Local conflict resolution for automated taxi operations

Sebastian Bode, Thomas Feuerle, Peter Hecker
Institute of Flight Guidance
Technische Universitaet Braunschweig
Hermann-Blenk-Str. 27, 38108 Braunschweig, Germany
+49 (0) 531 391 98 62, seb.bode@tu-bs.de

ABSTRACT
This paper describes two techniques to achieve conflict free automated taxi processes which have been implemented by an airport traffic simulator. The first technique is based on the transitive reduction of a given routing graph, automatically creating one-way taxiway segments that allow the application of local conflict resolution strategies, preventing aircraft from getting stuck within deadlock situations. It allows an automated assignment of reasonable taxi routes to flights without the need of further manual optimizations of the search graph. The second strategy presents a dynamic method of determining temporarily shared taxiway segments during movement, which are used to generate appropriate commands to guarantee safe and efficient taxi operations. Both are then compared with planning and scheduling driven strategies and results from preliminary simulations are given to emphasize the value of the presented techniques.

Keywords
Automation, surface operations, conflict prediction, conflict resolution, taxiway routing, taxiway scheduling, simulation

CHALLENGES OF AUTOMATED TAXI OPERATIONS
The taxi process of flights is a major part of present and future airport surface operations, which encompasses an aircraft’s movement from a parking position to an entry point of the departure runway or vice versa from a runway exit to a parking position. Fast-time air traffic simulation as well as future Air Traffic Control (ATC) procedures and systems, like Trajectory Based Surface Operations (TBSO) [1], the Advanced Surface Movement Guidance and Control System (A-SMGCS) [2] and others [3] require a partly or fully automated taxi phase. In nowadays taxi operations a controller assigns a route to a flight, starts the taxi process by issuing a taxi clearance and clears successive segments of the route for the flight. The pilots execute the movement following the given route and obey to further instructions. This procedure leaves the controller in charge of ensuring a safe and preferably efficient taxi phase. During periods of high traffic density the workload might exceed a controller’s capacity resulting in less efficient or less safe taxi operations. Another problem is the dependency to bad weather conditions because most of the surveillance is based on visual observation of the maneuvering area. During low visibility conditions special procedures are conducted to maintain safety while dramatically reducing airport capacity [4]. Several technologies like Automatic Dependent Surveillance Broadcast (ADS-B) [5] were introduced to support controllers in their surveillance task and to increase the situational awareness concerning aerodrome traffic. Newer developments like A-SMGCS or the Surface Operation Automation Research (SOAR) project [3] tend to improve efficiency and safety of surface operations through an increased degree of automation, supporting the decision making process of controllers. Both, simulation and operational systems need to implement three functions in order to realize automated taxi operations: Routing, control and movement execution. As described above, nowadays ATC procedures place the responsibility for routing and traffic control on controllers, while pilots are in charge of the movement execution process. Routing is the task of finding a path from any point on the aerodrome surface to another. Control is the issuing of instructions to manage the movement execution process in order to guarantee safe and conflict free usage of the maneuvering area. The routing problem is well known and many solutions exist, which are in most cases based on a graph representation of the airport’s taxiway layout and the application of some graph search algorithms (e.g. Depth-First-Search, or A* [6]).

Many of these algorithms are not applicable within multi agent environments, where more than one aircraft utilizes the graphs’ edges (taxiways) for movement.

(a)  (b)

Figure 1: Taxiway conflicts. The dashed line in (a) represents a holding instruction. (b) is a deadlock.

Conflicts appear when different agents try to use the same edge or crossings of edges at the same time (Figure 1a). Therefore, the control function is in charge of managing competing demands for graph segments, guaranteeing an exclusive usage for single aircraft at a time and preventing deadlocks, which are conflict situations that can not be
solved (Figure 1b). Again, different solutions exist solving the control problem. One approach is adapted from real controllers, who successively clear taxiway segments according to short term prediction of possible conflicts. This approach tries to solve conflicts locally without additional knowledge about other parts of the whole taxiway structure. Therefore, deadlocks are a major problem and must be avoided in order to prevent aircraft taxiway structure. Therefore, deadlocks are a major concept of a possible controller decision making support line without the need of specifying one-way edges and uses appropriate one-way segments from any taxiway graph. The technique is a static approach of automatically determining through a short term prediction of conflicts. The first tech-nique which tends to achieve efficient and safe taxi operations this information for conflict avoidance. In addition a concept of a possible controller decision making support tool utilizing the shared segment algorithm is presented. Finally both introduced approaches are then compared with planning and scheduling driven strategies and results from preliminary simulations are given to emphasize the value of the presented techniques.

