

MODELLING AND SIMULATION OF VEHICLE MOVEMENTS USING A SPPTW-ALGORITHM AND THE APPLICATION TO AIRPORT SURFACE MOVEMENTS ANALYSIS

Niclas Dzikus, Volker Gollnick

German Aerospace Center, Air Transportation Concepts and Technology Assessment,
21079 Hamburg, Blohmstr.18, Germany

Niclas.Dzikus@DLR.de (Niclas Dzikus)

Volker.Gollnick@DLR.de (Volker Gollnick)

Abstract

For the optimization of aircraft ground movements a method is described herein based on means of modelling and simulation. The paths of the vehicles are described as a network. Based on graph theory, an algorithm is developed that attempts to find the least time consuming, conflict-free path. The algorithm presented is based on one designed for Automated Guided Vehicles, which was adapted for an Advanced Surface Movement Guidance and Control System at airports. It is derived from a Dijkstra algorithm which calculates the shortest possible path between two nodes in a given network.

Due to the fact that the time dependency of the planned paths are taken into account, approaches like this are called Shortest Path Planning with Time Windows (SPPTW), meaning that the path is segmented into parts of fixed duration. In case of a conflict, the vehicle is delayed in a preceding path segment or rerouted. The calculation of the paths for the different vehicles is initiated by a request, including the nodes at the start and at the end as well as the time at entry.

The results of the simulations are used to estimate the performance of an airport airside system, with particular focus on the taxiway system.

Keywords: graph theory, Shortest Path Planning with Time Windows, airport movements planning, aircraft taxiing

Presenting Authors' biography

Niclas Dzikus received his Diploma Degree in Aerospace Engineering from the Technical University of Berlin in 2008. After working at the Technical University of Braunschweig, Institute of Flight Guidance, he started working at the German Aerospace Center in 2009, faculty of Air Transportation Concepts and Technology Assessment.



1 General

Research in improvements of the air transportation system addressing the achievement of the Advisory Council for Aeronautics Research in Europe (ACARE) goals is often based on system models. The goal of technology assessment is to study and evaluate the impact of new technologies regarding to the fulfilment of these goals. An approach of a comprehensive air transportation modelling system is described in [1]. The provision of adequate mathematical models, either analytical or simulative, is often an important part for the assessment. In this paper a way to study the airport system by the means of simulation is described with focus on the airport airside processes, in particular the taxiway system. Therefore parameters are defined to constitute the model's required capabilities with respect to the object of investigation. With the definition of the requirements an adequate mathematical model can be selected.

In 2 the airport system is introduced and the relevant processes of the airport airside. The main performance indicators capacity and delay are described as well as the so-called taxi times. By the definition of the performance indicators a mathematical model can be selected, that is able to analyze the effect of different input data on these indicators. Some existing analytical and simulative models are introduced and their applicability for airport airside analysis, especially their applicability for the investigation of future scenarios and technologies. The method of choice herein is the use of a graph theoretic approach and a routing algorithm that is capable to reflect the properties of the real system with respect to the performance indicators defined.

In 3 the simplified routing algorithm based on the work for Automated Guided Vehicles (AGVs) presented in [2]-[4] and its application for a Advanced Surface Movement Guidance and Control System (A-SMGCS) by [5] is introduced. The described time-window-based method is a modification of a static method, namely the Dijkstra algorithm, for the planning of conflict-free taxi paths by the use of time windows.

In this context the differences of routing and scheduling should be mentioned since these are actually two related aspects. According to [6] each aspect can be described as follows:

Scheduling - The goals of scheduling are normally related to the processing time or utilization of resources, such as maximizing the throughput, or minimizing the total travel time of all vehicles, and the likes.

Routing - Once the scheduling decision is made, the mission of routing is to find a suitable route (e.g. shortest-distance path, shortest-time path or minimal

energy path) for every entity from its origin to destination based on the current traffic situation.

The application of the algorithm to a simulation model for airport airside analysis is described in 4, together with a discussion of preliminary results, since recorded data for validation was not available at the time of paper submission.

