

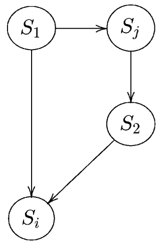


Fig. 1. Example influence flow graph to Determine selectability.

In our formulation, the rejectability function p_{R_i} reflects concerns for the safety of X_i. Each aircraft compares a linear extension of each of its directional options $u_l^i, l = 1, \ldots, |U|$ with linear projections of current headings of all aircraft in P_i. Each projected conflict adds a weight to that option, depending on its distance in time and the severity of the conflict: collisions are weighted more heavily than near misses. After all higher ranking aircrafs have been considered, the weight of each option is normalized over the option space so that a mass function is obtained.

A. Rejectability

The details of computing the rejectability social utility are as follows. Let R_c and R_{nm} denote the *collision radius* and *near miss radius*, respectively, with $R_c << R_{nm}$. Let u_c represent the current heading, $d(i,k)$ the projected distance from X_i's current position to the point of closest approach to $X_k \in P_i$, and $d_{\min}(i,k)$ the shortest distance between X_i and X_k on their projected paths. Then

$$p_{R_i}(u_l) \propto \sum_{X_k \in P_i} W_R(X_k(u_c),\, X_i(u_l)) \tag{4}$$

where $\propto$ signifies proportionality. (The raw weights are ultimately normalized to create a mass function, but proportionality is preserved.) The weighting function W_R is defined by

$$W_R(X_k(u_c), X_i(u_l)) = \begin{cases} 2\alpha, & \text{if} \quad d_{\min}(i,k) \le R_c \\ \alpha, & \text{if} \quad R_c < d_{\min}(i,k) \le R_{nm} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where α is defined as

$$\alpha = \begin{cases} \left(\left(1 + \dfrac{R_{nm} - d_{\min}(i,k)}{R_{nm}}\right)\left(\dfrac{1}{d(i,k)}\right)\right)^{\beta}, & \text{if } d(i,k) \le 3R_{nm} \\ \left(\dfrac{1}{d(i,k)}\right)^{\beta}, & \text{otherwise}. \end{cases} \tag{6}$$

The parameter β is a variable that was experimentally tuned; simulation results reported in Section VII were obtained using $\beta = 2/3$.

The aforementioned construction increases the weight (and thus, the rejectability) of heading options that lead to conflicts or small separation from other aircraft. If conflicts exist in every direction, the least rejectable option will be the one with the most distant conflicts. If no conflicts exist for all heading choices, p_{R_i} is set to a uniform distribution, so subsequent agent decisions depend only on selectability.

B. Selectability

In contrast with rejectability, selectability is influenced by the preferences of other agents. The first step in computing selectability is assigning a rank $r(u_l)$ to each option according to $|u_{\text{dir}} - u_l|$, where u_{dir} is the direct heading to the aircraft's destination. Ranks for the five options are assigned such that $r(u_l)$ is 1 for the heading option closest to u_{dir} and 5 for the option furthest from u_{dir}. A weight $w_S(u_l)$ is then assigned as a function of $r(u_l)$ and the magnitude of $|u_{\text{dir}} - u_l|$

$$w_S(u_l) = \begin{cases} 3, & r(u_l) = 1 \\ 2, & r(u_l) = 2 \\ 2, & r(u_l) = 3, 2.5^{\text{o}} < |u_{\text{dir}} - u_l| \leq 5^{\text{o}} \\ 1.1, & r(u_l) = 3, 5^{\text{o}} < |u_{\text{dir}} - u_l| \\ 1.1, & r(u_l) = 4, |u_{\text{dir}} - u_l| \leq 5^{\text{o}} \\ 1, & r(u_l) = 4, 5^{\text{o}} < |u_{\text{dir}} - u_l| \\ 1.1, & r(u_l) = 5, |u_{\text{dir}} - u_l| \leq 5^{\text{o}} \\ 1, & r(u_l) = 5, 5^{\text{o}} < |u_{\text{dir}} - u_l|. \end{cases} \tag{7}$$

Assigned weights are then normalized over X_i's option space to form mass function $\sigma_{S_i}(u_l^i)$.

Despite the apparent complexity of the previous definition, only five distinct mass functions can be constructed by this procedure. When no other agents influence X_i, either because it has the highest ranking or because no viewable aircraft conflict with it, the selectability $p_{S_i}(u^i) = \sigma_{S_i}(u^i)$. Otherwise $p_{S_i}(u^i)$ is formed as the convex combination of $\sigma_{S_i}(u^i)$ and another mass function $\rho_{S_i}(u^i)$, which accounts for the influence of other agents. $\rho_{S_i}(u^i)$ is created as follows.

For each of its parents $X_k \in P_i$, X_i calculates a matrix of weights

$$W_{ik}(u_l^i, u_j^k) = W_S(X_i(u_l^i), X_k(u_j^k)), \qquad k = 1, \ldots, |P_i| \tag{8}$$

where

$$W_S(X_i(u_l^i), X_k(u_j^k)) = \begin{cases} 1, & \text{if } d_{\min}(i,k) > R_{nm} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Thus, all pairs of heading options that cannot conflict are assigned a weight of one. The columns of the matrix of weights are then normalized such that

$$\sum_{u^i \in U} W_{ik}(u^i, u_j^k) = 1. \tag{10}$$

Let $\bar{u}^m,\ m = 1, \ldots, |U|^{|P_i|}$ be a vector of dimensionality $|P_i|$ representing a particular choice of heading for each of the aircraft in P_i, and let $[\bar{u}^m]_k$ be the kth element of that vector. Then

$$\rho_{S_i|S_1,\ldots,S_{|P_i|}}(u_l^i|\bar{u}^m) \propto \sum_{k=1}^{|P_i|} W_{ik}(u_l^i, [\bar{u}^m]_k), \qquad m = 1, \ldots, |U|^{|P_i|}. \tag{11}$$

The marginal selectability is computed by summing over all possible vectors $\bar{u}^m$ according to

$$\begin{aligned} \rho_{S_i}(u_l^i) &= \sum_{m=1}^{|U|^{|P_i|}} \rho_{S_i|S_1,\ldots,S_{|P_i|}}(u_l^i|\bar{u}^m) \cdot \hat{p}_{S_1}([u^m]_1) \\ &\quad \cdot \hat{p}_{S_2}([u^m]_2) \cdots \hat{p}_{S_{|P_i|}}([u^m]_{|P_i|}) \qquad (12) \\ &= \sum_{u^1 \in U} \sum_{u^2 \in U} \cdots \sum_{u^{|P_i|} \in U} \rho_{S_i|S_1,\ldots,S_{|P_i|}}(u_l^i|u^1, \ldots, u^{|P_i|}) \\ &\quad \cdot \hat{p}_{S_1}(u^1) \cdot \hat{p}_{S_2}(u^2) \cdots \hat{p}_{S_{|P_i|}}(u^{|P_i|}). \qquad (13) \end{aligned}$$