**TAXIWAY GRAPH REPRESENTATION**

This section introduces some graph theory terminology used in later sections to describe the presented conflict resolution strategies. The airport layout is represented as a directed graph $G = (V, E)$ with $V = \{v_1, ..., v_M\}$ being the set of nodes representing taxiway crossings and other hotspots like parking positions, runway exit or entry points and $E = V \times V = \{e_1, ..., e_N\}$ the set of directed edges connecting two nodes $v_{start} \in V$ and $v_{end} \in V$.

The function $s: E \rightarrow E$ returns the sibling $e_2$ of an edge $e_1 = \{v_{start}, v_{end}\} \in E$ with $e_2 = \{v_{end}, v_{start}\} \in E$.

Edge $e_2 = \{v_3, v_4\} \in E$ is called a successor of edge $e_1 = \{v_1, v_2\} \in E$ if $v_2 = v_3$ or $v_2, v_3, v_4 \in V$.

The ordered set $P_{vs, ve}$ of succeeding edges $\{e_1, ..., e_L\}$ with $e_j \in E$ for all $j = 1, ..., L$ and $e_1 = \{v_s, v_e\}$ and $e_L = \{v_e, v_s\}$ is called the path between $v_s$ and $v_e$ with a length of $L$. The set $A = \{a_1, ..., a_p\}$ represents all aircraft using the graph for taxi operations. Let $R = \{R_1, ..., R_P\}$ be the set of all taxi routes where $R_i$ is an optimal path connecting two nodes $v_s$ and $v_e$ for all $i = 1, ..., P$ minimizing a cost function $c: E^{N^{no}} \rightarrow \mathbb{R}^+$. The function $u_{edge}: E \rightarrow A^{N^{no}}$ denotes which aircraft use an edge of the graph. An edge $e_j$ is used by an aircraft $a_i$ if a route $R_i \in R$ exists and $e_j \in R_i$. $I = \{T_{start}, T_{end}\}$ is a time interval with $T_{start} < T_{end}$ being timestamps representing any concrete point in time.

The function $o_{edge}: E \times A \times I \rightarrow \{0, 1\}$ denotes if an aircraft occupies an edge during a given time interval. If $o_{edge}(e_i, a, I) = 1$ and sibling $s(e_i) = e_2$ then $o_{edge}(e_2, a, I) = 1$.

Let $S = \{seg_1, ..., seg_Q\} \subseteq E$ be the set of taxiway segments with $\text{seg}_i$ being the ordered set of edges $\{e_{K_i}, ..., e_{K_i}\}$ with $e_{K_i}$ is successor of $e_i$ or if $e_i \in \text{seg}_j$ and $s(e_i) \in E$ then $s(e_i) \in \text{seg}_j$ for all $j = 1, ..., Q$. A taxiway segment $\text{seg}_o$ is called a one-way segment if following condition is fulfilled: $\forall e_j \in \text{seg}_o \mid s(e_j) \notin E$. The function $o_{edge}: S \times A \times I \rightarrow \{0, 1\}$ denotes if an aircraft occupies a segment during a given time interval.

The airport layout is represented as a graph. It is assumed that taxiways are one-way at any time instant, though

**TAXI SCHEDULING SOLUTION**

The taxi scheduling problem is described as the approach to determine an optimal time for each aircraft on when to start taxiing and when to reach certain intermediate points along a predetermined route. It is a specialization of the general job shop scheduling problem adapted for taxi operations. A solution for the taxi scheduling problem based on a mixed integer linear program (MILP) optimization accounting key safety constraints like separation minima and maximum speed is presented in [7].

