AIAA Guidance, Navigation, and Control Conference and Exhibit
5-8 August 2002, Monterey, California

AIAA 2002-4857

IMPROVED TAXI PREDICTION ALGORITHMS FOR THE SURFACE MANAGEMENT SYSTEM

Chris Brinton*, Jimmy Krozel, Ph.D.†, Brian Capozzi, Ph.D.‡
Metron Aviation, Inc.
Herndon, VA 20170

Stephen Atkins, Ph.D.§
NASA Ames Research Center, MS 210-6
Moffett Field, CA 94035

ABSTRACT

Airport Surface Traffic Management Systems (STMSs) are emerging as a new technology to assist human operators in the management of surface traffic operations. One aspect of such operations is the routing of traffic. This includes the management of arrival traffic that must land and taxi to gates, as well as the management of departure traffic from pushback to take off. In general, arrivals and departures compete for the same taxi and runway resources. In this paper, we describe the Surface Management System (SMS) architecture and key algorithms that help controllers plan and manage arrival and departure traffic. Furthermore, we present results obtained with a mature SMS system designed for major airports. We then describe several algorithms that we have implemented based on the A* Algorithm and concepts from evolutionary computation as potential enhancements to the algorithms currently used in SMS for surface movement modeling. These algorithms have the potential for improving the accuracy of the taxi transit times predicted by SMS. Emphasis is placed on optimal surface routing given the constraints of limited available resources and the necessity of producing conflict-free surface trajectories. A key factor throughout these studies is the use of dynamically varying costs to model resource utilization.

INTRODUCTION

The anticipated future growth of the demand placed on the National Airspace System (NAS) will cause the airport surface, already burdened, to become a major constraint on system capacity and throughput. The most significant aspects of surface activity include the number of available runways and their physical layout, wake vortex separation requirements, runway occupancy time requirements, surface congestion, and gate availability. Conceptual research toward improving airport surface operations includes the queuing model of Pujet[1] used to evaluate preliminary control schemes for alleviating departure traffic congestion. Capacity and efficiency aside, however, as the density of airport demand increases, the danger of runway incursions grows – a trend that has already begun to emerge. For example, a recent FAA report cited that the number of runway incursions had increased by over 50% during a recent four-year period[2]. This highlights the obvious safety concerns related to operations on the surface.

There are four primary functions related to the control of surface traffic: (a) determination of the routings to be used in moving vehicles on the surface, (b) maintenance of separation between vehicles, (c) delivery of routing clearances to the flight deck, and (d) execution of the prescribed taxi routing by the pilot. Key limitations on current surface operations stem from the visual means through which both tower controllers and pilots develop situational awareness. Further, voice communication of clearances is limiting due to the capability of the controller to deliver only a single clearance at a given instant, the potential for miscommunication of the spoken word, and frequency congestion. Research aimed at mitigating some of these effects includes the T-NASA research[3] on alternative cockpit displays of information and moving maps to enable taxiing in low-visibility conditions. It should be noted, however, that despite the fact that control of surface traffic involves only two spatial dimensions, the temporal aspects and overall complexity of the information dimensions can make it difficult for

---

[*] Vice President, ATM R&D

[†] AIAA Associate Fellow, Director, Aviation R&D

[‡] AIAA Member, Systems Analyst

[§] AIAA Member, Aerospace Engineer

humans to organize and process efficiently. Research by Cheng[4] focuses on automated taxi control algorithms that would enable aircraft to autonomously follow time-based surface trajectories.

One means for overcoming these limitations with regard to functions (a) and (b) above, is to design decision-support tools (DSTs) to assist controllers and users of the National Airspace System (NAS) to manage the movement of aircraft on the surface of busy airports. Such DSTs have the potential to improve capacity, efficiency, flexibility, and safety. The Surface Management System (SMS), being developed by NASA Ames Research Center, in cooperation with the FAA, is being developed for just this purpose[5],[6]. SMS, developed to operate in real-time using electronic inputs including flight plan and surface position track data, provides decision support capabilities to improve airport surface movement efficiency and flexibility. Further, SMS supports improvements in the use of airport capacity by providing Air Traffic Control Tower (ATCT) controllers with more complete and precise information about departure and arrival demand, predicted pushback times, and runway utilization. SMS information can be provided to Ramp Tower controllers and Air Traffic Service Providers (ATSPs) in Air Route Traffic Control Centers (ARTCCs) and Terminal Radar Approach Controls (TRACONs) to enhance Collaborative Decision-Making.[7]

This paper extends the ideas that are the foundation for the routing algorithms contained in SMS. The routing algorithms predict the taxi path that the aircraft will follow, for the purpose of predicting the transit time between points of interest on the airport surface. SMS must model surface movements to this level of detail so that its higher-level products, such as predicted takeoff times and advisories based on those predictions, are accurate and useful. The current SMS system includes both network event simulation components for developing models of the evolving surface traffic as well as decision-support capability such as recommendations regarding pushback times, runway assignments and runway configuration change times. Currently, however, the routing algorithms and logic providing de-confliction of trajectories are decoupled. Further, the computational complexity of the current implementation requires the use of lower fidelity surface traffic models to meet the needs for real-time interactive control and advisories.

This research takes the next step by postulating improved algorithms for predicting the taxi paths that aircraft will follow, and the resulting transit times, that are based on complete, de-conflicted surface motion plans for all vehicles. The focus of this paper is similar to previous research by Gotteland et. al.[8] that describes several ground traffic optimization strategies that combine genetic algorithms (GAs) with an A* search to develop taxi routes minimizing time spent between gate and runway. Strategies for generating taxi routes include a "global" method in which the GA effectively chooses between alternative (a priori defined) paths and assigns an optional holding location and time; and a "mixed" method in which the GA assigns a path and priority level for each aircraft and the fitness function is computed by an A* algorithm applied to the priority-sorted aircraft. These GA strategies are compared to a "1-to-n" strategy similar to that described in this paper.