The computation of the marginal $\rho_{S_i}(u_l^i)$ uses the product of estimated marginals $\Pi_{k=1}^{|P_i|} \hat{p}_{S_k}(u^k)$ instead of the true joint selectability $p_{S_1, S_2, \ldots, S_{|P_i|}}(u^1, \ldots, u^{|P_i|})$ because agents are limited to local information, and neither the true joint selectability nor the true marginal selectabilities of other agents are generally available. The estimated marginal $\hat{p}_{S_k}(u^k)$ for all $X_k \in P_i$ is calculated by considering the effects on X_k of all aircraft in P_k that are within X_i's field of view. The accuracy of X_i's model on p_{S_k} is dependent on how many and which of X_k's parents are viewable by X_i. Of course, X_i has no way of knowing to what degree its viewable list and P_k overlap but, as a general rule, the lower the distance from X_k to X_i, the greater the likelihood of members of P_k being in X_i's viewable set. Thus, X_i's estimates are most accurate for the aircraft nearest—and therefore, of most immediate concern—to itself. Note that $\rho_{S_k}(u^k)$ will be uniform for agent $X_k \in P_i$ whenever the aircraft that influence X_k, if any, are not viewable by X_i.

Finally, the selectability mass function $p_{S_i}(u_l^i)$ is formed by the convex combination

$$p_{S_i}(u_l^i) = \lambda \sigma_{S_i}(u_l^i) + (1 - \lambda)\rho_{S_i}(u_l^i), \qquad \lambda \in [0, 1]. \tag{14}$$

All aircraft for which $P_i = \emptyset$ have $\lambda = 1$. Otherwise, parameter λ affects the relative weight given to the perceived heading preferences of other aircraft. Our simulation results used a very small λ (0.001) relegating σ_{S_i} to the role of a tiebreaker between otherwise equivalent options.

C. Satisficing Set

Once aircraft X_i has determined its selectability and rejectability, it can identify the set of satisficing options. In general, if the satisficing set is not a singleton, a variety of tie-breaking rules can be employed. For example, agents willing to tolerate risk for high gains could maximize selectability. Risk averse agents could choose to minimize rejectability, but this gives no guarantee of progress toward the goal. For this application, the satisficing option is selected with the largest difference between selectability and rejectability values, insuring the greatest possible progress toward the goal relative to the risk incurred.

In general, the effectiveness of a satisficing solution depends on selectability and rejectability utilities reflecting different aspects of the problem, yet both utilities defined earlier consider projected conflicts. An explanation is in order. For this application, selectability reflects goal achievement, or getting to the destination as directly as possible. Selectability considers the preferred direction of other aircraft so that the most effective heading will be chosen. (The concern here is that steering directly into the preferred path of another aircraft will require additional avoidance maneuvers in the future.) In contrast, rejectability reflects safety concerns, or maintaining adequate separation from other aircraft. In the case of rejectability, we test for possible collisions with linear projections of the current *actual* headings of other aircraft, rather than with estimates of preferred headings at the next time step. The distinction is subtle, but important.

D. Simplified Model

The computation required for selectability scales as $|U|^{|P_i|}$. For current and projected aircraft densities, the priority sets are small and the computational overhead is manageable. We have run real-time simulations with the full computations of over 70 aircraft on a single processor. However, for some of the stylized scenarios considered in Section VII, aircraft densities increase significantly (to unrealistic levels) and the computational demand is too great for real-time execution. For this reason, we explored an alternative approach with reduced computational requirements that would permit real-time operation in densely congested airspace.

As described earlier, the construction of the weighting matrices is intuitive and heuristic. Given those matrices, the construction of $\rho_{S_i|S_1,\ldots,S_{|P_i|}}(u_l^i|\bar{u}^m)$ involves a good deal of averaging, and the final computation of the marginal $\rho_{S_i}(u_l^i)$ is necessarily approximate. These facts suggest that it should be possible to simplify the model and reduce computation without incurring significant losses in performance. The simplified model we developed, which scales in complexity with $|U|^{|U|}$, exploits the fact that there are only five unique $\sigma_i(u_l^i)$ mass functions. The simpler model permits real-time operation in all modeled scenarios, and as will be shown, it performs nearly as well as the full model.

The principal simplification comes from partitioning the priority set P_i into $|U| = 5$ sets $S_1^g, \ldots, S_5^g$ according to each aircraft's preferred heading option, as determined by the mass function σ_{S_i}. That is, all aircraft in group S_l^g have u_l as their most selectable option, based solely on the fact that u_l leads most directly to the destination. Let $W_g(j)$ denote the cardinality of S_j^g. As with the full model, we create a matrix of weights, but this time there is only one matrix for each partition $S_l^g \in P_i$ rather than one for each aircraft $X_k \in P_i$. The matrix of weights for the kth partition is given by

$$W_{ik}(u_l^i, u_m^k) = \sum_{X_j \in S_k^g} W_S(X_i(u_l^i), X_j(u_m^k)). \tag{15}$$

The normalization procedure is the same as that described earlier, namely

$$\sum_{u^i \in U} W_{ik}(u^i, u_m^k) = 1. \tag{16}$$

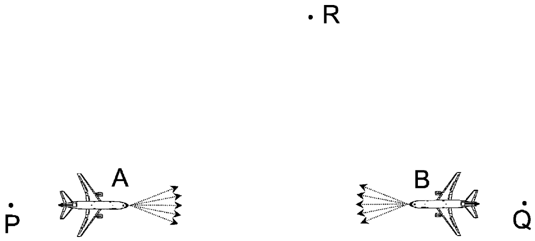


Fig. 2. Example conflict.

The simplified conditional selectability is calculated as

$$\rho_{S_i|S_1^g,\ldots,S_5^g}(u_l^i|u^1, u^2, \ldots, u^5) \propto \sum_{k=1}^{5} W_g(j) W_{ik}(u_l^i, u^k). \tag{17}$$

The conditional $\rho_{S_i|S_1^g,\ldots,S_5^g}$ can be interpreted as the selectability of X_i conditioned on the selectability of each heading option aggregated over each partition S_j^g. In effect, each partition S_l^g is treated as an individual aircraft, and as there are only $|U|$ heading options per aircraft, there are only $|U|^{|U|}$ values of the conditioning option vector $u^1, \ldots, u^5$. At this step, the simplified model reduces the memory required more than it reduces the computational overhead.

The marginal selectability is now calculated as

$$\rho_{S_i}(u_l) = \sum_{u^1 \in U} \sum_{u^2 \in U} \cdots \sum_{u^5 \in U} \rho_{S_i|S_1^g,\ldots,S_5^g}(u_l^i|u^1, \ldots, u^5) \cdot \sigma_{S_1^g}(u^1) \cdot \sigma_{S_2^g}(u^2) \ldots \sigma_{S_5^g}(u^5). \tag{18}$$

The computational savings relative to the full model come at this step, and they arise from two sources. First, there is no longer any need to compute the $|P_i|$ marginals $\hat{p}$, and second, the marginalization process is much easier because there are only $|U|$ conditioning agents. As before, the selectability function $p_{S_i}(u_l^i)$ is formed by a convex combination of $\sigma_{S_i}(u_l^i)$ and $\rho_{S_i}(u_l^i)$.

