A Satisficing Approach to Aircraft Conflict Resolution

J. K. Archibald, J. C. Hill, N. A. Jepsen, W. C. Stirling, and R. L. Frost
Department of Electrical and Computer Engineering
Brigham Young University

***Abstract*— Future generations of air traffic management systems will give appropriately equipped aircraft the freedom to change flight paths in real-time. This will require a conflict avoidance and resolution scheme that is both decentralized and cooperative. Satisficing game theory provides a theoretical framework in which autonomous decision makers may coordinate their decisions. 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 which 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 describe a solution to aircraft conflict resolution based on satisficing game theory. We present simulation results of 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 multi-agent problems in which self-interested participants are inclined to cooperation.**

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]. Part of the anticipated solution is *free flight*, an operating environment in which pilots are given increased autonomy to select or modify their flight path in real time [4]. In the current system, centralized air traffic controllers issue directives that guide aircraft along prespecified corridors; in free flight, controllers will monitor flight paths and intervene only when necessary.

Safety is the most important aspect of ATC; whether centralized or distributed, its purpose is to maintain adequate separation between aircraft. Substantial changes in the ATC system will take place only experts conclude that the modified system will be stable and work reliably under all possible conditions. Relative to the safety of the current centralized system, free flight offers certain advantages; the increased automation is likely to reduce controller workloads, and the overall system is less affected by failures in the ground control system [5].

To ensure separation from other aircraft during free flight, pilots are likely to rely on automated decision support systems that make use of recent advancements in navigation aids, communication technologies, and computing power. In general, separation is maintained by detecting and resolving possible conflicts before proximity violations can occur. Many conflict resolution techniques have been proposed, particularly for use in open airspace between airports. This *enroute* airspace is an attractive candidate for automation because plane density is low compared with airport airspace and because rigid scheduling is unnecessary.

This paper presents a framework for an intelligent decision-support system for aircraft conflict resolution. In our approach, aircraft are viewed as autonomous agents inclined to cooperative decision making [6–8]. The framework is based on *satisficing game theory*, a decision making approach that allows the modeling of complex social relationships. Satisficing agents are able to condition their own preferences on the preferences of others, allowing agents to compromise in order to achieve both individual and group goals. In contrast, true cooperation is difficult to obtain when agents employ conventional decision-making approaches based on optimization of a self-centered utility function.

In our approach, it is assumed that each aircraft is aware of critical information (e.g., position, velocity, destination) of all aircraft within a particular communication radius. We also assume a popular model of 2D airspace in which all aircraft fly at the same altitude and at constant velocity: conflicts can be avoided only through heading change maneuvers. In contrast with typical conflict resolution schemes that resolve conflicts pairwise, satisficing agents can consider many projected conflicts in the choice of next avoidance maneuver. Within the satisficing framework, many possible behaviors can be specified; we describe two that are suitable for real-time implementation and present simulation results that show them to compare favorably with previously published approaches, both in terms of safety (maintaining separation) and efficiency (reducing flight length).

Our results are strong indicators that satisficing is a promising and viable method of synthesizing multiagent systems in which self-interest and cooperation are both naturally present. While satisficing shows promise as a framework for a multiagent solution to conflict resolution in free flight, a full ATC system based on this approach would require much more analysis and additional components.

The remainder of this paper is organized as follows: In the next section, we describe satisficing game theory, the basis for our cooperative multiagent approach to conflict resolution. In Section III, we describe two behavioral models of conflict resolution constructed within a satisficing framework, detailing precisely how a decision option is selected. To illustrate the operation of the resolution schemes, Section IV presents an example conflict involving two aircraft and discusses the resulting computation and outcome. Previously proposed schemes for conflict resolution are summarized in V, and appropriate measures of system performance are discussed in VI. Section VII describes our simulator and presents simulation results in a variety of conflict scenarios. Where possible, results of previous schemes are included for comparison. Section VIII presents a brief summary and discusses further work.

II. Satisficing Game Theory

Game theory, as established by von Neumann and Morgenstern [9], provides the logical foundation for much of multiagent decision making. Even if a game-theoretic format is not explicitly used, game theoretic logic — maximization of expected utility — is the principle that guides much of the theory. Unfortunately, this view of rationality possesses serious limitations — only individuals can optimize. If a group were to optimize its behavior, then it must act as if it were a single entity, but the resulting solution would not necessarily be optimal, or even acceptable, for its individual members. Individual optimization is the Occam's razor of social relationships: every agent will do the best thing for itself *regardless of the effect doing so has on others*. Such a sociology is simply not sophisticated enough to accommodate the type of cooperative behavior that is essential to the operation of a distributed ATC system.

Furthermore, optimization is not a well-conceived concept with open systems, even for individuals, where each decision maker responds to its immediate environment. What may be viewed as a good joint decision by one agent from its limited perspective may be viewed as a bad one by others with different perspectives. Thus, the agents must be flexible in their decision making, and the group must possess the capability to negotiate to reach a mutually agreeable decision. Optimization, by its very structure, precludes such flexible behavior.