In 5 a short outlook for further investigation and enhancements is given.

2 System Specification

2.1 Airport Airside Processes

The elements of an airport can be divided into two major components, the airside and the landside. On the airside the processes mainly focus on the flow of aircraft, whereas on the landside the flow of passengers is of special interest. Furthermore the airside processes can be divided into the components: runway, taxiway, apron and gate (Fig. 1):

Arriving aircraft land on the runway and move on the taxiways to their allocated gate. After ground handling, i.e. boarding and deplaning of passengers, refueling, cabin cleaning etc., the aircraft is ready for departure and moves on the taxiways to the runway for take-off.

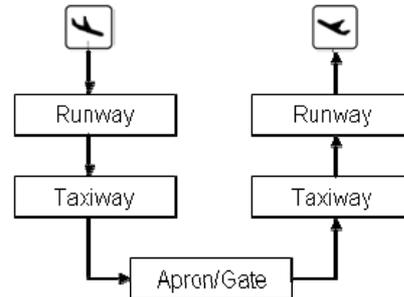


Fig. 1 Processes of the airport airside system

For a precise definition of the taxi-out time for departing aircraft and the taxi-in time for arriving aircraft, the following equations for their calculation are given:

$$t_{TaxiOut} = ATOT - AOBT \quad (1)$$

$$t_{TaxiIn} = AIBT - ALDT \quad (2)$$

ATOT and ALDT refer to the actual take-off time and the actual landing time, i.e. the time the aircraft lifts off and the time the aircraft touches ground. AOBT and AIBT refer to the in-block and off-block time, i.e. the time when aircraft reaches the gate and leaves the gate respectively.

The capacity is associated with the flow of aircraft and is an important performance indicator of the airport airside system. It is important to notice that the capacity of an airport airside is determined by the capacity of the runways, the taxiways or the gates,

whichever is the least. In most cases the runway capacity determines the overall airport capacity.

According to [7] capacity and delay are defined as:

- Capacity: “the number of aircraft operations during a specific time interval corresponding to a tolerable level of average delay”
- Delay: “the difference between the time it would take an aircraft to be served without interference from other aircraft and the actual time it takes the aircraft to be served”

The time an aircraft needs be served without interference from other aircraft is also denoted as the “unimpeded” time.

The interrelationship of the capacity and delay is depicted in Fig. 2. As demand rises, i.e. the number of movements at an airport during a specific time interval, delay rises accordingly due to capacity constraints. Since in most cases the runway system is the “bottleneck” of the airport airside, queuing of departing aircraft at the runways is the main reason for congestion influencing taxi-out time, whereas taxi-in time of arriving aircraft is mainly impeded by queuing at the gates. The practical capacity is the number of movements at a tolerable level of delay, the so called level-of-service, e.g. limited to 4min.

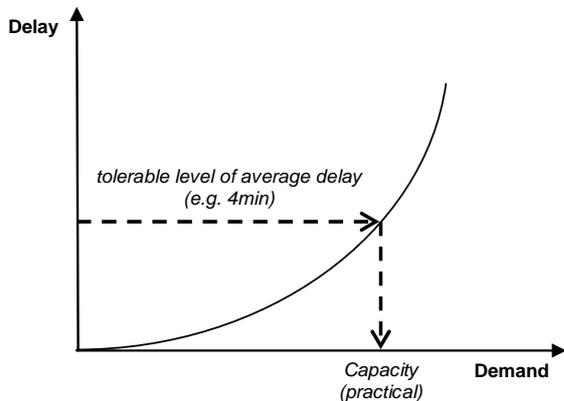


Fig. 2 Determination of capacity by the interrelationship of demand and delay

According to the ACARE goals emissions and noise are a general environmental issue of aviation, which is affected by airport operations with respect to scarce resources and increasing demand. The time each aircraft spends on the taxiway system has great influence on both as it is shown e.g. in [8].