IV. Example Scenario

To illustrate satisficing-based conflict resolution, consider the two-aircraft scenario depicted in Fig. 2. The aircraft A and B are headed directly to their destinations, points Q and P, respectively, and a collision will occur if both continue on their current heading. (Neither the aircraft nor the five heading-change options are drawn to scale in the figure.) Assume that A and B are 10 and 5 min behind schedule, respectively.

The selectability and rejectability of A are straightforward to compute. Because its value is reduced for options that take the aircraft off course, p_{S_A} is the highest for the option of flying straight, somewhat lower for moderate turns either direction, and the lowest for sharp turns in either direction. Since A has a higher ranking because of its higher delay, it will not consider B in computing its selectability. Similarly, because A has no conflicts with higher priority aircraft, it determines that p_{R_A} is a uniform distribution over its option space. Because it is the option for which $p_{S_A} - p_{R_A}$ is largest, A will choose to fly straight.

B has lower priority, so it will sacrifice some efficiency to resolve the conflict, and it will do so in a way that takes it the least off course. The value of B's selectability p_{S_B} is largest for the option that leads to a conflict-free path—assuming A flies in its preferred direction—and that deviates least from B's direct path to its destination. If the distance between A and B is sufficient to avoid a conflict with a moderate turn, then the two moderate turns will have the highest selectability. The two sharp turns will have slightly lower values, while the straight option will be assigned a near-zero value.

B's rejectability p_{R_B} is determined by comparing directional options with A's actual heading. If moderate turns avoid conflicts, going straight will be assigned the value 1 and all other options will have value 0. If moderate turns result in a near miss and only sharp turns avoid conflicts, going straight will be assigned the highest rejectability, slight turns will have smaller values, and sharp turns will have a rejectability of 0. Ties are broken by picking the option that takes the aircraft closest to its destination. The option for which $p_{S_B} - p_{R_B}$ is greatest will, therefore, correspond to the smallest detour that B can take that avoids a conflict with A.

Suppose now that A's destination is actually R, and that it is not headed directly to its destination because it is completing an avoidance maneuver. In this case, p_{S_A} is highest for a sharp left turn, and p_{R_A} is uniform, so A will choose the sharp left turn. In determining p_{S_B}, A's preference to turn left (taking the shortest path to its destination) will be taken into consideration, and B's options to go straight or turn left (away from A's anticipated path) will be assigned higher values. Values of p_{R_B} are based on A's actual heading, so they are identical to the previous scenario. Thus, B will choose a left turn, but within a small number of (1 s) time steps, A will have changed its heading enough that B will turn directly toward its goal, since that heading will no longer lead to a projected conflict with A.

Finally, important concerns about safety can be addressed in the context of this scenario. If both A and B produce consistent rankings and are operating normally, exactly one of them will turn and the conflict will be resolved. (This avoids both the case where neither turns, and the case where both turn, making mirror-opposite moves in sequence and prolonging the conflict.) If A and B were to obtain inconsistent rankings, an eventuality that could be prevented in a well designed system, the lower ranked aircraft might not defer to the other aircraft as expected. Similar behavior would be observed if B were an instance of a *noncompliant* aircraft that transmits the expected information regarding its location and intentions but never defers to other aircraft, simply flying a direct path to its destination. While no conflict resolution scheme can be effective if the participants fail to follow a fundamental set of expectations, our satisficing approach offers a simple and natural way of dealing with the failure of B to defer: other aircraft detect the unexpected behavior and move B to the top of their rankings. In Section VII, we consider the impact on performance when 10% of the aircraft in the system are noncompliant.

V. Alternative Conflict Resolution Techniques

Widespread interest in ATC enhancements has resulted in the development and analysis of a variety of conflict-resolution approaches. Proposed schemes differ in several important ways, including centralized or distributed control, the actions allowed to avoid conflicts, and the feasibility of completing the required computation in a real-time setting.

Krozel *et al.* describe three different conflict resolution algorithms, one centralized and two distributed [12], all of which are implemented as constant-speed heading change maneuvers. The centralized approach determines the set of conflicts arising in the next 8 min if no corrective actions were taken. Aircraft are partitioned into clusters such that each pair of aircraft that conflict are in the same cluster. All aircraft within a cluster are ranked using a permutation sequence, and the highest ranking aircraft is allowed to fly its nominal trajectory. A conflict-free trajectory is then sought for each remaining aircraft in sequence. If, at any point, an acceptable conflict-free path cannot be found, the algorithm restarts with a different ranking and permutation sequence.

In Krozel's decentralized approaches, aircraft resolve their own conflicts as they are detected. Multiple conflicts within the 8-min look-ahead window are resolved in a sequential pair-wise fashion, either passing in front of or behind the conflicting aircraft. A *myopic* strategy selects the alternative that requires the smallest heading change. A second *look-ahead* strategy further examines the selected maneuver to ensure that it does not produce a conflict that would occur earlier than the original conflict. If such a conflict is detected, the strategy tries the alternative maneuver, and then small heading offsets from the original choice, if needed.

Pappas *et al.* propose a decentralized conflict architecture that views the aircraft as a hybrid system incorporating both discrete events and individual dynamics modeled by differential equations [13]. Projected conflicts are resolved in two phases. First, noncooperative methods from game theory are used by each aircraft to search for a velocity change that guarantees separation regardless of the actions of the opponent. If the first phase is unsuccessful, the aircraft employs coordinated constant-velocity heading-change maneuvers to avoid the conflict. Maneuvers are described for up to three aircraft depending on the geometry of the scenario. The noncooperative game-theoretic approach is expanded in [14] to include both path deviations and speed variations. Subsequent extensions have included the following: a complete methodology for generating provably safe conflict heading-change and velocity-change resolution maneuvers for two aircraft [15], [16], a comparison of the hybrid approach relative to a continuous kinematic planner proven to be safe with up to three aircraft [17], and a protocol for resolving conflicts with instantaneous heading-change maneuvers when conflicting aircraft are out of direct communication range [18].

Kosecka *et al.* use distributed motion planning algorithms based on potential and vortex fields to generate prototype heading-change maneuvers for multiaircraft conflicts; actual maneuvers are flyable, piece-wise linear approximations of the prototypes which can be proven safe using hybrid verification techniques [19]. Selected maneuvers are shown for up to four aircraft. This work was extended in [20] to include altitude change maneuvers if heading changes in the horizontal plane were unable to resolve the conflict.

Dugail *et al.* analyze a decentralized conflict resolution scheme for two perpendicular flows of air traffic that intersect at a fixed point [21]. Upon entering the airspace, each aircraft makes a single instantaneous heading change—the minimum required to avoid conflicts with those aircraft already present. After the maneuver, each aircraft flies in a straight line until leaving the modeled airspace. The authors prove that this conflict resolution scheme does not result in arbitrarily large avoidance maneuvers and is therefore stable. In related work [22], scenarios are examined with traffic flows that meet at arbitrary angles. Avoidance maneuvers include both instantaneous heading changes and instantaneous lateral position changes.