A. Social Utilities

To formulate a more sophisticated concept, it is necessary to return to the "headwaters" of rational choice; namely, to review the way utilities are formed. Conventional utility theory, as established by von Neumann and Morgenstern, assumes that each individual possesses a preference ordering of its set of possible actions as a function of the actions that others may take. Thus, considering a set of n decision makers, the ith participant has a utility function $\phi_i(u_1, u_2, \ldots, u_n)$ where u_k is the action taken by the kth agent. It is not until these utilities are juxtaposed in a payoff array that opportunities for conflict and cooperation become evident. Under this paradigm, the possibilities for cooperation and conflict are not considered when defining the utilities. It is as if each participant forms its utilities in a social vacuum, without taking into consideration any social relationships that may exist between agents. *This is a fundamental limitation of von Neumann-Morgenstern utility theory.*

One way to overcome this limitation is to form *social utilities* as functions of agent *preferences for action*, rather than directly as functions of agent actions, as is done with von Neumann-Morgenstern utilities. To achieve this goal, let us consider the notion that each agent has two roles, or personas. One persona focuses on achieving the fundamental goal of the decision problem, regardless of cost, while the other persona focuses on conserving resources and reducing costs without worrying about achieving the goal. Together, these two (possibly conflicting) personas provide a complete description of a decision maker who must balance the desire to achieve its goal with the cost of doing so. We require two utilities to account for the preferences of these two personas. One utility characterizes the *selectability* of the options available to the decision maker; that is, the degree of effectiveness of the options with respect to achieving the goal without worrying about cost or other consequences. The other utility characterizes *rejectability* of the options; that is, the degree to which resources are consumed (e.g., energy costs, social costs, time delays, exposure to hazards). These two utilities are normalized to be mass functions.

In the multiagent case, they are multivariate mass functions that permit the simultaneous characterization of a multiagent decision system. By normalizing the utilities, they assume the same mathematical structure as probability mass functions, and therefore we may characterize relationships such as independence and conditioning that are analogous to the probabilistic notions. That is, they posses the same syntax as do probabilities, but with different semantics. The justification for this structure is described in detail elsewhere [10, 11].

Let p_S and p_R denote selectability and rejectability mass functions, respectively, and let p_{SR} denote the joint mass function when simultaneously taking into consideration the selectable and rejectable attributes of the options. For an n agent system, the joint selectability/rejectability is a mass function with $2n$ variables of the form

$$p_{S_1S_2\cdots S_nR_1R_2\cdots R_n}(u_1,u_2,\ldots,u_n;v_1,v_2,\ldots,v_n);$$

where $S_1S_2\cdots S_n$ corresponds to the collection of selectability personas and $R_1R_2\cdots R_n$ corresponds to the collection of rejectability personas. This function is called the *interdependence function*. The variables u_i, $i=1,\ldots,n$ correspond to the options available to the ith agent *as viewed from the perspective of goal achievement* and the variables v_i, $i=1,\ldots,n$ correspond to the options available to the ith agent *as viewed from the perspective of resource conservation*. By characterizing these joint preferences with a multivariate mass function, we are able to account for all of the relationships that exist between all personas of a multiagent decision problem in much the same way as a joint probability mass function characterizes the joint behavior of a random vector.

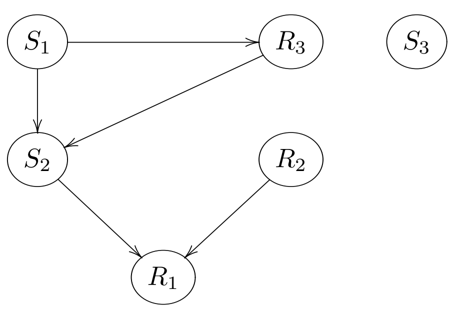

Fig. 1. Network of influence flows

The interdependence function captures all of the decision-making considerations that may affect a multiagent decision system. Fortunately, its construction can often be guided by appealing to the conditional influences that exist between agent personas. To illustrate, consider the directed acyclic graph (DAG) displayed in Figure 1, which corresponds to a three-agent system (and hence it has three selectability personas and three rejectability personas). In this system, the selectability of S_1 influences S_2 and R_3. Furthermore, R_3 influences S_2, and both S_2 and R_2 influence R_1. Finally, S_3 neither influences nor is influenced by any other persona. The interdependence function of this influence structure may be expressed as

$$p_{S_1S_2S_3R_1R_2R_3}=p_{R_1|S_2R_2}\cdot p_{S_2|S_1R_3}\cdot p_{R_3|S_1}\cdot p_{S_1}\cdot p_{R_2}\cdot p_{S_3},$$

where arguments have been suppressed in the interest of brevity. The conditional mass functions represent the influence flows between nodes of the graph. For example, $p_{R_1|S_2R_2}(v_1|u_2;v_2)$ expresses the amount of rejectability that Agent 1 should ascribe to option v_1, given that Agent 2 were to select option u_2 in the interest of achieving its own goal and reject option v_2 on the basis of conserving resources.