The goal of the research described herein is to provide a coordinated motion plan for all vehicles currently using, and those anticipated to be using, the surface resources (runways, taxiways, gates) over a certain time horizon. These time-based surface trajectories would guarantee separation and optimize a performance metric either related directly to surface operations or incorporating impacts on neighboring NAS domains (e.g. terminal area, other airports, etc.). Our approach is to establish a means to enable metrics such as airport throughput and impedance matching to neighboring NAS domains (both landside and airside) to ultimately be the drivers for determination of allocation of airport resources.

REQUIREMENTS FOR SURFACE TRAFFIC MANAGEMENT SYSTEMS

The airport surface can be thought of as a set of resources (runway, taxiways, gates). The users of the airport movement area (aircraft, service vehicles, etc.) can be thought of as consumers that desire to utilize these resources. Each airport has different inherent characteristics – for example, motion through the taxiway system is constrained via its spatial layout. Further, the airport resource availability is a function of time, and resource utilization contains significant uncertainty in the present day operational environment.

Due to these considerations, planning the utilization of airport resources has key information dependencies on "both sides" of the surface – both from the "upstream" en-route and terminal environments for arrivals (possibly coordinating with Center-TRACON Automation System (CTAS) tools such as TMA[9] and FAST[10]) as well as the airlines (departure/pushback schedules) and "downstream" constraints (e.g. en-route spacing programs for merging, fix balancing, etc.). The SMS continuously receives dynamic inputs, which represent the dynamic situation on the airport surface, as well as updated predictions of future flight pushbacks and arrivals. SMS information requirements include:

- Arrival Demand Data – forecasted either via the Enhanced Traffic Management System (ETMS) or through CTAS tools such as TMA
- Surface Surveillance Data
- Terminal Airborne Surveillance Data – enables accurate predictions of inbound demand and modeling of runway usage considering required separation between arrivals, departures, and crossing traffic.
- Airline CDM Interface. CDM with airline organizations is an important element of STMS functionality – particularly with regard to departure demand and gate utilization. Airlines would provide last-minute flight and gate changes and the STMS would provide an electronic output of data to airlines for their use.

Based on these inputs, the routing and de-confliction algorithms approximate the taxi routings and resource utilization (gates/runways) that are most likely to be realized by tower controllers. Further, by postulating various fast-time "what-if" scenarios, the STMS can display the potential value of different courses of action with regard to resource usage and/or system efficiency. For this purpose, the STMS needs to provide a number of different "views" into its modeled operations. The current research SMS Map Display (Figure 1), for example, depicts the layout of the airport surface and positions of active flights. Timeline and load graph displays for various resources (handoff spots, key intersections, and runways) are also part of the SMS displays.

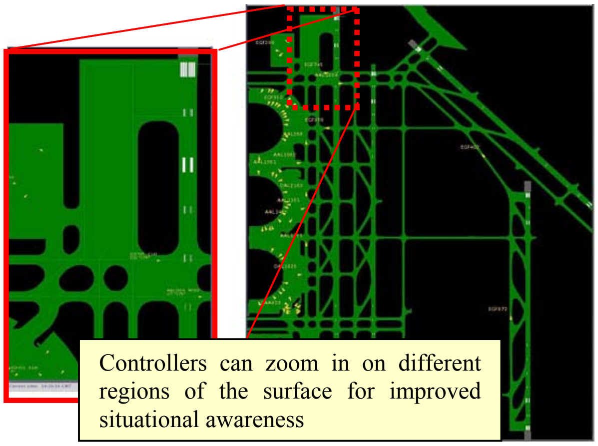

Controllers can zoom in on different regions of the surface for improved situational awareness

Figure 1 Depiction of the surface layout and aircraft positions at DFW with a zoom capability.

REVIEW OF CURRENT SMS

The current SMS consists of three primary components: Surface Trajectory Synthesis, Surface Flow Modeling, and Flow Management Decision Support. We focus here on the algorithms defining the Trajectory Synthesis and Flow Modeling capability as these relate directly to surface traffic planning.

Algorithm Description

The surface is modeled as a set of intersections and links. For this purpose, runways and gates are modeled as intersections. Initially, SMS computes and stores the shortest path taxi routes from every intersection on the airport surface to each runway and to each parking gate (ramp links are taken to be twice as long as non-ramp links to model slower speeds in the ramp area). At each trajectory prediction cycle (currently every 10 seconds), Trajectory Synthesis uses initial conditions defined by updated surface position data and these shortest paths to compute arrival times at all intersections from each flight's initial position to its assigned runway or gate. Taxi speed along these shortest paths is currently modeled using independent constant parameters for arrivals and departures (future work involves definition of acceleration and deceleration parameterizations for higher fidelity modeling). The times computed in this fashion represent each vehicle's earliest possible arrival time at each intersection. Note, however, that this set of trajectories may well contain numerous conflicts. To resolve such conflicts, the Flow Modeling algorithm utilizes a network event simulation to model arrivals at intersections and to determine the amount of time that flights must hold at various locations on the surface to maintain separation requirements and insure safe separation from other flights (e.g. at intersection crossings, wake vortex separation, no head-to-head operations, etc.).