Resmerita and coworkers describe an approach that partitions the airspace into static cells that may be occupied by only one aircraft at a time, thus ensuring separation [23], [24]. Conflict resolution equates to finding a conflict-free path through a resource graph representing the cells in the airspace. The aircraft share a common database that includes preferred flight plans for all aircraft in the system. When a new aircraft desires to enter the system, it registers its flight plans in the database and compares its paths with those of active aircraft. If none of its preferred paths are conflict free, resources are requested from other aircraft, which are required to relinquish resources if an alternative path to their destination exists. If resource requests do not produce a solution, the new aircraft is not allowed to enter the airspace.

Bicchi and Pallottino propose a method for planning optimal conflict resolution maneuvers for kinematic models of aircraft flying in a horizontal plane with constant velocity and curvature bounds [25]. The approach is formulated as an optimal control problem to minimize total flight time: necessary conditions are derived, possible trajectories are parameterized, and solutions are numerically computed. In this approach, the number of optimization problems grows combinatorially with the number of aircraft involved. Both centralized and decentralized implementations are described and simulated. Similar approaches were later applied to systems with centralized control and aircraft maneuvers consisting of either instantaneous velocity changes or single instantaneous heading changes [26], to a decentralized hybrid approach with instantaneous heading changes including up to three aircraft [27], and to a decentralized hybrid system for an arbitrary number of nonholonomic vehicles [28].

Other authors have studied conflicts using probabilistic models that allow for uncertainty in aircraft position due to wind and errors in tracking, navigation, and control. Paielli and Erzberger describe a means for estimating the probability of a conflict between two aircraft, given predicted trajectories for each [29]. Trajectory prediction errors are modeled with a normal distribution, error covariances for an aircraft pair are combined into a single covariance of relative position, and a coordinate transformation is used that allows an analytical solution. Prandini *et al.* introduce two probabilistic prediction models, one for mid-range (tens of minutes to conflict) and one for short-range (seconds or minutes to conflict) [30]. When a probable conflict is detected, a decentralized conflict resolution algorithm is employed to make heading changes based on potential fields in which aircraft repel each other. Simulation results are included for up to eight aircraft.

Rong *et al.* describe a cooperative agent-based solution to conflict resolution based on constraint satisfaction problems [31]. Using direct communication, conflicting agents negotiate pair-wise until a mutually acceptable resolution is found. Agents take turns proposing solutions; if the other aircraft rejects the proposal, it sends a revised solution accompanied by information about whatever private constraint the previous solution violated. If negotiation fails to produce an acceptable alternative, the aircraft turn to centralized controllers for a resolution.

In an approach based on computational geometry, Chiang *et al.* employ a Delaunay diagram to represent the aircraft in flight [32]. Since nearest neighbor information is encoded in the diagram, a conflict alert is triggered if the length of an edge falls below a separation threshold. The conflict resolution algorithm is computationally intensive, amounting to the construction of a nonintersecting set of piecewise linear tubes or pipes through space–time, each of which corresponds to the trajectory of an aircraft.

Finally, Kuchar and Yang describe a framework in which 68 previously published methods for conflict detection and resolution are categorized [33]. Critical factors in their taxonomy included conflict resolution methods (prescribed, optimized, force-field, or manual), maneuvering options (speed change, lateral, vertical, or combined), and the management of multiple aircraft conflicts (pairwise or global).

VI. Performance Measures

In order to compare and evaluate alternative approaches to conflict resolution, appropriate metrics must be used to ensure that both safety and performance objectives are met. A variety of measures have been employed in previously published work. In this section, we discuss the most promising of these and motivate the metrics used in Section VII.

A. Separation Assurance

For any algorithm, the most important metric is that of safety, or spatial separation of aircraft. The frequency of conflicts is a function of traffic density and the physical geometry of the intersecting flight paths. Surprisingly, many papers describing algorithms for ATC and conflict resolution do not explicitly report the number of near misses or collisions that occurred in their simulation runs. In our studies, we track and report two distinct types of separation violations: *collisions*, when aircraft come within 300 ft of each other, and *near misses*, which occur when aircraft are separated by less than 5 nmi.

B. System Efficiency

System efficiency measures the degree to which the aircraft in the system are able to follow direct, linear flight paths to their destinations [12]. Conflict resolution maneuvers typically cause aircraft to deviate from ideal paths and to consume more resources. Good solutions to conflict resolution should meet safety criteria while maintaining high levels of efficiency.

Because all aircraft are identical and cruise at the same speed, and since conflict resolution maneuvers in our model are constant-speed changes in heading, system efficiency can be determined in our simulations from the time required by each aircraft to reach its destination. We define the individual efficiency for aircraft i as

$$E_i = \left(\frac{t_i}{t_{d_i} + t_i} \right) \tag{19}$$

where t_i is the ideal flight time of the aircraft and t_d is the added delay time it experienced. Then, the system efficiency is given by

$$E = \frac{1}{N} \sum_{i=1}^{N} E_i \tag{20}$$

where N is the number of aircraft in the system. In an ideal scenario, all aircraft fly their nominal direct-line paths, so $E = 1$. As traffic density and congestion increase, aircraft deviate further from their ideal paths, and E decreases in value.

C. Standard Deviation of Efficiency

The standard deviation of individual efficiencies (D_E) of the aircraft in a system is proposed as a measure of fairness. If the standard deviation is large, then, in order to avoid safety violations, some aircraft have paths that are inefficient relative to those of other aircraft. D_E is given by

$$D_E = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (E_i - E)^2}. \tag{21}$$

In the ideal system (where $E = 1$), the added delay of each aircraft is zero, so $D_E = 0$. As congestion and added delay increase, D_E increases unless the fractional increases in path length experienced by all aircraft are identical. While D_E gives some insight into the operation of the system for a particular conflict resolution algorithm, it is of secondary importance relative to E. In an inefficient system, it is of little consolation that delays are spread fairly across all participants.

D. System Stability

System stability (S) is a measure of the extent to which conflict resolution maneuvers create new conflicts that, in turn, will require additional resolution maneuvers [12]. Let A_1 represent the set of *conflict alerts* arising if all aircraft were to fly their nominal straight-line paths. A conflict alert is a projected separation violation between two aircraft if both were to continue on their present heading. A conflict alert can occur only if the aircraft involved are within each other's viewable range, and each projected incident counts as a single alert. If A_2 is the set of conflict alerts arising when conflict resolution maneuvers are employed, then S is given by

$$S = \frac{|A_1|}{|A_2|}. \tag{22}$$

For example, if S has the value 0.5, then the conflict resolution algorithm caused a doubling in the number of conflicts that had to be considered relative to flying the nominal straight-line path.

VII. Results

Our simulation environment is similar to that used in other studies [12], [21], [22]. All aircraft are constrained to fly at the same altitude and at a constant speed of 500 mph. Small, instantaneous heading changes are the sole means of resolving conflicts. At each 1 s time step, each aircraft chooses one of its five directional options based on information it has received from aircraft within 50 nmi. After an option has been selected, all aircraft update their headings and positions, the new information is distributed to all adjacent aircraft, and the display screen is updated.