2.2 Mathematical representation

As stated above, taxi time, i.e. taxi-in time for arriving and taxi-out time for departing aircraft respectively, is an important value for assessing the performance of airport airside operations and their environmental impact. It was also described that taxi times are impeded by the interaction with other aircraft, mainly by congestion at the runways or the gates. Therefore a model is needed that is able to replicate congestion

and the resulting taxi times for the determination of delay, and thus the practical capacity, as well as for emission calculation. It should be mentioned here that computation time often is an important factor for the selection of a model, too. Since it is not within the focus of this paper the reader might refer to [5] for further information.

Analytical approaches like the models based on time-distance charts presented in [7] and [9] are easy to implement and deliver rough estimates of capacity and delay. They focus on the capacity of the runway and gate system since these in most cases define the overall airport airside capacity. The method for taxiway capacity determination is based on graphical evaluation and is not able to deliver taxi times directly.

Stochastic models based on the analysis of empiric data are capable to predict taxi times realistically (see [10]). For proper implementation a lot of data is necessary which can often only be provided by the airport itself. Among the actual landing and take-off times as well as the in-block and off-block times needed to calculate the taxi times by using Eq.(1) and Eq.(2), additional data must be acquired. The goal is to show the dependency of taxi times on other values like meteorological conditions, runway and gate used, etc.. The calculated taxi times already contain the effect of congestion. In [11] it is shown that taxi-out time can then be expressed as a function of the number of aircraft on the taxiways.

The effect of congestion on taxi times is shown in Fig. 3 where real data of taxi-out times at George Bush International Airport in Houston is analyzed. As the number of aircraft on the taxiways (N_H) increases, the mean taxi-out time as well as the variability increases accordingly due to congestion effects by aircraft interaction.

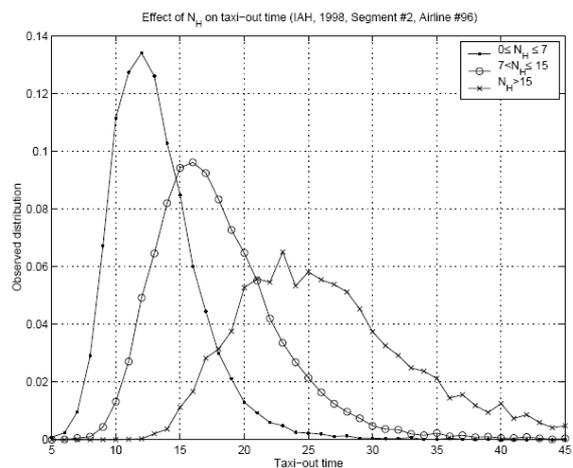


Fig. 3: Distribution of taxi-out times subject to the number of aircraft on the taxiways ([10])

Since the stochastic model in [10] is based on empirical data, it is very accurate and delivers valid results. On the contrary, future scenarios or changes in

the infrastructure can hardly be reflected, since these impacts are not captured in historical data.

Commercial software for airport simulation is often based on graph theory. Commonly used by airport planners is the simulation software SIMMOD. The lanes of the airport are modeled as links, i.e. vertices V connected by edges E representing the airport layout as a graph $G = (V;E)$.

As an example an airport layout is given in Fig. 4.

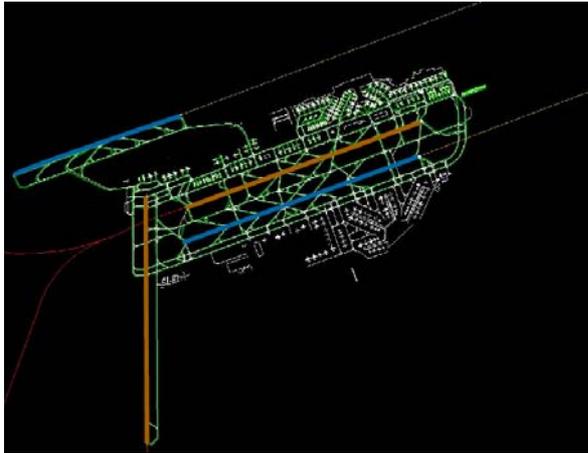


Fig. 4: Screenshot of the simulation software SIMMOD ([12])

SIMMOD is a microscopic fast-time discrete-event simulation model for airfield and airspace analyses, which was validated by the Federal Aviation Administration (FAA). It is widely used to examine the effects of e.g. changes in the infrastructure or operational procedures. A drawback of commercial software is the fact that the source code can not be modified directly by the user. Due to the many input parameters, partly stochastic, this drawback can often be overcome by experienced users. The initialization of such a model is often very time-consuming and requires a lot of expertise.