There are actually three different SMS surface Flow Modeling algorithms that have been developed. Each of the three algorithms has different levels of modeling accuracy and different levels of computational requirements. As expected, the more accurate modeling algorithms have significantly higher computational requirements. The three different surface Flow Modeling algorithms are called, in decreasing order of complexity: the SMS Model, the Basic Model, and the Simple Model. The SMS Model is the highest fidelity model and explicitly represents many of the constraints associated with the airport surface flow. The Basic Model only explicitly represents those constraints that are essential for accurate modeling of the times of airport surface events (crossing spots, taking off, landing, arriving at parking gate). In particular, the Basic Model ignores certain critical constraints that have the highest computational overhead – namely those related to prohibiting opposite direction flights and overtakes. The Simple Model further simplifies the airport surface flow modeling detail to capture only overall flight delays and predicted OUT, OFF, ON, IN times – resulting in much faster execution time. Next we describe some calibration results obtained using the Simple Model on adaptations of four airports.

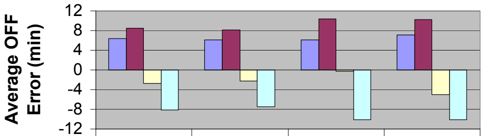
Calibration Study Results

The four airports selected for adaptation (ATL, DFW, JFK and MEM) represent different airport categories. Each adaptation is a model of the airport layout in terms of runways, configurations, taxiways, intersections, gates, links, aprons, spots and spot areas. Jeppesen airport diagrams and detailed measurements were used in creating the adaptations. These newly adapted models were first validated to insure accurate modeling of the airport surface. Each adaptation was independently calibrated by varying the (1) arrival and departure taxi speeds, and (2) inter-departure separation times in order to minimize SMS predicted OFF and IN errors as compared against Bureau of Transportation Statistics (BTS) "Out, Off, On, In" (OOOI) data. Input data files for the calibration process were created for three days in January 2002 (29, 30, 31) using data from the following sources: BTS OOOI data, ETMS flight data and fix list (used to assign departure runways), Flight Information Display System (FIDS) gate assignments, and Aviation Safety Performance Metric (ASPM) runway configuration schedules.

Several quantitative measures of effectiveness (MOEs) were compared including

- OFF Error (modeled OFF – Actual OFF) versus time of day for all runways
- IN Error (modeled IN – Actual IN) versus time of day for all runways.

SMS was used to predict the modeled OFF times given OOOI "OUT" data (related to pushback, door closure, etc. depending on the airline) as well as the modeled IN times for arrivals given OOOI "ON" data. The quantitative MOEs were computed for each airport adaptation and for each day. Overall, the validation results indicate that SMS tends to under predict the OFF and IN times for each airport as shown in Figure 2. Note that the relative errors in OFF time are significantly larger than those for the IN time predictions. This is consistent with the fact that congestion and complexity in the departure process is generally higher than for arrivals. Results from previous simulation exercises at NASA Ames' FFC facility[11] indicate that airport surface choke points and requirements to cross runways can have a significant effect on departure flow rates and airport performance. The fact that the Simple Model does not explicitly model these aspects of the surface operation caused inaccuracies in the SMS predictions. Next, we explore algorithmic approaches to explicitly represent these separation constraints while aiming to lessen the computational burden of the current SMS Model's approach to space/time de-confliction.



	ATL	DFW	JFK	MEM
■ Calibrated Absolute	6.36	6.09	6.12	7.18
■ SIM #2 Absolute	8.49	8.15	10.43	10.26
□ Calibrated Signed	-2.78	-2.21	-0.23	-5.01
□ SIM #2 Signed	-8.17	-7.48	-10.09	-10.11

Figure 2 Calibration Results for SMS using the Simple Model, with taxi speed and inter-departure/arrival spacing selected to minimize OFF and IN time prediction error relative to OOOI data at four airport adaptations.

ALGORITHMIC APPROACHES TO TAXI PREDICTION

The taxi prediction problem involves creating a surface trajectory (time-based) for each aircraft. A key feature is that these surface trajectories must be conflict-free and maintain a specified level of separation between each aircraft. Ideally, these motion plans are formulated so as to, when considered en masse, optimize the surface operation in some manner, which is assumed to be on average the best prediction of how controllers will route aircraft on the surface. Objectives may include minimizing asset utilization times (e.g. runway occupancy), maximizing airport throughput (in terms of operations per time period), or maximally distributing delays in an equitable manner. Note that the nature of the performance goals for the surface can be critically linked to both upstream and downstream NAS constraints. For example, it does no good to be able to release more departures than the terminal area controllers can handle. By the same token, pushbacks from the gate are inherently limited by the time required for passengers to board the aircraft. This implies the need for impedance matching between neighboring NAS domains so as to not create excessive surplus or deficit in the capacity/demand balance.

Space/Time Network Search

The set of intersections and links define a graph structure. The motion of aircraft through the surface resources can be captured through a space/time representation in which the occupancy of the nodes and links of the graph changes in time. One approach to developing a motion plan is to utilize a generalization of Dijkstra's algorithm[12] or A*[13] to search a graph in which time is implicitly encoded in the cost traversing the edges connecting the nodes of the graph. The search proceeds in a wave-like fashion, outward from a given location, establishing the local "best direction to head" from each node expanded relative to an individual vehicle's goal location (either gate or runway). At each node, the direction of desired motion is taken to be along the link that has the lowest cost when viewed over some "decision" time interval. The aggregation of these local "best directions" forms an Optimal Path Map (OPM) for aircraft.

The cost of traversing a taxiway is taken to have a default value equal to the time required for a vehicle to travel along the length of the taxiway at a given speed. This cost is modeled to increase proportional to the density of traffic distributed along the taxiway length during a given period of time. Costs along a given link can be direction-dependent in the sense that traffic in the opposite direction to that being considered during the search is modeled as having a very large (e.g. infinite) cost in order to avoid head-to-head traffic conflicts. The nature of this time-varying cost is indicated in Figure 3.