While certain patterns of conflicting aircraft might be common occurrences in any ATC system, it is impossible to enumerate all possible interaction geometries. For this reason, conflict resolution algorithms should be evaluated across a wide range of cases. Our studies include scenarios with both fixed geometries and random traffic patterns. We note that some simulated situations have high traffic densities that exceed the capabilities of any known solution technique, including our own. While they are highly unlikely to occur in actual air traffic, these scenarios were included in our study to allow comparison of our results with previous studies employing the same traffic patterns. Moreover, these cases give valuable insight into the capabilities and limitations of any conflict resolution scheme.

A. Random Flights

Based on a model used in [12], this scenario uses two concentric circles in open air space. Aircraft appear at random points on the outer circle (radius 120 nmi) and are assigned a random destination point on the inner circle (radius 100 nmi). The 20-nmi buffer between the circles decreases the probability of generating an aircraft initially in conflict with another aircraft. Because this scenario creates and tests a wide range of conflicts with varying geometries, it is a good test of conflict resolution algorithms.

Each simulation run has an associated traffic density, measured in aircraft per 10 000 nmi^2. Simulated densities range from 1 to 25 per 10 000 nmi^2. (For comparison, peak traffic densities in U.S. airspace are typically between 1 and 5 aircraft per 10 000 nmi^2 [12].) As each simulation run begins, new aircraft are generated at approximately 5-s intervals until the target density is achieved, after which new aircraft are generated only to replace those that arrive at their destinations. Statistics are collected only during a 50-min interval that begins once the target density is reached. Reported results are averaged over 20 simulation runs at each density. For comparison, we include results reported in [12] for their centralized scheme and their best performing decentralized algorithm—the look-ahead approach.

Fig. 3 confirms that the efficiency of conflict resolution algorithms generally declines as the traffic density increases. Among decentralized schemes, the satisficing approaches offer significantly better performance. Surprisingly, there is little difference in the system efficiency of the two satisficing schemes. Their performance relative to the centralized scheme is particularly noteworthy, given that a central controller can consider the entire airspace.

Even at the highest traffic densities, aircraft experience little added delay on average. In our simulations, the average flight length was approximately 163 nmi. At 500 mph, with a system efficiency of 97%, the average flight takes just 37 s longer than the ideal flight time. However, a change in system efficiency as

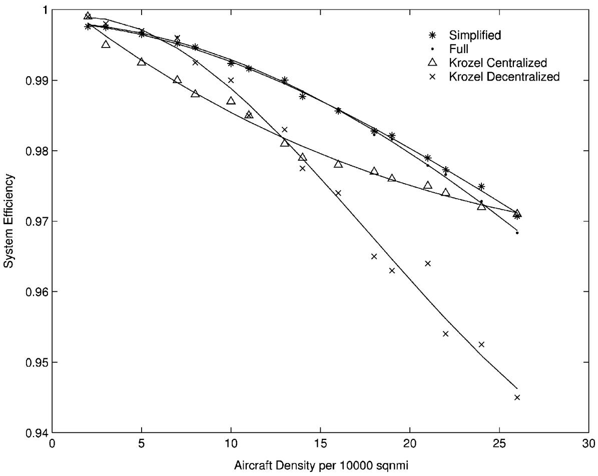


Fig. 3. Average system efficiency.

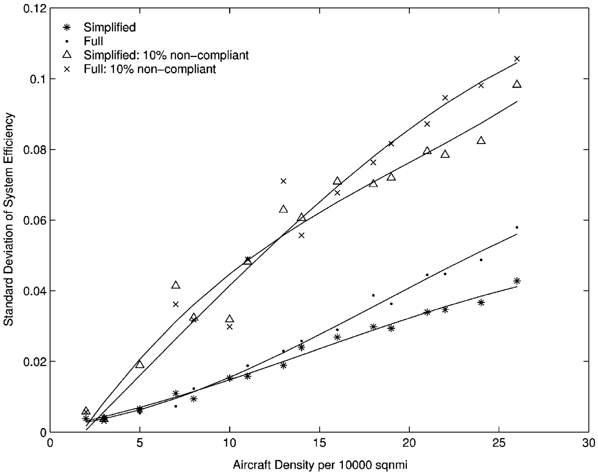


Fig. 4. Standard deviation of system efficiency.

small as 0.01 is considered significant, since it results directly in a comparable reduction in operating costs [12].

While the satisficing schemes exhibit impressive performance in terms of system efficiency, how evenly do they distribute the added delay as the density increases? The bottom two curves in Fig. 4 show that the standard deviation of system efficiency for our two algorithms grows smoothly as the traffic density increases. (No comparable numbers are available that describe other published schemes; the other two curves in the picture will be explained shortly.) Many aircraft fly near optimal flight paths, but as traffic density increases, more aircraft must participate in conflict resolution maneuvers that also increase in complexity.

Fig. 5 summarizes measurements of system stability. Because of its global perspective, the centralized approach is better than any decentralized scheme at avoiding resolution maneuvers that cause future conflict alerts. The system stability of the satisficing approaches is far better than the decentralized look-ahead algorithm. The satisficing algorithms are able to resolve multiple conflicts in a single maneuver, whereas the look-ahead scheme resolves conflicts pairwise and sequentially. Moreover, the satisficing approaches make effective use of the knowledge of the intended destinations of other aircraft, resulting in cooperative solutions that are effective in resolving conflicts.

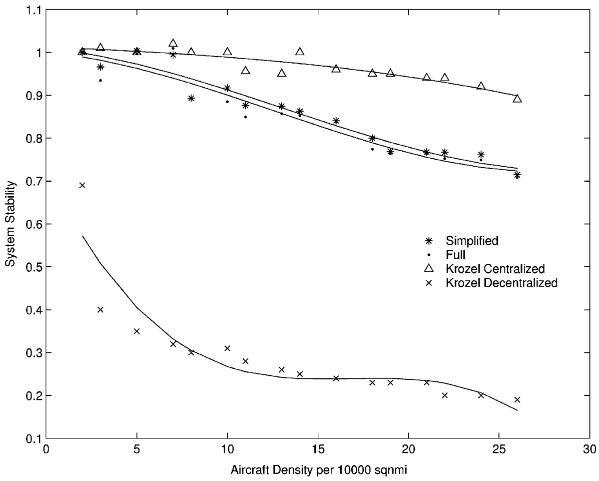


Fig. 5. Average system stability.