This paper presents two local conflict resolution strategies and efficient and safe taxi operations to achieve efficient and safe taxi operations. The first technique which tends to achieve efficient and safe taxi operations this information for conflict avoidance. In addition a concept of a possible controller decision making support tool utilizing the shared segment algorithm is presented. Finally both introduced approaches are then compared with planning and scheduling driven strategies and results from preliminary simulations are given to emphasize the value of the presented techniques.
conflicts can only appear at taxiway crossings or between aircraft moving in the same direction not maintaining a minimum horizontal separation (200m). The absence of bi-directional taxiway segments averts deadlock situations as described earlier. This constraint simplifies the problem and reduces it to finding correct usage intervals of taxiway intersections which are represented by single nodes and need no further accounting of any connectivity information of the used taxiway graph. To determine usage intervals at taxiway intersections a prediction of aircraft trajectories is required incorporating additional complexity like modeling uncertainty aspects within movement execution.

Simulations were conducted with a set of 25 aircraft at Dallas Fort Worth International Airport. The average taxi times resulting from the MILP solution were compared to a first-come-first-serve approach and a significant reduction could be observed.

TIME WINDOW BASED GRAPH SEARCH
Another approach is to extend the routing function by taking the temporary occupancy of edges into account. The cost function used to determine an optimal path between two spots on the movement area does not depend solely on spatial distances but in addition it depends on the time needed to travel between those nodes, including holding times if edges are occupied by other traffic participants. The Time Window Based Dijkstra algorithm with a polynomial complexity presented in [8] is an example of such a strategy. This approach differs from the previous described as it minimizes taxi duration instead of path length. Though, the routing function instead of the control function optimizes the overall taxi duration of aircraft resulting in theory in time optimal taxiway routes.

Additional difficulties are incorporated, e.g. the need to predict the movement process of aircraft in order to determine correct usage intervals for single taxiway segments [9] or the accounting of uncertainty in trajectory prediction. Furthermore, if the predicted and actual trajectories differ too much a replanning of taxi routes is needed which might cascade to other planned routes.

As stated earlier the strategies with the best results in optimizing traffic flow are often based on a decision making that appears not transparent and comprehensive to human controllers. The time window based routing function is a good example of such a case, because the advantage of a time optimal taxi route is hard to be intuitively observed. Instead an increase in complexity of aircraft routing is recognized and might lead to confusion resulting in a decreased level of safety.

TRANSITIVE REDUCTION APPROACH
As stated earlier local conflict resolution strategies suffer from deadlocks. A common approach to avoid deadlocks is the utilization of one-way segments to generate appropriate holding instructions when entering two-way segments depending on their current occupancy state.

Figure 2: Conflict resolution at a one-way (hatched) / two-way (light) segments crossing. A waits for B, which occupies the two-way segment.

Figure 2 depicts a typical situation where aircraft A holds at the end of a one-way (light area) before entering a two-way segment (hatched area) which is currently occupied by B. As soon as B leaves the occupied segment, the control function clears aircraft A to continue along its route. This approach for deadlock prevention requires the appropriate placement of one-way segments, which might have strong impact on the efficiency of taxi operations. This modification of the graph can be done manually but for complex airport layouts this task results in a notable amount of effort without the guarantee of an optimal placement. Therefore, an algorithm automatically determining one-way segments with the goal of minimizing the negative impact through the modification of the original graph has been developed.

The algorithm creates one-way from two-way segments by removing the sibling of an edge under the constraint of maintaining the connectivity of the original graph. The connectivity of any graph is given by its transitive closure. The transitive closure $H(G) = V \times V$ of the graph $G$ is defined as the set of edges $e$ for which a path exists that connects nodes $v_{\text{start}}(e)$ and $v_{\text{end}}(e)$. Figure 3 is an example of the transitive closure of a graph. Another graph $G_R = (V, E_R)$ with $E_R \subset E$ and $H(G_R) = H(G)$ is called the transitive reduction of $G$. Figure 4 shows a transitive reduction of the same graph.