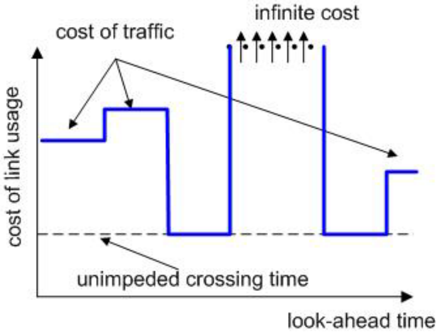


Figure 3 Edge costs as a function of time.

Based on this figure, it appears as if there are two crossing times that allow for unimpeded travel along the link. In making such a decision, however, it is important to factor in the time related to waiting. Such would also be the case, for example, if a vehicle needs (or chooses) to come to a stop at a given intersection to allow for spacing or crossing of another aircraft.

As a first approximation, this additional cost of delay can be modeled as a linear function of time. Its impact on the nature of the time-varying costs for traversing a given arc is illustrated in Figure 4. Here, as indicated, the first crossing opportunity leads to the accumulation of less cost.

Given this formulation, the traffic management problem becomes an issue of establishing time-varying arc costs and using them to evaluate the optimal decisions for each vehicle at each intersection. Issues include the planning horizon and the dependency of each vehicle on the decisions of other vehicles. The next several sections describe different approaches toward the solution of this problem – in particular the issue of developing "global" as opposed to local optimal solutions.

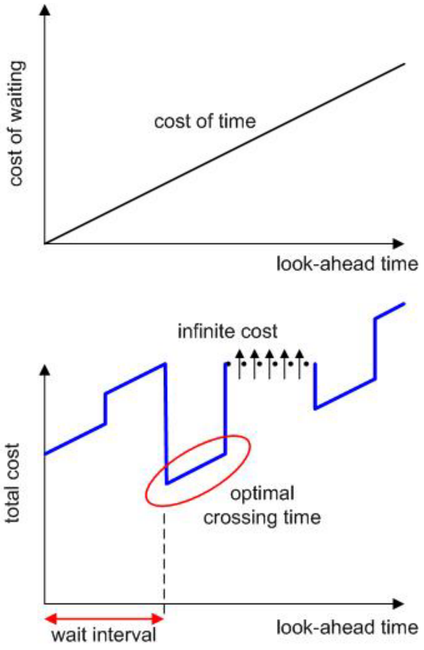


Figure 4 Optimal crossing time for an edge.

Event-Based Decision-Making

Our first approach involves processing vehicles in a first come, first serve sense – allowing the first processed vehicle to make its local best decision. We then process the next vehicle to arrive at an intersection (might be the same one). This decision is made assuming that all other vehicles have chosen which routes to take over the time period of interest. This process is described graphically in the timelines of Figure 5. Notice that vehicle A chooses a routing that is not necessarily its shortest time path to its next intersection. Such decisions hinge on the computation of the time-varying occupancy for each link as described in the previous section. This link occupancy changes as each new vehicle is processed forward in time. At each decision point, we thus choose the link to follow that has the lowest cost –

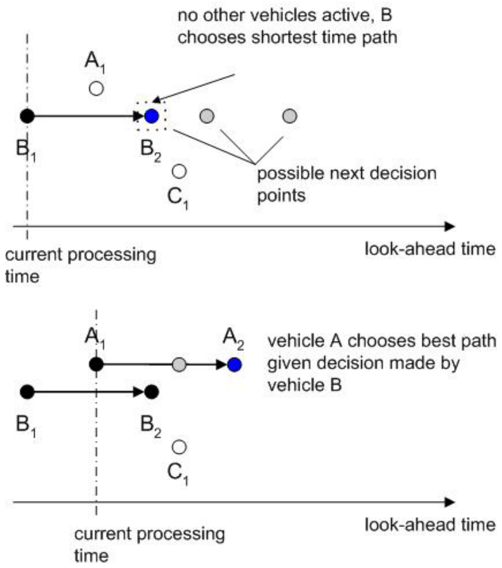


Figure 5 Timeline used to determine the order of decision-making in the event-based scheme.

whether it is the shortest (fastest) *with* congestion or a longer route that is currently unused by another vehicles. Future link occupancy (whether exact or only an estimate) provides the opportunity for each vehicle to look forward in time to see when congestion on a desired link might decrease and thus choose to hold at the current intersection. In making such a choice, however, the aircraft would increase the cost to all vehicles wanting to use this node and/or link.

Algorithm

1. Find next event time over all vehicles.
2. Search over links – Evaluate the cost of traversing each link given decisions of all other vehicles made prior to this event time.
3. Select the link and/or waiting time yielding the lowest cost – updating the link occupancy function over the time period required for this vehicle to traverse the selected link.
4. Iterate to Step 1 until all vehicles reach their goal locations.

In this approach, we are choosing the best direction to head given only information related to the current and/or projected usage of taxiways over some planning horizon of interest.

Again, this usage is due only to *previous* decisions made with regard to other aircraft. We do not attempt to model future intent of the vehicles at this time. As such, we are making temporally local optimal decisions – akin to those which would be made in a decentralized, distributed system in which each vehicle acts autonomously in its own best interest given only local sensing.

Note that the solution obtained in this manner is unlikely to be optimal in a group or "system" sense – although this remains to be formally established. It seems plausible that a decision made early on in a given taxi plan might lead to rather disastrous circumstances downstream in terms of congestion or accumulation of delay – possibly not even including the vehicle which made the critical routing decision in the first place. Ideally, one could encode some measure of group performance into the cost functions used for local decision making. Such measures could be related to, for example, the distribution of delays over each of the vehicles over a given planning horizon.