When using microscopic simulation based on graph theory attention has to be paid for conflicts. According to [2] conflicts can be distinguished not only in congestions and collisions but also include so-called deadlocks and livelocks. As shown in Fig. 5 a deadlock situation occurs when two vehicles wish to allocate the same edge, whereas in a livelock a vehicle is blocked continuously by others.



Fig. 5: Deadlock and Livelock ([2])

In the next chapter a method for vehicle routing is described that is capable for conflict-free route planning, including deadlocks and livelocks.

3 Routing Algorithms

According to [6] the different methods used for routing algorithms can be distinguished in:

- static methods: The entire paths remains occupied until a vehicle completes the tour
- time-window-based methods: The path segments may be used by different vehicles during different time-windows
- dynamic methods: The utilization of any segment of the path is dynamically determined during routing rather than before routing

In the following sections the Dijkstra-algorithm as a static method for shortest path planning is described. Thereafter the algorithm used for the simulation of the airport airside system is described as a modification of the Dijkstra algorithm and the use of time windows. Therefore this approach is denoted a time-window-based method or Shortest Path Planning with Time Windows respectively. For both the applicability for airport surface movement simulation is discussed with respect to the requirements.

3.1 Static Routing (Shortest Path Planning)

The Dijkstra algorithm finds the shortest path from a single starting point to every other node in a network with non-negative weighted edges. The shortest path corresponds to the path with minimal costs in terms of e.g. time, distance, etc. Thus the necessary input parameters for the Dijkstra algorithm are an adjacency matrix representing the graph, a cost matrix representing the weighting of the edges, a source node and a sink node representing the start and the end of the path to be planned.

The basic idea of the algorithm can be described as follows:

- For all adjacent nodes follow always the edge with the least costs (e.g. time) to the start node.
- The path by another edge is only followed if the shorter path segments were already calculated
- If the end node is reached, the computed path must be the shortest one

The costs on the different path segments is fixed once they are computed, thus the Dijkstra algorithm is denoted as a greedy algorithm.

Although conflict-free vehicle movements can be assured at execution time of the routes by a dynamic "reservation procedure", the avoidance of deadlocks and livelocks is a drawback of this method ([4]). This is because the time-dependet behaviour of the network, i.e. the temporary blocking of edges by other vehicles, is not taken into account at the time the route

is planned. Therefore this static approach turned out to be too time-consuming in terms of initialization and computing.

3.2 Time-Window-Based Routing (Shortest Path Planning with Time Windows)

To avoid conflicts already at the time a route is planned rather than at the execution time of route, a time-window-based approach is used based on [2]-[5].

As an extension to the static SPP-algorithm the time-window-based method stores a list of time intervals for every path segment, when the corresponding edge is not blocked by another vehicle. These time intervals are denoted as time windows. Here the complementary set of blocked time intervals is used. For every request the algorithm determines the shortest route from the start to the end node, according to the procedure of the SPP-algorithm. Additionally the algorithm checks within every planning step, whether the edge is possibly blocked by a previous planned path traversing the same edge.

In Fig. 6 a graphical representation of this principle of path planning taken from [13] is depicted. Below the columns three edges are shown linking 4 nodes. The costs on every path segment are defined as 1:

- Fig. 6, “graph with blockings”: The columns represent the time intervals of the time windows, i.e. the green areas. Red areas are blockings of previous planned paths.
- Fig. 6, “new path”: The blue areas are the preliminary blockings of the currently planned, new path. Since the blocking times of the new path do not overlap with previous planned paths the route computed must be conflict-free
- Fig. 6, “graph with new blockings”: The blocked time windows of the new path are stored in the corresponding time lists.