Conditional utilities permit a sophisticated form of altruism. In contrast to what may be termed *categorical altruism*, where an agent unconditionally changes its preferences in order to benefit another, conditional utilities permit a concept of *situational altruism*, whereby an agent may change its preferences as a function of the preferences of another, rather than unilaterally. The essential difference between these two concepts is that, with the former, the agent sacrifices its own interest regardless of the other's desire to take advantage of it; with the latter, the agent is willing to accommodate, at least to some degree, the preferences of another in lieu of its own preferences if, but only if, the other wishes to take advantage of the offered largesse. Otherwise, the agent would be governed by its own preferences and would avoid needless sacrifice. Situational altruism may be either benevolent, where an agent sacrifices its performance to benefit another, or malevolent, where an agent sacrifices to harm another. This more sophisticated notion of altruism is difficult to model with von Neumann-Morgenstern utilities. Conditional utilities, however, are explicitly designed to accommodate situational altruism and other forms of sophisticated social behavior.

B. Satisficing Games

Satisficing game theory [12] provides a mathematically rigorous way to make such compromises in a controlled way. Let us consider a set of n decision makers, and let U_i denote the set of options available to agent i, $i=1,\ldots,n$. A satisficing game is the triple $(n, U_1\times\cdots\times U_n, p_{S_1\cdots S_nR_1\cdots R_n})$. To solve this game, we must compute the joint selectability and rejectability marginals as

$$p_{S_1\cdots S_n}(u_i,\ldots,u_n) = \sum_{v_1\in U_1}\cdots\sum_{v_n\in U_n} p_{S_1\cdots S_n R_1\cdots R_n}(u_i,\ldots,u_n,v_i,\ldots,v_n) \quad (1)$$

and

$$p_{R_1\cdots R_n}(v_i,\ldots,v_n) = \sum_{u_1\in U_1}\cdots\sum_{u_n\in U_n} p_{S_1\cdots S_n R_1\cdots R_n}(u_i,\ldots,u_n,v_i,\ldots,v_n) \quad (2)$$

and the individual selectability and rejectability marginals as

$$p_{S_i}(u_i) = \sum_{u_1\in U_1}\cdots\sum_{u_{i-1}\in U_{i-1}}\sum_{u_{i+1}\in U_{i+1}}\cdots\sum_{u_n\in U_n} p_{S_1\cdots S_n}(u_1,\ldots,u_{i-1},u_i,u_{i+1},\ldots,u_n). \quad (3)$$

and

$$p_{R_i}(u_i) = \sum_{u_1\in U_1}\cdots\sum_{u_{i-1}\in U_{i-1}}\sum_{u_{i+1}\in U_{i+1}}\cdots\sum_{u_n\in U_n} p_{R_1\cdots R_n}(u_1,\ldots,u_{i-1},u_i,u_{i+1},\ldots,u_n). \quad (4)$$

The *jointly satisficing solution at caution level q* of a satisficing game is the subset of all option vectors such that the joint selectability is at least as great as the caution level multiplied by the joint rejectability, that is,

$$\mathbf{\Sigma}_q = \big\{(u_1,\ldots,u_n)\in U_1\times\cdots\times U_n : p_{S_1\cdots S_n}(u_1,\ldots,u_n) \geq q\cdot p_{R_1\cdots R_n}(u_1,\ldots,u_n)\big\}. \quad (5)$$

The scalar q represents a relative weight between achieving the goal and conserving resources; nominally, q = 1.

The individually satisficing solutions for each agent are obtained from the marginal selectability and rejectability functions, yielding the *individually satisficing solutions:*

$$\Sigma_q^i = \big\{u_i\in U_i : p_{S_i}(u_i)\geq q\cdot p_{R_i}(u_i)\big\}.$$

The *satisficing rectangle* is the product set of the individually satisficing sets, namely,

$$\Re = \Sigma_q^1\times\cdots\times\Sigma_q^n.$$

In general, the satisficing rectangle will not be the same as the jointly satisficing set; they may even be disjoint. However, the following theorem relates the two sets.

Theorem 1: The Negotiation Theorem. If u_i is individually satisficing for agent i, that is, $u_i\in\Sigma_q^i$, then it must be the ith element of some jointly satisficing vector $(u_1,\ldots,u_n)\in\Sigma_q$.