As stated earlier local conflict resolution strategies suffer from deadlocks. A common approach to avoid deadlocks is the utilization of one-way segments to generate appropriate holding instructions when entering two-way segments depending on their current occupancy state.
Step 4: IF P has been found DO remove s(eᵢ) from Eᵣ

Step 3 and 4 ensure that the transitive closure is maintained after removing s(eᵢ). The same graph search algorithm is invoked to determine the path from \( v_{\text{start}}(s(eᵢ)) \) to \( v_{\text{end}}(s(eᵢ)) \) that is used for taxiway routing, to ensure that the same connectivity is given during run time. So far, the algorithm creates a reduced graph with a minimum number of edges and a maximum number of one-way segments what results in suboptimal taxi routes, which are in some cases remarkably longer than optimal routes determined by the graph search algorithm using the original graph or standard taxiway routes. Two extensions were introduced to prevent this inefficiency. The first adds an additional constraint which limits the length of the path determined in Step 3. This leaves edges within Eᵣ if connectivity is maintained but the detour introduced by the reduction would be too expensive. The second extension determines all possible standard taxi routes in a preprocessing step and marks edges that are part of a certain number of routes. If this number exceeds a specified threshold, the edge is excluded from the reduction process. While the first extension tends to reduce the length of possible detours the second introduces a manageable trade off between preserving default taxi routes and having a minimum number and length of two-way segments which must be administrated to ensure deadlock-freeness of taxi operations. Analyzing the average length of all possible taxi routes for Frankfurt airport comparing results of the original graph (3230 m), the reduced graph without (4709 m) and with extensions enabled (3411 m), emphasize that the transitive reduction has minimal effect on the overall route length if used in combination with the presented extensions.

SHARED SEGMENT APPROACH

While the first strategy tends to create an appropriate graph with one-way taxiway segments in a preprocessing step, the second approach determines shared two-way segments during the movement execution. These shared segments are determined whenever a new route \( R_i \) is assigned to an aircraft \( a_i \), by adding \( a_i \) to the list of users of every edge \( e \in R_i \). All active users of an edge might have competing demands when they need to occupy it. In the example given in Figure 2 both aircraft need to occupy the two-way segment at the same time which results in a conflict. After determining the shared segments along its route the aircraft starts the movement execution. During movement it checks successively new edges that are next on its taxi route. If an explored edge belongs to at least one shared segment which is currently occupied by another aircraft a conflict is detected and the aircraft is advised to stop and let the other pass before it checks again if it might enter the shared segment.

While this approach is far more flexible then the transitive reduction it needs to administrate and store additional data in order to manage the utilization of shared segments efficiently. Simulation results depicted that the overhead in computation time and memory consumption is low enough to be applicable in real time operation environments and is less consuming than other long term planning approaches.

Figure 5: The both topmost images show the taxi routes of flights DLH44A and USA701. The third image renders the shared segment demanded by both flights. DLH44A has right of way and occupies the segment while USA701 is waiting in front of the segment’s entry. The last image shows the situation after the conflict resolution. USA701 entered the shared segment after DLH44A left it.

A difference to the taxiway scheduling approach and time window based routing function described in the preceding sections is that there is no requirement for a trajectory prediction in order to operate the system efficiently. All decisions made by the shared segment approach tend to solve conflicts locally and are based on the current traffic situation. This is analog to the decision making of reflex agents in conjunction with an appropriate environment model in order to prevent deadlocks. Reflex agents have the advantage of an inherently transparent decision making process [6] that, if adequate information is presented, is comprehensive for human controllers. An appropriate
environment model or environments where deadlocks can not appear are important preconditions for both presented local conflict resolution strategies. The transitive reduction results in an environment that enables the reflex based decision making to operate without deadlocks. The shared segment approach is an example of an appropriate environment model providing adequate information to the decision making process in order to prevent actions that lead to deadlocks.

There is no need for trajectory prediction because no assumptions on the future movement of aircraft are required to operate a shared segment strategy. This has the advantage that no modelling of uncertainty is needed resulting in a simplification compared to the planning based approaches.

The decision on the right of way if two or more aircraft compete for using a shared segment proposes the possibility for further taxi time optimization or accounting of the demands of different air traffic system participants. Right now a straightforward first-come-first-serve solution has been implemented. Different prioritization criteria would enhance the system in the means of collaborative decision making (CDM) taking airline or ATC demands or constraints like preservation of air traffic flow management (ATFM) slot times and other into account. In addition, an optimization routine could determine a sequence of aircraft granted the right of way that minimizes the total taxi time increasing airport traffic capacity. The latter would incorporate a dependency to a trajectory prediction component in order to estimate the usage time of a shared segment by aircraft possibly resulting in a loss of decision transparency.