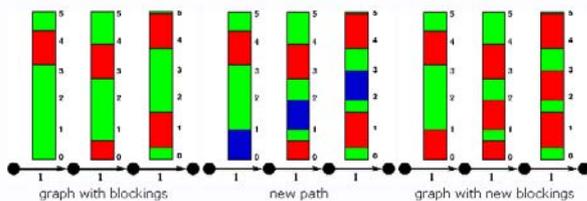


Fig. 6: Graphical representation of the principle of path planning with time-windows ([13])

If the blocking time of the new path overlaps with a previous planned path, the entry time into that path segment is shifted to the earliest possible one. Thus the vehicle is delayed on the previous edge by this specific amount of time. In case of a subsequent conflict, the blocking time of every edge of the new path is adjusted iteratively in the same manner. Thus conflicts like queuing effects can be modelled and deadlocks are avoided. Due to prioritized requests the

occurrence of livelocks can be avoided, too. Since the length of the path segments is fixed, i.e. the distance from the entry node to the exit node of an edge, the resulting speed can be computed by dividing the edge’s length by the computed time on this edge.

Because of the time dependency taken into account by the algorithm, an entry time has to be given for route computation in addition to the adjacency-, costs-matrix and entry-, exit-nodes. Thus a request for planning consists of the start node, the end node and the time of entry into the network.

It should be pointed out that the routes are optimal with respect to the minimum travel time of the current request being computed in the given network. An overall optimum of all requests is not within the scope, but is rather a question of scheduling.

Since the vehicles’ dimensions are not considered by the algorithm directly, the definition of dependent edges is a method to overcome this problem. If an edge is defined as dependent from another both are blocked for the amount of time one of the edges is blocked. As an example for the use of dependent edges, a crossing of two unidirectional lanes is shown in Fig. 7. To avoid conflicts, both lanes have to be signed as dependent, i.e. a_1 is dependent of a_2 and vice versa.

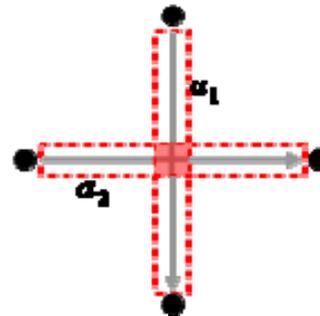


Fig. 7: Example for the use of dependent edges

The definition of dependent edges can be automated by computing polygon intersections. Therefore spatial dimensions for the edges are predetermined. If polygons of edges intersect each other they are stored as dependent.

3.3 Summary

The advantages for using a time-window-based algorithm for the microscopic modelling and simulation of vehicle movements on an airport are:

- The computed routes are conflict-free, i.e. collisions, deadlocks and livelocks are avoided, at the time the routes are planned. In contrast to a static method, SPPTW avoids complex conflict management at execution time of the route.
- Specific requests can be prioritized. For the case of airport movements this allows

arrivals to be planned prior to departures and Ground Support Equipment respectively.

- Spatial dimensions of the vehicles can be modelled by means of dependent edges.

4 Application to Airport Simulation

4.1 Modeling the Infrastructure

For the representation of the airport infrastructure as a graph, the nodes, edges, costs must be defined. According to [5] the longitudinal separation criteria due to the physical dimensions of the aircraft can be modeled by choosing an appropriate average length of the edges. E.g. if the average length of the aircraft at the airport under investigation is around 60m, the distance between the nodes should be accordingly. In Fig. 8 this principle is shown by the fragmentation of an edge with the length of 300m into smaller edges. Thus the capacity of one vehicle for the taxiway lane, represented by edge $a_{4,1}$ and $a_{4,2}$ respectively, is increased to a capacity of five vehicles in the fragmented graph, $a_{4,1,1} \dots a_{4,1,5}$ and $a_{4,2,1} \dots a_{4,2,5}$.