A proof of this theorem was previously published [12]. The content of this theorem is that *no one is ever completely frozen out of a deal — every decision maker has, from its own perspective, a seat at the negotiating table*. This condition is perhaps the weakest condition under which negotiations are possible. Setting $q = 1$ grants equal weight to achieving the goal and conserving resources and ensures that the satisficing sets are not empty. In practice, q can be viewed as a negotiation parameter; reducing q increases the size of the satisficing sets, and permits the participants to lower their standards in a controlled way to reach a compromise — a solution that is individually satisficing for each member of the group and is jointly satisficing for the group.

III. A Satisficing Approach to ATC

Although existing multi-agent solution techniques might not offer a satisfactory solution because of the inherent complexity, ATC can be viewed as a multi-agent problem. In an implementation of free flight, decisions will be made by individual aircraft in a distributed fashion using incomplete local knowledge; this matches essential characteristics of multi-agent systems. The challenges of ATC with high flight densities strain the capability of any solution technique, particularly since any useful solution must satisfy stringent safety criteria. Moreover, it is infeasible to create a solution based on exhaustive enumeration of possible scenarios, so any solution technique must be truly general, flexible, and scalable.

In our representation of the problem, all aircraft fly at the same altitude. At each time step (currently 1 second intervals), each aircraft chooses from one of five directional options, including flying straight, moderate turns (2.5 degrees) left or right, and sharper turns (5 degrees) left or right.

The first step in applying satisficing decision theory to the problem is to create influence flows that describe relationships between agents in the system. Because the problem is dynamic and the number of neighboring aircraft varies, static influence flows cannot accurately represent the system. We employ an algorithm to create these flows dynamically; this is equivalent to creating a ranking of aircraft.

Our algorithm for ranking the aircraft is relatively simple. Initially, aircraft within fifty miles of their destination are placed in a group with priority over remaining aircraft which are also treated as a group. Within each of the two groups, aircraft are ranked according to the delay they have accumulated (relative to flying directly to their destination). Aircraft with more delay are ranked higher. Finally, aircraft in the same group with the same delay are ranked according to their length of time in flight, with longer flight times getting higher rankings. Once the aircraft are ranked, a directed acyclic graph can be created representing influence flows.

At each time step, every aircraft receives a list of all aircraft within a fifty mile radius of itself. The aircraft are then ranked using the algorithm described above. A *priority list* is then created by removing from the original list the aircraft itself, all aircraft ranked lower than itself, and all other aircraft with which it has no possibility of a conflict.

The rejectability function of each aircraft is determined by anticipated conflicts. For each directional option, the intended direction of each (higher-priority) aircraft on the priority list is compared to see if flying in that direction would cause a conflict. Each conflict that is detected adds a predetermined weight to that option. Predicted collisions carry a higher weight than near misses. After all higher ranking aircraft have been considered, the weight of each option is then normalized over the option space into a probability mass function. The rejectability utility thus indicates which directional choices are most (or least) likely to lead to conflicts with higher priority aircraft. Note that the utility functions of other aircraft do not influence the calculation of rejectability.

The *base selectability* of each directional option is determined by its difference from the desired heading of the aircraft. An option that takes the aircraft more directly to its destination will have higher base selectability. The values are then normalized over the option space. These values do not reflect social utility since they are not affected by the preferences of other aircraft.

The next step is to create a DAG representing the influence flow for the full selectability function. In our formulation, the selectabilities of all aircraft with higher ranking influence the selectability of the current aircraft. One by one, the directional options of the current aircraft are compared against the directional preferences of aircraft with higher rankings. Using Pearl's Belief Propagation Algorithm [13], we sum over the option set of all other aircraft and normalize, producing the selectability function of the current aircraft.

Our preliminary investigation has resulted in the development of two models that produce good performance across a variety of test scenarios. In the *full model*, each aircraft makes use of all available information in determining the selectability of its neighbors. Rather than using only the base selectability of higher-ranked aircraft, each aircraft uses its (incomplete) knowledge of the environment around aircraft that influence it to approximate their full selectability, including their ranking of aircraft and influences from those aircraft. Although this approach cannot model the selectability of influencing aircraft exactly, it does improve the overall performance.

The *simplified model* takes advantage of the geometric similarity of the base selectabilities of all higher-ranked aircraft. Influencing aircraft are grouped according to which of the five options has the highest base selectability. The number of aircraft in each group is then used as the weight for each directional option. The model also calculates the number of conflicts each group will create given each option. It then normalizes over the option set to produce the selectability function for a given aircraft.

Once the selectability and rejectability functions for a given aircraft have been determined, the set of satisficing options can be computed. Given a satisficing set, there are several defensible options for picking a single decision. Agents willing to tolerate risk for high gains could maximize selectability. Risk averse agents could minimize rejectability, but this gives no guarantee of progress towards the goal. In our approach, we choose the satisficing option with the largest difference between selectability and rejectability. This insures the greatest possible progress towards the goal relative to the risk incurred.