TABLE I
AVERAGE SAFETY RESULTS FOR RANDOM SCENARIO

	Full		Simplified		Full+NC		Simp+NC	
Density	C	NM	C	NM	C	NM	C	NM
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0
5	0.0	0.2	0.1	0.2	0.0	0.1	0.1	0.0
7	0.0	0.1	0.1	0.2	0.0	0.0	0.1	0.0
8	0.0	0.7	0.1	0.7	0.0	1.0	0.1	1.0
10	0.0	0.8	0.1	1.0	0.0	0.8	0.1	0.7
11	0.0	1.2	0.1	1.5	0.0	1.0	0.1	1.6
13	0.1	1.4	0.1	2.3	0.0	2.4	0.1	3.7
14	0.1	2.1	0.1	2.6	0.1	3.0	0.1	4.8
16	0.0	3.7	0.1	4.9	0.2	3.7	0.2	5.1
18	0.0	4.4	0.1	6.4	0.0	6.6	0.1	9.1
19	0.1	7.0	0.1	9.8	0.0	8.7	0.2	11.6
21	0.1	6.8	0.1	12.6	0.3	9.2	0.3	12.9
22	0.0	9.2	0.1	14.6	0.3	10.7	0.1	16.5
24	0.2	11.7	0.2	17.1	0.1	13.4	0.1	23.0
26	0.2	15.8	0.1	23.4	0.2	18.3	0.2	27.8

By the metrics represented in the figures, the simplified model outperforms the full model by a small margin, but this is not the full picture. As the traffic density increases, separation violations occur more frequently using the simplified model, as shown in Table I. The full model is more effective in avoiding separation violations, but this results in an increase in avoidance maneuvers and a corresponding decrease in system efficiency and system stability.

To further explore the capabilities of the satisficing algorithms, we investigated the effects of including aircraft that do not follow behavioral conventions. Consider the *noncompliant* aircraft described in Section IV that transmit the same information as other aircraft, but always fly directly to their destinations regardless of their rankings. As previously noted, satisficing algorithms offer a straightforward way of dealing with noncompliant aircraft: other aircraft move them to the top of their rankings after observing their failure to defer to others.

When noncompliant aircraft are included in the random flight scenario, the overall system efficiency is reduced. The noncompliant aircraft themselves have perfect individual efficiencies, but conforming aircraft are forced to make more detours, and conflicts are resolved without the full cooperation of all participants. The top two curves in Fig. 4 show the

TABLE II
RESULTS FOR THE CHOKE POINT SCENARIO

Aircraft	Near Misses	Collisions	E (%)
12	0	0	97.9
14	1	0	97.3
16	1	0	95.5
18	3	0	94.4
20	7	0	93.6
22	8	0	91.4
24	8	0	86.9
26	14	0	85.6
28	14	0	84.5
30	18	0	84.4
32	19	0	85.7

standard deviation of system efficiency for the two satisficing schemes when 10% of the aircraft are noncompliant. As traffic density increases, an increasing number of aircraft must make significant detours to avoid conflicts since the resolution of many conflicts is now unilateral and noncooperative. As expected, the system with noncompliant aircraft is markedly less fair than when all aircraft conform.

Table I summarizes the average number of collisions and near misses occurring during the 50-min simulation intervals as the aircraft density (per 10 000 nmi^2) increases. Results are shown for both full and simplified models, both with and without noncompliant aircraft. (Collisions and near misses were not reported in [12], so comparison with their approaches is not possible.) The results suggest that situations arise occasionally in the random flight scenario that are difficult to resolve. The frequency of separation violations increases as the traffic density increases, and it also increases when noncompliant aircraft are added to the system.

B. Choke Point

In this scenario, based on a model used in [26], all aircraft begin from evenly spaced points on a circle with radius 50 nmi. Each aircraft's destination is the point on the circle directly opposite its starting point, so all ideal paths coincide at the center of the circle. Although not representative of actual traffic patterns, this scenario presents a considerable challenge for any conflict resolution algorithm, both in terms of cooperatively resolving simultaneous multiway conflicts and in completing all required computation in real time. In the full satisficing model, influence flows must consider all aircraft, and the resulting computation cannot run in real time. We present results only for the simplified model.

Table II summarizes simulation results as the number of aircraft is varied. (Each result is from a single run; neither the scenario nor the algorithm include random aspects, so multiple runs with a given number of aircraft give identical results.) The number of near misses increases as the number of aircraft is increased, but there are no collisions. Because the circle is of fixed size, an increase in the number of aircraft causes a corresponding increase in traffic density. As the density increases, the satisficing algorithm exhibits a graceful degradation in efficiency.

The pattern of the solution that emerges is particularly noteworthy. Fig. 6 shows four interim snapshots for a run with 32 aircraft. Although the algorithm is in no way preprogrammed to handle this specific situation, the solution is essentially that of a multilayered roundabout, a solution technique suggested and evaluated by other researchers [13], [15], [16], [19], [26].

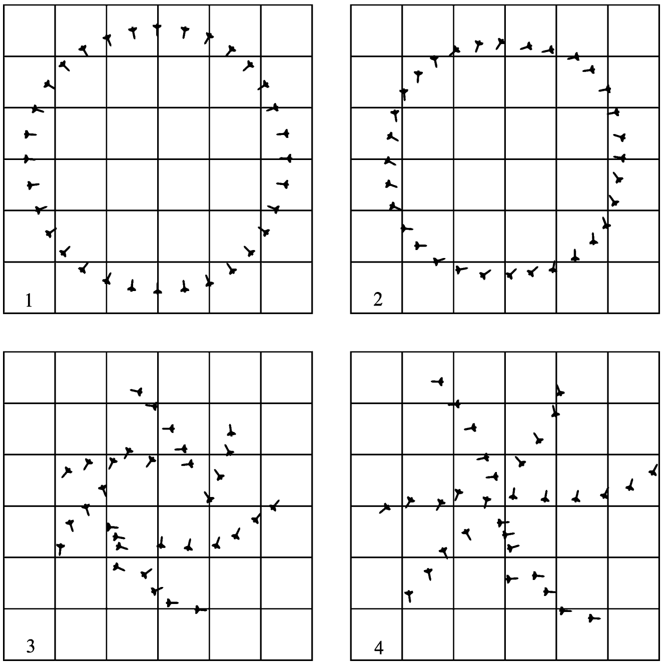


Fig. 6. Choke point snapshots with 10-nmi grid.

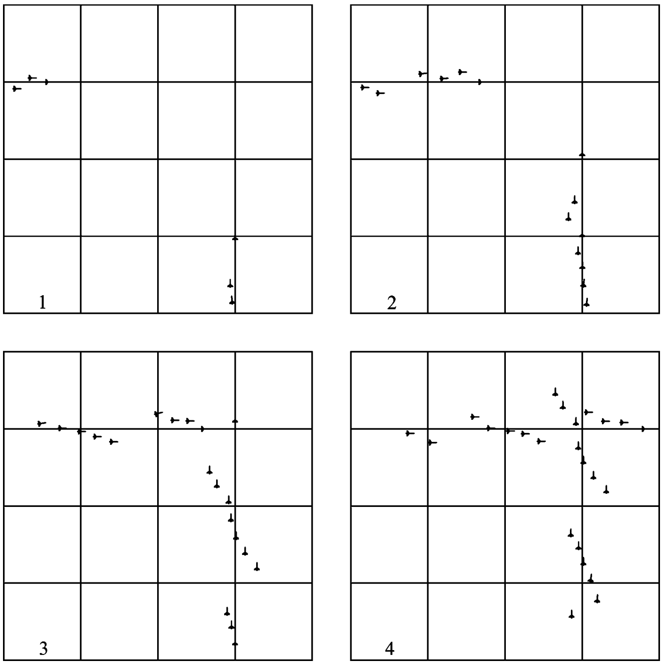


Fig. 7. Perpendicular flows with 25-nmi grid.