Need for Replanning

Nominally, the cost of traversing an edge (e.g. taxiway link) is determined by the time required to cross the edge, the direction of travel used in doing so, and potential separation conflicts. These nominal costs, however, may change due to events that happen on the surface of the airport. Thus, replanning and the event-based simulation of current and future surface motion are inherently intertwined. As mentioned previously, an aircraft may stop on an edge and cause a delay, thus causing an increase in the time to cross the edge for all aircraft behind it. The weather may also change, and average transit speeds across edges might be reduced. Thus, it is essential that the routing algorithms be capable of handling situations in which the arc costs change dynamically with time. In this fashion, the corresponding OPM used for routing decisions can be adjusted dynamically to maintain optimal paths in light of a changing surface environment. Not only can the routing algorithm respond to sensed changes in the current environment, but also it can be used to handle uncertainties in sensed and predicted information to provide optimal paths given probabilistic future state. Such ideas are an extension of Linden and Glicksman[14], which describes an algorithm for autonomous vehicles incorporating contingency plans to account for uncertainties. In this work, the planner finds preferred routes taking into account the potential cost of detours (e.g. if "choke" regions are found to be blocked). A similar scheme could be used for airports whose taxiway layout lends itself to bottlenecks.

In the event that a graph edge changes in cost (up or down), the optimal directions to proceed within the graph will possibly change. In either case, the entire graph does not necessarily have to be completely re-searched, although in the worst case this may have to happen. In general, two cases need to be considered with regard to cost changes: (a) costs that change with no a priori knowledge of the change, and (b) edge costs that are expected to change as a known (or predicted) function of time. When the cost of an edge changes, it can change higher or lower. The re-planning of the graph differs for these two cases. However, either Dijkstra's algorithm or the A* algorithm can be implemented for the re-plan. Stentz[15] presents one such modification to enable focused replanning in the vicinity of sensed changes in the cost of using different links. We do not address such replanning issues in the results obtained in this paper.

Bander and White[16] present an adaptive A* algorithm in which the search is guided by: a generalization of the heuristic function; a set of pre-determined optimal paths; and a set of desirable paths that may or may not be optimal. This work investigates mechanisms for incorporation of knowledge from a variety of sources (some possibly human) to guide the numeric search process as well as the utilization of previously computed optimal paths for accelerating the determination of new optimal paths in similar (but not identical) scenarios. Their work is highly relevant to our problem – as feedback from current tower controllers could be incorporated into the planning framework in this fashion.

Co-Evolution of Taxi Routings

Recall that the potential problem described in the aforementioned space/time graph search technique for developing coordinated taxi routings was the fact that local decisions made for the good of an individual may in fact lead to a set of taxi routes which are not optimal in an overall system or group sense. This problem stems from the fact that local decisions are made without knowledge of the future congestion and delay possibilities.

As an alternative to developing the motion plan for each vehicle on a "step-by-step" basis, we instead investigate the potential for simultaneously developing complete paths connecting the start and goal locations for each vehicle. We utilize the collaborative co-evolution architecture described by Potter[17] for this purpose. The idea behind this approach is to represent the taxi route for each individual vehicle as a separate population of genetic "strings" which are allowed to evolve independently, but which are evaluated in the context of "representatives" selected from each of the other populations (see Figure 6). This allows constraints related to the simultaneous action of all participants to be evaluated and included in the performance function. Through this process, each population tends to create its own "niche" – in this case, its taxi routing – that best meets the performance goals for the overall group. A key advantage of this evolution-based approach is that we can evaluate complete taxi-routings for each vehicle simultaneously.

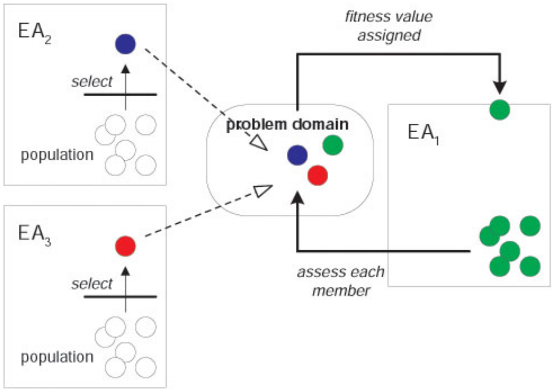


Figure 6 Collaborative Co-Evolution of Species applied to taxi routing

The evolutionary process is modeled to consist of an initialization and then a series of "generations" consisting of evaluation, selection, and mutation. The limited space for solutions emulates the competition for survival. Selection of survivors is done using a probabilistic tournament selection process to allow lesser performing individuals to survive with some non-zero probability. This has the effect of allowing greater portions of the search space to be explored rather than quickly narrowing the search to only highly promising regions.

For this initial study, we represent the taxi route for each vehicle as a string of integers depicting the sequence of nodes that are to be visited. The sequence for each vehicle begins with the initial location and ends with the vehicle's goal location (either gate/handoff spot or runway). The taxi speed of each vehicle is taken to be constant at this point, although this could be included as a parameter. For each candidate routing (e.g. member of the population), the vehicle's time of arrival at each intersection can be computed given the spatial layout of the taxiway network. We implicitly represent the connectivity of the taxiway network in each sequence so as to only evolve physically realizable routings.

For this initial study, we choose a population of size ten members (e.g. candidate routings) for each vehicle. We do not a priori limit the size of the sequences representing each routing but rather allow these to be of variable length, possibly including circuits or loops. The mutation scheme used for these initial experiments is to randomly replace selected nodes in each sequence with a different "valid" (e.g. connected) neighbor of the previous node. The choice of which neighbor is made with uniform probability over the set of all connected nodes. This process is depicted graphically in Figure 7, which shows the impact of a possible mutation on a candidate path.