IV. An Example Scenario

To illustrate the process of satisficing-based decision making, consider an example involving just two aircraft, A and B, that are both headed directly to their destinations. If both continue on their current heading, the aircraft will collide. Aircraft A and B are 10 and 5 minutes behind schedule respectively, so A is ranked higher.

The selectability and rejectability of aircraft A are relatively 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 based on its delay, it will not B in computing its selectability. Similarly, because A sees no conflicts with aircraft on its priority list, it determines that p_{R_A} is a uniform distribution over the option space.

According to our algorithm, B will sacrifice some efficiency to resolve the conflict. In effect, B will calculate which of its options will take it the least off course and still resolve the conflict. p_{S_B} The selectability of B depends on the selectability of A and the current distance between the aircraft. (As noted above, the selectability of A is the largest for the option of going straight.) If a slight heading change for B to the right or left will resolve the conflict, then so will sharp turns, so all four turning options have the same selectability since they avoid conflicts. If the distance between the aircraft is such that a sharp turn is required, only the sharp turns will receive the highest values of selectability for B.

The rejectability of B is calculated by looking at possible conflicts. If moderate turns avoid conflicts, going straight will be assigned the value one and all other options will have value zero. On the other hand, 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 zero. Ties are broken by picking the option that takes the aircraft closer to its destination. The smallest detour with no conflicts will be chosen — this corresponds to maximizing the difference between selectability and rejectability values for each option.

If the selectability of A were highest for a sharp right turn (to put it back on course), the result would be quite different. The selectability of B would see that A would prefer a sharp right turn, and therefore B's options to turn left (into A's anticipated path) would have lower selectability values. Rejectability values would remain the same because they are independent of the preferences of other aircraft. This means that B would choose to go straight because it is the direction that would take the aircraft towards its destination while still avoiding conflicts. This is an example where the benefits of situational altruism and the efficiency it introduces through cooperation are easily observed.

V. Previous Conflict Resolution Schemes

Widespread interest in free flight has resulted in a wide 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 [14], all of which are implemented as constant-speed heading change maneuvers. The centralized approach determines the set of conflicts arising in the next eight minutes if no corrective actions were taken. Aircraft are partitioned into clusters such that no aircraft has a conflict with an aircraft in a different 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 strategies, aircraft resolve their own conflicts as they are detected. Multiple conflicts within the eight-minute 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, Tomlin and Sastry propose a decentralized conflict architecture that views the aircraft as a hybrid system incorporating both discrete events and individual dynamics modeled by differential equations [15]. Projected conflicts are resolved in two phases. First, non-cooperative 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 then employ 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. In [16] the non-cooperative game-theoretic approach is expanded to include both path deviations and speed variations. Subsequent extensions have included: a complete methodology for generating provably safe conflict heading-change and velocity-change resolution maneuvers for two aircraft [17, 18], a comparison of the hybrid approach relative to a continuous kinematic planner proven to be safe with up to three aircraft [19], and a protocol for resolving conflicts with instantaneous heading-change maneuvers when conflicting aircraft are out of direct communication range [20].

Kosecka *et al.* use distributed motion planning algorithms based on potential and vortex fields are used to generate prototype heading-change maneuvers for multi-aircraft conflicts; actual maneuvers are flyable, piece-wise linear approximations of the prototypes which can be proven safe using hybrid verification techniques [5]. Selected maneuvers are shown for up to four aircraft. This work was extended in [21] to include altitude change maneuvers if heading changes in the horizontal plane were unable to resolve the conflict.

Dugail, Feron and Bilimoria analyze a decentralized conflict resolution scheme for two perpendicular flows of air traffic that intersect at a fixed point [22]. 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 [23], 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 Heymann describe an approach that partitions the airspace into static cells that may be occupied by only one aircraft at a time, thus ensuring separation [24, 25]. 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 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 [26]. 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 [27], to a decentralized hybrid approach with instantaneous heading changes including up to three aircraft [28], and to a decentralized hybrid system for an arbitrary number of nonholonomic vehicles [29].

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 [30]. 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) [31]. 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 an cooperative agent-based solution to conflict resolution based on constraint satisfaction problems [32]. Using direct communication, conflicting agents negotiate pairwise 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 [33]. 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 non-intersecting 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 [34]. 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. System Performance Measures

A variety of metrics have been employed to evaluate algorithms that maintain separation and resolve conflicts in ATC. We discuss the most promising of these below.

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 scenario studied. Surprisingly, many papers describing algorithms for ATC 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 feet of each other, and near misses, which occur when aircraft come within five miles of each other.