C. Perpendicular Flows

In this scenario, introduced in [21], two linear traffic flows intersect at right angles, one moving left to right and the second moving from top to bottom. Fig. 7 shows a sequence

TABLE III
RESULTS FOR PERPENDICULAR FLOW SCENARIO

Distribution	Flights	Near Misses	Collisions	E (%)
$\mu = 4$	2732	61	0.2	99.4
$\mu = 8$	3366	378	1.0	98.1
$\mu = 12$	3629	626	1.7	97.1
$\mu = 16$	3767	796	2.7	96.4
$\mu = 20$	3860	1011	2.4	95.4
$\mu = 24$	3915	1042	2.0	95.5
$\mu = 28$	3948	1132	2.3	95.1
$\mu = 32$	3989	1169	3.1	95.0
Constant Flow	4217	1847	3.0	92.3

of four screen snapshots from one simulation run. In a given flow, all aircraft are generated at the same point (slightly off-screen in the figure) with the same destination. As the flows approach the intersection point, the aircraft position themselves to avoid violations of the 5-nmi separation distance. For this scenario, aircraft are aware of all other aircraft within 100 nmi. At the default rate, aircraft are generated 40 s apart, creating a separation from the previous aircraft of about 5.5 nmi. Each aircraft can thus perform avoidance maneuvers without violating its safety margin with the aircraft immediately behind it.

Table III reports simulation results of the simplified model for the perpendicular flows scenario. Occasional gaps are inserted that are approximately the size of the safety zone around a single aircraft. The distribution column in the table describes the average number of aircraft in a consecutive *string* before a gap is introduced. The string length is a uniform random variable with distribution $U(\mu/2, 3\mu/2)$ for the values of μ shown. (The row marked *constant flow* includes no gaps.) Gap sizes in the vertical and horizontal flows are 75 and 80 s, respectively. The difference shifts the relative alignment of aircraft in the flows over time, creating different intersection geometries. The table reports the total number of flights completed, as well as the number of safety violations of each type. The results are averaged over ten different simulation runs, each modeling a 24-h period. The results show that safety and efficiency metrics show steady improvement as the frequency of inserted gaps increases.

Although the satisficing algorithm is not preprogrammed to handle this specific scenario, the solution that emerges (as depicted in Fig. 7) exhibits the same wave-like patterns as the solution presented in [21].

D. Computational Load

An important concern in any real-time implementation is computational load. Our custom simulator, written entirely in Java, employs a central thread to model the shared airspace with separate threads for the decision support system or agent onboard each aircraft. Communication between the simulator and each agent thread takes place via explicit message objects. In our simulations, we used a single Pentium IV processor to control all aircraft in all simulations, with up to 80 aircraft modeled simultaneously.

Of the two satisficing models, the full model is the more computationally demanding. For an n-agent system where each agent has k options, the time required for the computation of the full model is at worst $O(nk^n)$. The number of computations required depends on the number of conflicting aircraft within 50 nmi. For a large number of conflicting aircraft (as in the choke point scenario), this becomes difficult to complete in real time. This was our main motivation for developing the simplified algorithm that runs in constant time for each aircraft regardless of the number of conflicting aircraft. Of course, in an actual implementation, the decision code for each individual aircraft would run on its own dedicated computing resources, rather than sharing resources with all aircraft in the system as occurs in our simulator.

VIII. CONCLUSION

The need for new algorithms that automate decision making will continue to grow as air traffic densities increase. Satisficing decision theory offers an attractive method of modeling and solving distributed multiagent problems that are inherently cooperative as in the case of ATC. Satisficing theory is mathematically sound, robust, and flexible. Solutions based on satisficing theory can exhibit complex behavior, yet are based on relatively simple algorithms that are not specific to any fixed problem scenario. While many envisioned extensions to satisficing theory remain to be explored, our results suggest that a satisficing-based approach can offer good performance and safety for the challenging problem of resolving conflicts between aircraft in a decentralized ATC system. In particular, our satisficing algorithms demonstrate emergent behavior that matches previous solutions resulting from the analysis of special cases. This suggests that the rule set implemented within the satisficing framework captures much of the essence of ideal conflict resolution using constant-speed heading-change maneuvers.

REFERENCES

[1] G. Weiss, "Europe favors united air traffic control system," *IEEE Spectr.*, vol. 39, no. 11, pp. 22–23, Nov. 2002.

[2] T. S. Perry, "In search of the future of air traffic control," *IEEE Spectr.*, vol. 34, no. 8, pp. 18–35, Aug. 1997.

[3] D. K. Chin and F. Melone, "Using airspace simulation to assess environmental improvements from free flight and CNS/ATM enhancements," in *Proc. 1999 Winter Simul. Conf.*, pp. 1295–1301.

[4] "National Research Council Panel on Human Factors in Air Traffic Control Automation," in *The Future of Air Traffic Control: Human Factors and Automation*, C. D. Wickens, A. S. Mavor, R. Parasuraman, and J. P. McGee, Eds. Washington, DC: National Academy Press, 1998.

[5] J. C. Hill, J. K. Archibald, W. C. Stirling, and R. L. Frost, "A multi-agent system architecture for distributed air traffic control," presented at the AIAA Guid., Navigat. Control Conf., San Francisco, CA, Aug. 2005, Paper AIAA-2005-6049.

[6] J. C. Hill, F. R. Johnson, J. K. Archibald, R. L. Frost, and W. C. Stirling, "A cooperative multi-agent approach to free flight," in *Proc. 4th Int. Joint Conf. Auton. Agents Multiagent Syst. (AAMS 2005)*. New York: ACM Press, pp. 1083–1090.

[7] F. R. Johnson, J. C. Hill, J. K. Archibald, R. L. Frost, and W. C. Stirling, "A satisficing approach to free flight," in *Proc. IEEE Int. Conf. Netw., Sens. Control*, 2005, pp. 123–128.

[8] R. Axelrod, *The Complexity of Cooperation*. Princeton, NJ: Princeton Univ. Press, 1997.

[9] W. C. Stirling, *Satisficing Games and Decision Making: With Applications to Engineering and Computer Science*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[10] W. C. Stirling, "Social utility functions—Part I. Theory," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 4, pp. 522–532, Nov. 2005.

[11] W. C. Stirling and R. L. Frost, "Social utility functions—Part II. Applications," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 4, pp. 533–543, Nov. 2005.

[12] J. Krozel, M. Peters, K. D. Bilimoria, C. Lee, and J. S. B. Mitchell, "System performance characteristics of centralized and decentralized air traffic separation strategies," presented at the 4th USA/Eur. Air Traffic Manag. R&D Seminar, Santa Fe, NM, Dec. 2001.