We compute the cost for each member of a given vehicle's population by first selecting the best performing individual from each of the other populations (for the first iteration, these representatives are selected at random). The sequence of nodes comprising a given surface trajectory is evaluated in much the same manner as that described in the graph search scheme – with the advantage of knowing all "future" decisions.

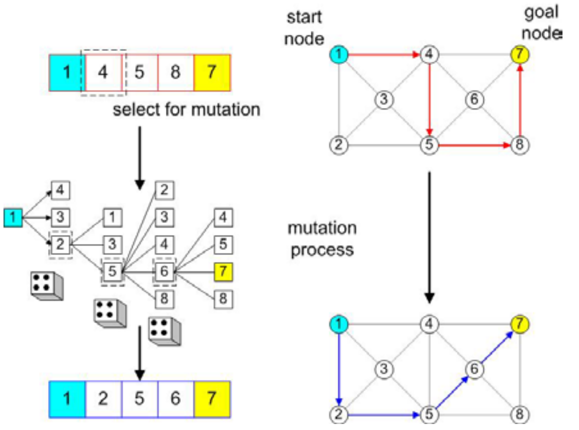


Figure 7 Population representation and mutation scheme used for evolution of taxi-routings.

The cost function consists of evaluating the link occupancy and intersection crossing separations as well as the distance and/or time accrued by each vehicle along a given taxi routing – given the decisions made for "all time" by the other vehicles. In addition, group performance metrics are computed, allowing, for example, the evolutionary search to find solutions which tend to minimize the overall time of completion for all vehicles. Other group cost functions could be related to the maximum delay imposed on any particular vehicle. In this manner, the evolutionary search can develop solutions that approach "global" group-optimal or system-optimal taxi routings.

SIMULATION RESULTS

In this section we describe results obtained by applying both the Event-Based A* and the GA Co-Evolution strategies to a simple 6-vehicle problem. Our "airport" consists of a simple spatial layout of 8 nodes arranged as indicated in Figure 8.

Departures

Flt	Spot	t_D	Rwy
v1	1	0	7
v2	2	1	8
v3	1	2	7

Arrivals

Flt	Rwy	t_A	Spot
v4	8	0	1
v5	7	1	2
v6	8	2	1

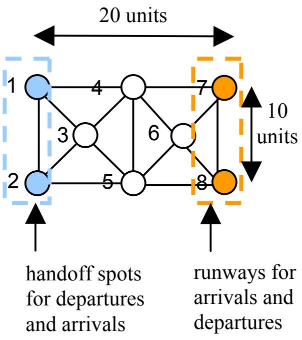


Figure 8 Simple airport layout used for initial investigations comparing GA and A* algorithms.

The time at which each departure reaches the handoff spot and at which each arrival reaches the runway exit point are defined to be t_D and t_A, respectively. The total distance between Spot 1 and Runway 8 is 20 units, while the distance between Spots and distance between Runway exits is 10 units. For purposes of this study, each vehicle was assigned a constant taxi speed of 2 units/time. Thus, the minimum time needed to get from Spot 2 to Runway entrance 8 is 10 time units.

For the Event-Based A* search, we utilized a link occupancy cost (traffic which does not violate the separation minimum of 2 distance units) equal to the product of the unimpeded link travel time at a given speed and the number of vehicles safely occupying the link. This cost is developed to model the fact that the more vehicles using a given link, the higher the likelihood that one of the vehicles will be required to hold downstream. Additionally, the cost of waiting is taken to be the time value (in units of time) desired to wait (for example, for congestion to clear and entry onto a link to become safe relative to separation minimums).

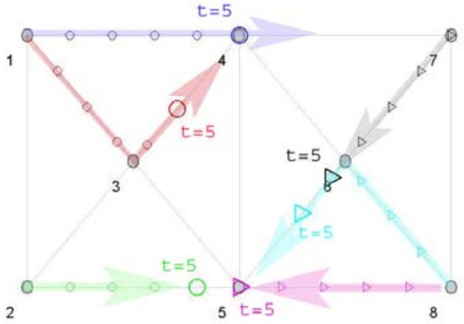
For the Co-Evolution-Based GA strategy, the base cost used for judging performance was a summation over all vehicles' time to reach their respective goal locations. The objective of the GA search was thus to develop complementary routings which tend to minimize a measure of the group time to completion. In addition, we model constraints using a penalty formulation in which excessive cost is assigned to a given vehicle if its routing, when considered in the context of the other representatives' routings, violates separation criteria. For this purpose, we utilized a more conservative link occupancy criteria in which we assigned a rather large penalty to the selection of any link already containing traffic. Thus, the solutions developed by the GA in its present form tend to produce routings with minimal sharing of resources, even in cases where such sharing would not violate separation requirements. We are currently working to incorporate the same node/link occupancy costs used in the A* search into the GA formulation.

The "IN" times and "OFF" times for each of the six vehicles for both the Event-Based A* and GA strategy are compared in Table 1.

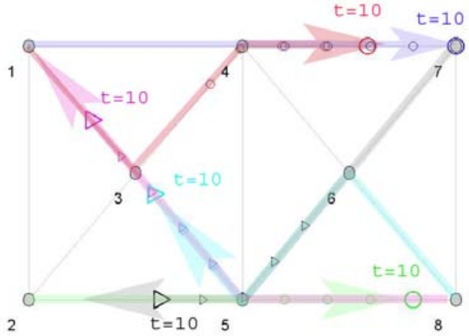
Table 1 Comparison of time to reach goal for 6-vehicle problem using Event-Based A* and GA.