B. System Efficiency

System efficiency (SE) is loosely defined as the degree to which an aircraft is able to follow its ideal flight path [14]. In general, conflict resolution maneuvers will cause each aircraft to deviate from its ideal path and to consume more resources. For free flight to be successful, conflicts must be avoided while maintaining acceptable efficiency. Since all aircraft are identical and cruise at the same speed, and since conflict resolution maneuvers are constant-speed heading-change maneuvers, the efficiency can be calculated by tracking the time it takes each aircraft to get from its starting point to its destination. The actual flight time, t_t, is compared with the aircraft's ideal flight time, t_i, determined when the aircraft first appeared in the simulation. The delay time for each aircraft is calculated as $t_d = t_t - t_i$. For a system with N total aircraft having completed their flights, the system efficiency is computed by

$$SE = \frac{1}{N} \cdot \sum_{i=1}^{N} \Big(\frac{t_i}{t_{d_i} + t_i} \Big).$$

In the ideal system, all aircraft are able to fly their ideal paths, so $SE = 1$. As traffic density and congestion increase, aircraft deviate further from their ideal paths, and SE decreases in value.

C. System Stability

System stability (SS) is defined as the degree to which conflict resolution maneuvers create new conflicts with adjacent aircraft that must be resolved [14]. As originally proposed, the metric counts as conflict threats all separation violations predicted to occur within a 50-mile radius. Define S_1 as the set of conflicts that occur if all aircraft fly their ideal flight paths, and S_2 as the set of all conflict threats that occur when conflict resolution maneuvers are employed. Then SS is given by

$$SS = \frac{|S_1|}{|S_2|}.$$

If SS has the value 0.1, then 10 times more conflict threats occurred with the resolution algorithm employed than occurred with no conflict resolution maneuvers.

When a truly cooperative algorithm is employed, conflicts can be predicted and avoided before they are ever officially tallied as a threat. For example, if aircraft A wants to turn right to avoid one conflict, but that turn would create a conflict with lower ranked aircraft B, B is likely to alter its path to avoid the potential conflict before it is ever realized as a threat. For cooperative satisficing algorithms, S_2 can be much smaller than S_1, resulting in very large values of SS. Because this measure as originally formulated does not give much insight into satisficing algorithms, we did not make use of it in our studies.

VII. Results

Our simulation environment is similar to simulations used by other researchers [14, 22, 23]. All aircraft are constrained to fly at the same altitude. While altitude changes are an effective means of resolving conflicts, they were disallowed to make it simpler to create high aircraft densities that would stress conflict resolution techniques. Furthermore, all aircraft travel at the constant velocity of 500 mph. As previously described, once each second each plan receives a list of information about all aircraft within 50 miles. Each aircraft uses this information to make a decision from its option space. Maneuvers are modeled by instantaneous heading changes. After making local decisions, all aircraft update their own headings and positions, and then that information is distributed to all other aircraft within the 50-mile limit and the display screen is updated with the new information.

While certain patterns of conflicting aircraft are likely to be common in free flight, it would be impossible to enumerate all possible interaction geometries. For this reason, we feel that any conflict resolution algorithm must be evaluated across a wide range of scenarios. Our study included scenarios with fixed geometries as well as scenarios with completely random traffic patterns and arbitrary traffic density. None of our scenarios include obstacles, problematic weather areas, or no-fly zones. We describe each scenario in detail and report the results of our simulation runs.

A. Random Flights

This environment, based on a model used by Krozel and colleagues [14], consists of two concentric circles in open air space. Aircraft appear at random points on the outer circle (radius 120 miles) and are assigned a random destination point on the inner circle (radius 100 miles). The 20 mile buffer between the circles is used to ensure that no aircraft are generated already in conflict with other aircraft. Because this scenario tests a wide range of conflict types, it is a good test of any algorithm's ability to deal safely and efficiently with unpredictable patterns. Table I shows the average of results from four simulation runs of the full model at each reported density, each run lasting 50 minutes.

Aircraft	Near Miss	Collisions	Efficiency (%)
80	26	.25	97.1
70	19.8	.25	97.7
60	11	.25	98.0
50	6.25	0	98.1
45	4.75	0	98.6
40	3.75	0	99.1
30	1.75	0	99.1
20	0	0	99.6

TABLE I
RESULTS FOR RANDOM FLIGHT SCENARIO

Note the high efficiency that results, even with extremely dense traffic. High efficiency equates to a decrease in flight delay and in resources consumed — desirable outcomes for both passengers and industry. Results with lower traffic densities were not included to save space; in those cases, the conflict count is zero and the efficiency increases as the density decreases.

B. Choke Point

In this scenario, based on a model used by Pallottino and colleagues[27], all aircraft begin from evenly spaced points on a circle with radius 50 miles. Each aircraft's destination is the point on the circle opposite its starting point. Thus, all aircraft are set to pass through the center of the circle at the same time, creating a considerable challenge for any conflict resolution algorithm. This scenario is good for testing computational load, as all aircraft are in the same influence net. Because of this complexity, the full model requires too many calculations to run in real time. The results shown are for the simplified model only. While this scenario is unrealistic, it does provide insight into how a conflict resolution algorithm deals with a complicated situation.