[13] G. J. Pappas, C. Tomlin, and S. Sastry, "Conflict resolution for multi-agent hybrid systems," in *Proc. IEEE Conf. Decis. Control*, Kobe, Japan, Dec. 1996, vol. 2, pp. 1184–1189.

[14] C. Tomlin, G. J. Pappas, and S. Shastry, "Noncooperative conflict resolution," in *Proc. IEEE Conf. Decis. Control*, San Diego, CA, Dec. 1997, vol. 2, pp. 1816–1821.

[15] C. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 509–521, Apr. 1998.

[16] C. Tomlin, I. Mitchell, and R. Ghosh, "Safety verification of conflict resolution maneuvers," *IEEE Trans. Intell. Transp. Syst.*, vol. 2, no. 2, pp. 110–120, Jun. 2001.

[17] R. Ghosh and C. Tomlin, "Maneuver design for multiple aircraft conflict resolution," in *Proc. Amer. Control Conf.*, Chicago, IL, Jun. 2000, vol. 1, pp. 672–676.

[18] I. Hwang and C. Tomlin, "Protocol-based conflict resolution for finite information horizon," in *Proc. Amer. Control Conf.*, Anchorage, AK, May 2002, vol. 1, pp. 748–753.

[19] J. Kosecka, C. Tomlin, G. Pappas, and S. Sastry, "Generation of conflict resolution maneuvers for air traffic management," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Grenoble, France, Sep. 1997, vol. 3, pp. 1598–1603.

[20] J. Kosecka, C. Tomlin, G. Pappas, and S. Sastry, "2 1/2 D conflict resolution maneuvers for ATMS," in *Proc. IEEE Conf. Decis. Control*, Tampa, FL, Dec. 1998, vol. 3, pp. 2650–2655.

[21] D. Dugail, E. Feron, and K. Bilimoria, "Stability of intersecting aircraft flows using heading change maneuvers for conflict avoidance," in *Proc. Amer. Control Conf.*, Anchorage, AK, May 2002, pp. 760–766.

[22] Z.-H. Mao, E. Feron, and K. Bilimoria, "Stability and performance of intersecting aircraft flows under decentralized conflict avoidance rules," *IEEE Trans. Intell. Transp. Syst.*, vol. 2, no. 2, pp. 101–109, Jun. 2001.

[23] S. Resmerita and M. Heymann, "Conflict resolution in multi-agent systems," in *Proc. IEEE Conf. Decis. Control*, Maui, HI, Dec. 2003, vol. 2, pp. 2537–2542.

[24] S. Resmerita, M. Heymann, and G. Meyer, "A framework for conflict resolution in air traffic management," in *Proc. IEEE Conf. Decis. Control*, vol. 2, Maui, HI, Dec. 2003, pp. 2035–2040.

[25] A. Bicchi and L. Pallottino, "On optimal cooperative conflict resolution for air traffic management systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 4, pp. 221–232, Dec. 2000.

[26] L. Pallottino, E. M. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 3–11, Mar. 2002.

[27] L. Pallottino, A. Bicchi, and S. Pancanti, "Safety of a decentralized scheme for free-flight ATMS using mixed integer linear programming," in *Proc. Amer. Control Conf.*, Anchorage, AK, May 2002, pp. 742–747.

[28] E. Frazzoli, L. Pallottino, V. Scordio, and A. Bicchi, "Decentralized cooperative conflict resolution for multiple nonholonomic vehicles," presented at the AIAA Guid., Navigat. Control Conf., San Francisco, CA, Aug. 2005.

[29] R. A. Paielli and H. Erzberger, "Conflict probability estimation for free flight," *J. Guid., Control, Dyn.*, vol. 20, no. 3, pp. 588–596, May/Jun. 1997.

[30] M. Prandini, J. Hu, J. Lygeros, and S. Sastry, "A probabilistic approach to aircraft conflict detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 4, pp. 199–220, Dec. 2000.

[31] J. Rong, S. Geng, J. Valasek, and T. R. Ioerger, "Air traffic control negotiation and resolution using an onboard multi-agent system," in *Proc. 21st Digital Avionics Syst. Conf.*, 2002, vol. 2, pp. 7B2-1–7B2-12.

[32] Y.-J. Chiang, J. T. Klosowski, C. Lee, and J. S. B. Mitchell, "Geometric algorithms for conflict detection/resolution in air traffic management," in *Proc. IEEE Conf. Decis. Control*. San Diego, CA: IEEE Press, Dec. 1997, pp. 1835–1840.

[33] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 4, pp. 179–189, Dec. 2000.

James K. Archibald (S'85–M'86–SM'05) received the B.S. degree (*summa cum laude*) in mathematics from Brigham Young University, Provo, UT, in 1981, and the M.S. and Ph.D. degrees in computer science from the University of Washington, Seattle, in 1983 and 1987, respectively.

Since 1987, he has been with the Department of Electrical and Computer Engineering, Brigham Young University. His current research interests include robotics and multiagent systems.

Dr. Archibald is a member of the Association for Computing Machinery and the Phi Kappa Phi.

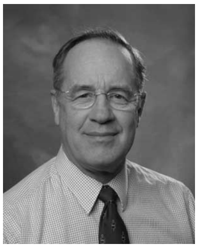
Jared C. Hill received the B.S. and M.S. degrees in electrical engineering from Brigham Young University, Provo, UT, in 2005 and 2006, respectively.

He is currently a Firmware Engineer at Rincon Research Corporation, Tucson, AZ. His research interests include embedded systems, wire sensor networks, and image processing.

Nicholas A. Jepsen received the B.S. degree (*cum laude*) in electrical engineering in 2006 from Brigham Young University, Provo, UT, where he is currently working toward the J.D. degree.

His research interests include digital design, machine vision, and multiagent systems.

Mr. Jepsen is a member of the Tau Beta Pi and Eta Kappa Nu Honor Societies.



Wynn C. Stirling received the B.A. (Hons.) degree (*magna cum laude*) in mathematics and the M.S. degree from the University of Utah, Salt Lake City, in 1969 and 1971, respectively, and the Ph.D. degree from Stanford University, Stanford, CA, in 1983, all in electrical engineering.

From 1972 to 1975, he was with Rockwell International Corporation, Anaheim, CA. During 1975–1984, he was with ESL, Inc., Sunnyvale, CA. He is currently with the Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT, where, in 1984, he was a Professor. His research interests include multiagent decision theory, estimation theory, information theory, and stochastic processes.

Dr. Stirling is a member of the Phi Beta Kappa and Tau Beta Pi Honor Societies.

Richard L. Frost received the B.S. (Hons.) degree in physics in 1975, and the M.S. and Ph.D. degrees in electrical engineering from the University of Utah, Salt Lake City, in 1977 and 1979, respectively.

In 1979, he was with the Lincoln Laboratory, Massachusetts Institute of Technology. From 1981 to 1984, he was a faculty member at the University of Utah. He was with Sperry Corporation for three years. In 1987, he joined the faculty of the Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT. His current research interests include information theory, signal processing, and multiagent decision theory.