	Vehicle Index (time to reach goal)					
Algorithm	1	2	3	4	5	6
Event A*	10.0	11.0	12.1	14.1	13.1	12.1
GA	12.1	11.0	17.1	15.0	13.1	14.1

The corresponding routings are summarized in Figure 9 and Figure 10, respectively. In each figure, we show a "snapshot" of the taxi routings after five and ten time units have expired. The colors of the routes are consistent with that of the symbols used for each vehicle. Departures and arrivals are shown using circles and triangles, respectively. Larger circles denote the snapshot time. Arrows have been added to aid in discerning the direction of the vehicles and their relative order in terms of utilization of different taxiway segments. In comparing the results of in terms of the time required to reach the goal locations, we find that the Event-Based A* performs better than the Co-Evolution strategy on this simple problem.

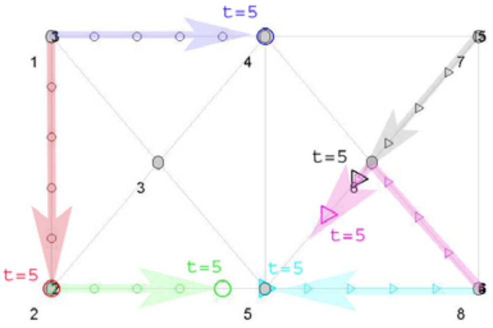


(a) Local optimal routing at time t = 5 units.

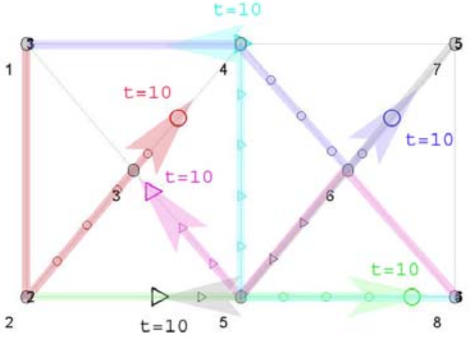


(b) Local optimal routing at time t = 10 units.

Figure 9 Snapshots of the temporally local optimal solution obtained using the event-based A* search.



(a) Co-evolution routing at time t = 5 units



(b) Co-evolution routing at time t = 10 units

Figure 10 Resulting taxi routing for the 6-vehicle problem using Co-Evolution GA.

A total cumulative time of 72.4 time units was found for the A* solution with a corresponding GA time cost of 82.4 time units. Two of the vehicles (2 and 5) were assigned to identical taxi routes by both algorithms. Our current belief is that the conservative link occupancy penalty applied in the GA formulation (via the performance function) kept it from exploring certain regions of the solution space. Further research will be done to assess the extent to which this difference in cost function between the two algorithms influenced the solution.

Note that, for this particular problem, the locally optimal solution determined by the Event-Based A* algorithm contained no queuing or holding at an intersection (for crossing) for any of the vehicles. Further, at this time, the GA population representation does not contain a parameter to establish wait time at any of the nodes – thus, its solution space is effectively reduced in size relative to that of the Event-Based A*. This limitation will be removed in subsequent investigations.

The taxi routings obtained using the Co-Evolution strategy were taken to be the best candidate routing "representative" of each of the populations – where the GA search was artificially terminated after 50 generations. The Co-Evolution strategy was run several times with different mutation parameters (influencing the number of mutations per individual). Essentially the results of these various runs showed the capability of trading off trip durations between the various vehicles but the overall group completion time remained nearly constant. This again points to the fact that it is the penalty formulation regarding link/node occupancy that may be influencing the nature of the search space explored by the GA. It remains to be seen in future research as to whether or not the Co-Evolution formulation has an advantage in terms of overall performance of the vehicles on more complex problems.

SUMMARY AND CONCLUSIONS

This paper has described several algorithmic approaches to developing complete, de-conflicted motion plans for surface traffic. In particular, we implemented an event-based A* search which develops motion plans on an incremental basis, making local decisions on the basis of previous routing decisions made (or assigned to) other vehicles. Further, we explored the use of a Co-Evolution strategy to simultaneously evolve different species (corresponding to vehicles) of routing plans that are evaluated in a group context or problem domain. This latter approach is anticipated to have potential for optimizing system-wide performance objectives as opposed to local-decisions based on limited information.

Our intention in the development of these algorithmic approaches is to better model the flow of traffic on the airport surface under the guidance of tower controllers so as to more accurately predict times of arrival at different points on the airport surface (e.g. reaching the gate or ready for departure at the runway). In essence, we are trying to algorithmically capture the factors that influence the decision-making process of the tower controllers so as to better predict their actions and thus better model the resulting traffic flow.

We demonstrated each of these algorithms on a simple planning problem involving the simultaneous routing of three arrivals and three departures on a mock symmetric airport layout. It was found that the Event-Based A* outperformed the Co-Evolution strategy on this simple problem in terms of cumulative time of completion over the set of all vehicles. Granted, given the same initial conditions, we do not have a "controller generated" solution to compare with – so we can not necessarily argue which algorithm modeled the arrival times more accurately in that respect. Future work will involve application of these algorithms to increasingly realistic demand levels so as to facilitate such comparisons.

A key issue identified through this exploratory research, particularly for the Event-Based A*, is the planning horizon – and the degree to which combinations of local decisions may not be optimal in terms of potential downstream (in space and time) problems that can occur. Thus, we plan to further investigate the intertwining of the event-based simulation and the establishment and update of the time-varying link costs. Features we will investigate include multiple decision step planning horizons and iterative decision-making. For example, should the traverse time for vehicle A cross the next event time for vehicle B, should vehicle B be allowed to make a decision first? One option is to stochastically sample the input space and run a set of potential future scenarios, gather statistics and use the likely values to determine the average values/spread of costs on each link to make decisions. This could be done using the "local" N-best choices for each vehicle. Of course, this could be quite combinatorically expensive depending on the number of vehicles and the time horizon considered. One possible approach toward this end is to apply the fictitious game paradigm proposed by Smith[18].