Figure 2 shows a series of screen shots as 32 aircraft attempt to reach their destination. Although the conflict resolution algorithm is in no way preprogrammed to handle this specific problem scenario, the behavior that emerges from the satisficing approach is very similar to the previously published solution[27].

Table II summarizes simulation results for a range of problem sizes. (Results from a single run are re-

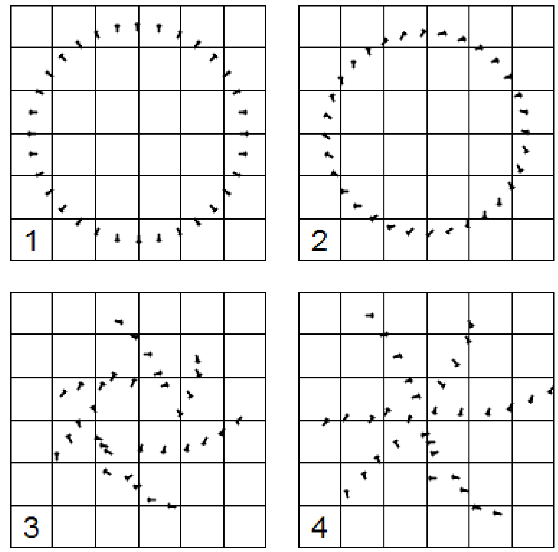

Fig. 2. Choke Point Snapshots With 10 Mile Grid

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

TABLE II
RESULTS FOR THE CHOKE POINT SCENARIO

ported; neither the scenario nor the algorithm include randomness, so multiple simulation runs for a given aircraft density give identical results.) Because the circle is of fixed size, increases in the number of aircraft cause corresponding increases in traffic density. The satisficing approach is able to achieve high efficiency while completely avoiding collisions for the densities reported.

C. Perpendicular Flows

In this scenario, similar to case study introduced by Dugail and colleagues[22], two flows of traffic are routed to intersect in the middle of the 100×100 mile world. The aircraft trajectories in each flow are generated with an initial separation just over five miles from the preceding aircraft, ensuring that each can make small avoidance maneuvers without violating the required separation distance from the following aircraft.

Distribution	Flights	Near Miss	SE (%)
Constant Flow	97	29	98.2
$\mu = 25$	97	26	98.8
$\mu = 20$	94	24	97.4
$\mu = 15$	93	6	99.8
$\mu = 10$	87	4	99.7

TABLE III

RESULTS FOR PERPENDICULAR FLOW SCENARIO

Table III reports simulation results for the simplified model for this scenario. In the case labeled "constant flow", a new aircraft is added to each flow every 40 seconds. For the other runs, strings of aircraft separated by 40 seconds were generated with length given by the Gaussian random variable $\mathcal{N}(\mu,1)$. At the end of each string, there is a gap of 80 seconds before the aircraft that begins the next string. The results illustrate that the insertion of occasional gaps can increase both safety and efficiency. Overall, the satisficing algorithm is impressive in both its efficiency and its safety record: no collisions occurred.

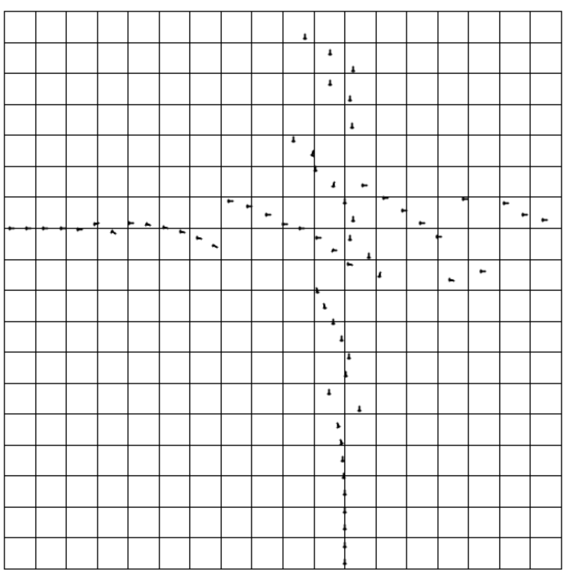

Fig. 3. Perpendicular Flows With 10 Mile Grid

Figure 3 shows a snapshot of the pattern aircraft assume when employing the satisficing algorithm. Although the algorithm is not preprogrammed to handle this specific scenario, the solution that emerges has the same geometric characteristics as the algorithm published by Dugail *et al.* In particular, both solutions exhibit the formation of waves or rows of aircraft. Again, the performance and emergent behavior of the satisficing approach are promising.