Another application of the shared segment approach is the improvement of ground trajectory prediction quality. The information about possible conflicts along a route and estimation of resulting holding times increasing the overall duration of the taxi process would improve the predictability of ground movement operations. Again, this involves an adequate model of aircraft movement in order to determine in advance if a shared segment incorporates a conflict or if the time interval between different uses of the segment won’t overlap.

CONCEPT OF A DECISION MAKING SUPPORT TOOL

The shared segment strategy described in the preceding section copies the behavior of human controllers in the means of a short term conflict prediction. As stated earlier the decision making process inherent to a local conflict resolution strategy is that of a reflex based agent and offers a high degree of transparency towards human controllers.

These properties offer the possibility of an application of the shared segment approach within a decision making support tool for ground movement controllers. The information about possible conflicts and deadlocks gathered through the shared segments are utilized to display appropriate advice messages to the competent controller in order to support its decision making during conflict resolution and detection.

The shared segment assistance system would require either the controllers manually or an automated routing module automatically to input the assigned taxi routes in order to determine shared segments. Whenever the system recognizes a competing demand of two or more aircraft for using a shared segment the system sends a signal (aural, visual) to the controller that an instruction has to be issued to the involved pilots in order to avoid a conflict. This advice has to be generated and presented sufficiently in advance of a possible conflict allowing the controller to decide on an adequate solution without to much time pressure. This would require a prediction of aircraft movement along a given route and incorporates some uncertainty into the system. Position and speed information received e.g. through ADS-B can be used to determine the time or a time interval when the aircraft reaches the conflict segment.

To increase the situational awareness towards possible conflicts and relate them to a certain part of the aerodrome movement area, the shared segment information could be utilized to render areas of high conflict potential. If aircraft enter these regions, the controller is well prepared for conflicts that might appear more often in certain areas of the aerodrome.

In the case of a detected conflict the system is able to display information to controllers relevant for an efficient conflict resolution:

- all aircraft involved in the conflict
- the shared taxiway segment where the conflict appears
- the taxi route of all conflict opponents
- estimated usage intervals of the shared segment for every aircraft
- aircraft related data that supports the decision making towards a right of way sequence, i.e. slot times, delay, airline, aircraft type, any emergency conditions, any other information related to CDM processes, etc.
- a proposal of an optimal decision determined by the system

Depending on the degree of automation the controller could either get unprocessed data to support its decision making or receive a proposal of an optimal decision from the system. The system could display relevant information resulting in the determined decision on demand.

The kind of information presented to the controller, during a conflict situation and the possibility to display the relevant information that led to an automatically determined conflict resolution, both render the high degree of transparency inherent to the local conflict resolution approach. Human controllers use a similar information foundation for decision making, what results in an additional advantage of the proposed tool concerning its acceptance in the operational environment of present-days ATC system.
This section summarizes the advantages and disadvantages of the presented local conflict resolution strategies and compares them to the scheduling approach presented in [7] and the time window based search algorithm from [8].

Advantages of both local conflict resolution strategies are:

- No dependency to a trajectory prediction algorithm.
- Applicable for any airport layout representation based on a directed graph and no need for additional information that must be generated manually (e.g. placement of one-way segments using any kind of graph editor).
- Fully automated conflict detection and straight forward resolution strategy (issuing hold instructions at crossings).
- Prevention of all deadlock situations (assuming a realistically distributed traffic density).
- Minimum computing overhead during runtime and high degree of scalability (real-time capable).
- Shared segments strategy minimizes size of taxiway regions where conflicts might appear and therefore optimizes holding times caused by conflict resolution.

Some disadvantages compared to more sophisticated scheduling and planning approaches are:

- No theoretically time optimal taxi routes are generated.
- Conflicts are solved as they appear. Long term prevention and optimization of holding times is not achieved.
- Transitive reduction might lead to less optimal taxi routes compared to standard surface operations.

Both presented strategies are implemented in an airport traffic simulator and results from preliminary simulation runs at Frankfurt airport with about 200 flights involved showed that taxi times were reduced (~90 seconds per flight for the transitive reduction) in comparison to a chosen reference scenario. This improvement of taxi duration resulted from an optimized choice of holding instructions compared to standard taxi procedures used within the reference scenario. No deadlocks were detected during the simulations even in dense traffic conditions. More simulations will be conducted in order to gather more quantitative results concerning a reduction or increase of taxi times.