Another key area of concern is the robustness of solutions developed through either of these approaches to uncertainty in the input data driving the planning process. Future work will aim to minimize, to the extent possible, the fragility of surface plans to small perturbations in input conditions. Obviously it is desirable to minimize the need to completely replan the entire surface operation every time two flights get swapped or a gate changes. We will begin to incorporate a probabilistic nature to both the arrival/departure times as well as other resources (gate assignment, availability) to search for solutions that are more robust to uncertainty.

Related to the previous issue is the estimate of each vehicle's taxi speed and the ability of (a) controllers to deliver and (b) pilots to follow consistent motion plans. Thus, the planner must adapt not only to changes in external information related to resource demand, but must also account for failure of the algorithm to correctly predict the taxi paths used by the controllers or executed by the pilots. SMS requires this detailed planning of taxi routes only so that its higher-level products, such as predicted takeoff times and advisories based on those predictions, are accurate and useful. Cheng[4], on the other hand, proposed empowering the aircraft themselves to autonomously navigate the reference trajectories output by the planner. A less extreme solution, put forward by Foyle et. al.[3], is to provide an alternative display or communication of clearances to the controllers and pilots to facilitate their execution (e.g. the T-NASA moving map and heads-up display symbology).

ACKNOWLEDGEMENTS

This research was funded by the Advanced Air Transportation Technologies (AATT) Project at NASA Ames Research Center under Contract Task Order 5 of the Air Traffic Management System Development and Integration (ATMSDI) contract. The authors would like to thank Ted Carniol for providing the SMS validation results and commentary presented in this paper.

REFERENCES

[1] Pujet, N., Delcaire, B., and Feron, E., "Input-Output Modeling and Control of the Departure Process of Congested Airports", *Air Traffic Control Quarterly*, Vol. 8(1), pp. 1- 32, 2000.

[2] FAA, *FAA Runway Safety Report 1997-2000*, FAA Office of Runway Safety, Washington, DC, June, 2001.

[3] Foyle, D., Andre, A., McCann, R., Wenzel, E., D. Begault, and Battiste, V., "Taxiway Navigation and Situation Awareness (T-NASA) System: Problem, design philosophy and description of an integrated display suite for low-visibility airport surface operations", *SAE Transactions: Journal of Aerospace*, Vol. 105, pp. 1411-1418, 1996.

[4] Cheng, V., Sharma, V., and Foyle, D., "A Study of Aircraft Taxi Performance for Enhancing Surface Traffic Control", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 2, No. 2, June 2001.

[5] Atkins, S., and Brinton, C., "Concept Description and Development Plan for the Surface Management System," *Journal of Air Traffic Control*, 2002.

[6] Welch, J., Bussolari, S., and Atkins, S., "Using Surface Surveillance to Help Reduce Taxi Delays," *AIAA Guidance, Navigation, and Control Conference*, Montreal, Canada, August, 2001.

[7] Atkins, S. and Hall, W., "A Case for Integrating the CTAS Traffic Management Advisor and the Surface Management System," *AIAA Guidance, Navigation, and Control Conference*, Denver, CO, August, 2000.

[8] Gotteland, J-B, Durand, N., Alliot, J-M, and Page, E., "Aircraft Ground Traffic Optimization," *4th USA/Europe Air Traffic Management Seminar*, Santa Fe, NM, Dec., 2001.

[9] Farley, T., Foster, J., Hoang, T., Lee, K.K., "A Time-Based Approach to Metering Arrival Traffic to Philadelphia", *First AIAA Aircraft Technology, Integration, and Operations Forum*, Los Angeles, California, October, 2001.

[10] Robinson III, J.E. and Isaacson, D. "A Concurrent Sequencing, and Deconfliction Algorithm for Terminal Area Air Traffic Control", *AIAA Guidance, Navigation, and Control Conference*, Denver, CO, August, 2000.

[11] Brinton, C., Carniol, T., Butler, T., Duley, J., and Hitt II, J.M., *CTO-05 Surface Management System Phase I Report*, February 2002.

[12] Dijkstra, E.W., "A Note on Two Problems in Connection with Graph Theory," *Numerische Mathematik*, Vol. 1, pp. 269-271, 1959.

[13] Nilsson, N.J., Principles of Artificial Intelligence, Tioga Pub. Co., Palo Alto, CA, 1980.

[14] Linden, T., and Glicksman, J., "Contingency Planning for an Autonomous Land Vehicle," *Proc. of the Int'l. Joint Conf. on Artificial Intelligence*, Milan, Italy, pp. 1047-1054, 1987.

[15] Stentz, A., "The Focussed D* Algorithm for Real-Time Replanning", *Proc. of the Fourteenth Int'l Joint Conference on Artificial Intelligence*, Montreal, Quebec, Canada, August, 1995.

[16] Bander, J., and White, C., "A Heuristic Search Algorithm for Path Determination with Learning", *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 28, pp. 131-134, January, 1998.

[17] Potter, M. and De Jong, K., "Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents", *Evolutionary Computation*, 8(1), pp.1-29, MIT Press, 2000.

[18] Smith, R., Garcia A., and Reaume, D., "Fictitious Play for Finding System Optimal Routings in Dynamic Traffic Networks", *Transportation Research B*, Vol. 34, pp 147-156, 2000.