D. Computational Load

An important factor in the real-time implementation of any algorithm is computational load. If an algorithm cannot be implemented in real time, other measures of its performance are not very meaningful. In our simulations, we used a single Pentium IV processor to control all aircraft in all simulations. This reached a maximum loading of 80 aircraft modeled simultaneously. Of the two models, the full model is the more computationally intense. For an n agent system, where each agent has k options, the computational load (for the entire system) can be $O(n \cdot k^n)$. The actual number of computations depends on how many aircraft are within the 50 mile radius, and how many have the chance of causing a conflict with each other. The larger this web of influences grows, the more difficult it becomes to calculate. For this reason, simulations such as the choke point scenario cannot use the full model as a decision support algorithm. This realization was our main motivation for creating the simplified model.

The computational load (for the entire system) of the simplified model is simply $O(n)$. In other words, the algorithm runs in constant time for each aircraft regardless of the number of aircraft. With this algorithm, even scenarios with the complexity of the choke point can easily run in real time, and this makes the simplified model an attractive alternative to investigate. Note that the computational overhead of a truly distributed implementation is further reduced for both schemes because each aircraft runs its own satisficing algorithm locally.

VIII. Conclusions and Future Work

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 air traffic control. Satisficing theory is mathematically sound, robust, and flexible. Solutions based on satisficing theory can exhibit complex behavior, yet be 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 air traffic control.

References

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

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

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

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

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

[6] J. C. Hill, J. K. Archibald, W. C. Stirling, and R. L. Frost, "A multi-agent system architecture for distributed air traffic control," in *AIAA Guidance, Navigation and Control Conference*, San Francisco, CA, August 2005, aIAA-2005-6049.

[7] 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 *AAMAS '05: Proc. of 4th Int. Joint Conf. on Autonomous Agents and Multiagent Systems*. ACM Press, 2005, pp. 1083–1090.

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

[9] J. von Neumann and O. Morgenstern, *The Theory of Games and Economic Behavior*, 2nd ed. Princeton, NJ: Princeton Univ. Press, 1947.

[10] W. C. Stirling, "Social utility functions - part I: theory," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, no. 4, pp. 522–532, November 2005.

[11] W. C. Stirling and R. L. Frost, "Social utility functions - part II: applications," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, no. 4, pp. 533–543, November 2005.

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

[13] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufmann, 1988.

[14] 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," in *4th USA/Europe Air Traffic Management R&D Seminar*, Santa Fe, NM, December 2001.

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

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

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

[18] C. Tomlin, I. Mitchell, and R. Ghosh, "Safety verification of conflict resolution maneuvers," *IEEE Trans. on Intelligent Transportation Systems*, vol. 2, no. 2, pp. 110–120, June 2001.

[19] R. Ghosh and C. Tomlin, "Maneuver design for multiple aircraft conflict resolution," in *Proc. of the American Control Conference*, vol. 1, Chicago, IL, June 2000, pp. 672–676.

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

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

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

[23] Z.-H. Mao, E. Feron, and K. Bilimoria, "Stability and performance of intersecting aircraft flows under decentralized conflict avoidance rules," *IEEE Trans. on Intelligent Transportation Systems*, vol. 2, no. 2, pp. 101–109, June 2001.

[24] S. Resmerita and M. Heymann, "Conflict resolution in multiagent systems," in *Proc. of IEEE Conf. on Decision and Control*, vol. 2, Maui, Hawaii, December 2003, pp. 2537–2542.

[25] S. Resmerita, M. Heymann, and G. Meyer, "A framework for conflict resolution in air traffic management," in *Proc. of IEEE Conf. on Decision and Control*, vol. 2, Maui, Hawaii, December 2003, pp. 2035–2040.

[26] A. Bicchi and L. Pallottino, "On optimal cooperative conflict resolution for air traffic management systems," *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 221–232, December 2000.

[27] L. Pallottino, E. M. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *IEEE Trans. on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 3–11, March 2002.

[28] L. Pallottino, A. Bicchi, and S. Pancanti, "Safety of a decentralized scheme for Free-Flight ATMS using mixed integer linear programming," in *Proc. of the American Control Conference*, Anchorage, AK, May 2002, pp. 742–747.

[29] E. Frazzoli, L. Pallottino, V. Scordio, and A. Bicchi, "Decentralized cooperative conflict resolution for multiple nonholonomic vehicles," in *AIAA Guidance, Navigation and Control Conference*, San Francisco, CA, August 2005.

[30] R. A. Paielli and H. Erzberger, "Conflict probability estimation for free flight," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 3, pp. 588–596, May-June 1997.

[31] M. Prandini, J. Hu, J. Lygeros, and S. Sastry, "A probabilistic approach to aircraft conflict detection," *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 199–220, December 2000.

[32] J. Rong, S. Geng, J. Valasek, and T. R. Ioerger, "Air traffic control negotiation and resolution using an onboard multi-agent system," in *Proceedings of Digital Avionics Systems Conference*, vol. 2. IEEE, 2002, pp. 7B2–1–7B2–12.

[33] 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. of IEEE Conf. on Decision and Control*. San Diego, CA: IEEE, December 1997, pp. 1835–1840.

[34] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, December 2000.