Table 1 summarizes the comparison of all presented strategies for automated taxi operations.

<table>
<thead>
<tr>
<th>Taxiway Scheduling</th>
<th>Time Window Routing</th>
<th>Transitive Reduction</th>
<th>Shared Segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Time optimal utilization of predetermined taxi routes</td>
<td>+ In theory time optimal taxi routes</td>
<td>+ Deadlock prevention is done in a preprocessing step</td>
<td>+ Deadlocks are prevented during runtime</td>
</tr>
<tr>
<td>+ Applicable to any airport layout</td>
<td>+ Routing function accounts utilization of taxiways</td>
<td>+ Applicable to any airport layout</td>
<td>+ Linear runtime complexity of algorithm determining shared segments</td>
</tr>
<tr>
<td>- Less comprehensive decision making</td>
<td>+ Applicable to any airport layout</td>
<td>+ Smaller search space for routing function, fewer complexity</td>
<td>+ Applicable to any airport layout</td>
</tr>
<tr>
<td>- Dependency to trajectory prediction and uncertainty in movement execution</td>
<td>+ Polynomial complexity of routing algorithm</td>
<td>+ Straight forward local conflict resolution</td>
<td>+ Optimal taxi routes (spatial)</td>
</tr>
<tr>
<td>- Deadlocks prevented through strict one-way taxiways reducing flexibility in taxiway utilization</td>
<td>- Less comprehensive decision making</td>
<td>+ Transparent automated decision making</td>
<td>+ Transparent decision making during conflict resolution and prevention</td>
</tr>
<tr>
<td>- Replanning needed if predicted movement deviates to much from actual</td>
<td>- Complex routing function</td>
<td>+ Right of way decision supports CDM concepts</td>
<td>+ Different priority criteria for right of way supporting CDM concepts</td>
</tr>
<tr>
<td></td>
<td>- As number of time windows increases, search space and runtime increases</td>
<td>- Suboptimal taxi routes are generated</td>
<td>+ Improvement of ground operation predictability through conflict prediction and holding time estimation</td>
</tr>
<tr>
<td></td>
<td>- Dependency to trajectory prediction and uncertainty in movement execution</td>
<td>- Additional criteria needed to limit reduction and maintain default taxiway routes</td>
<td>+ Optimization of taxi times through routing or control function</td>
</tr>
<tr>
<td></td>
<td>- High replanning effort if predicted trajectory deviates to much from real trajectory</td>
<td>- No optimization of overall taxi times through routing or control function</td>
<td>+ Optimization of taxi times through utilization sequence of shared segments</td>
</tr>
</tbody>
</table>
CONCLUSIONS
Two local conflict resolution strategies have been introduced. The first approach uses transitive reduction to generate appropriate one-way taxiway segments from any directed graph representing an airport layout. Its goal is to prevent deadlock situations and to preserve a preferably optimal taxi process. The second strategy is based on the detection of route segments where conflicts might appear during movement execution. This approach is more flexible and results in slightly more efficient taxi operations but incorporates more computational overhead during runtime.

Both techniques were compared to scheduling and timely planning algorithms, which have the advantage of generating in theory time optimal taxi operations by incorporating a prediction of future conflicts. This is achieved at the cost of a dependency to a trajectory/movement prediction algorithm and remarkable computational overhead, caused by replanning of taxi routes in the case of random incidents like traffic jams at runway entry points. Simulation results and the analysis of the reduced graph and shared segments emphasized the value of both techniques at least within a simulation environment. In addition, a possibility of using the shared segment strategy within a controller decision making assistance system is proposed.

A concept of a decision making support tool for ground movement controllers has been proposed that is based on the shared segment approach. It renders the possibilities of an automation system based on a short term conflict prediction and resolution strategy when applied to an assistance tool that has the goal to relieve ground controller workload caused by conflict detection and resolution tasks.

ACKNOWLEDGMENTS
The authors would like to thank all persons involved for providing helpful comments during the creation of this document and our industry partners for supporting the implementation of the airport traffic simulator.

REFERENCES