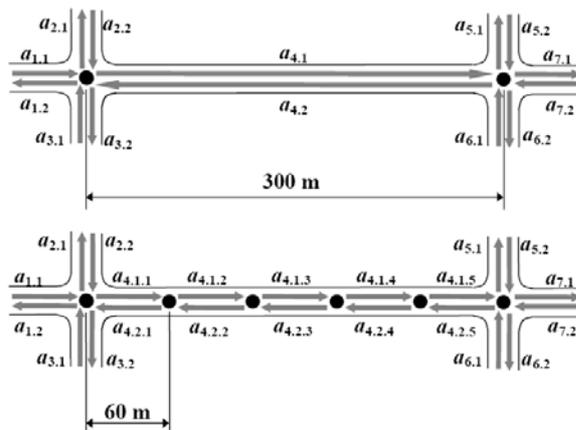


Fig. 8: Fragmentation of a single edge representing a taxiway ([5])

Since a time-window-based routing algorithm is used, the definition of time-based costs is obvious, although additional parameters like energy consumption could be assigned as well. The minimum costs on every edge are the travel times as the division of the edges' length divided by the predefined maximum velocities on the edges. These velocities can be defined individually.

In Fig. 9 an example of a graph $G=(V;E)$ representing an airport airside system is depicted. The upper figure shows a breakdown of the airport airside into the runway, taxiway and apron/gate system. The figures below show the vertices' positions and the edges defined by the adjacency matrix.

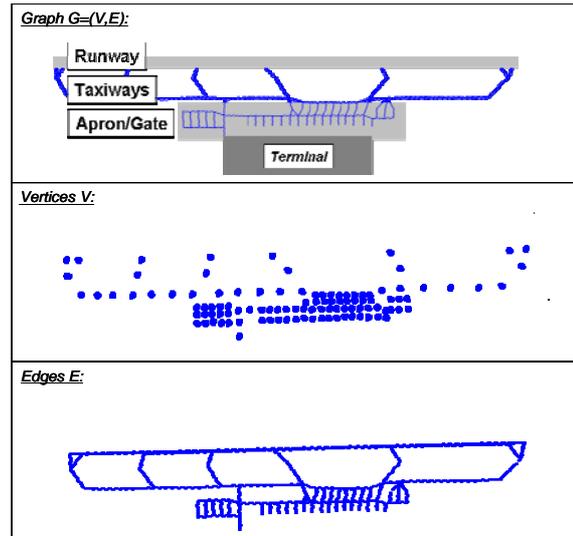


Fig. 9: Example of a graph representing an airport with a single runway

A welcome side effect of the time-window-based approach is the possibility to model runway operations, i.e. aircraft landing and taking off on the runway, in a simplified manner. According to [7] runway operations can be characterized by intermovement times (IMT) between the numerous combinations of arriving and departing aircraft on the runways in use (runway configuration). The IMT result from legal minimum separation criteria to make sure that aircraft can be operated safely. Since the IMT between aircraft can be interpreted as a blocking interval, they can easily be incorporated in the model. This method requires modification of the input parameters to take different aircraft types into account, but has no effect on the algorithm itself. The separation criteria are the reason why in many cases the runways are the "bottleneck" of an airport system in terms of capacity and thus being the main reason for queuing-effects and the resulting delay of departing aircraft.

4.2 Model Input

Despite the network data, i.e. the adjacency matrix, cost matrix, list of dependent edges, etc., the model requires a list of requests in form of a flight plan. Since no real data in form of a flight plan with actual landing and off-block times is available as input for the model, a generic flight plan is generated to show that the airport model captures the main effects of the system.

The generic flight plan consists of the following input parameters for each flight:

- *EntryTime* (for arrivals this is the actual landing time, for departures the actual off-block time)
- *Arr/Dep* (arriving or departing aircraft)

- *StartNode* (for arriving aircraft this is the runway threshold, for departing aircraft the allocated gate)
- *EndNode* (for arrivals this is the allocated gate for departures the end of the runway)
- *AircraftType* (aircraft can be classified as heavy, medium, light according to their maximum take-off weight (MTOW))

In Fig. 10 an example of accumulated demand on a hourly basis is presented. Altogether 570 movements are simulated with two demand peaks of movements at the third and the eleventh hour.

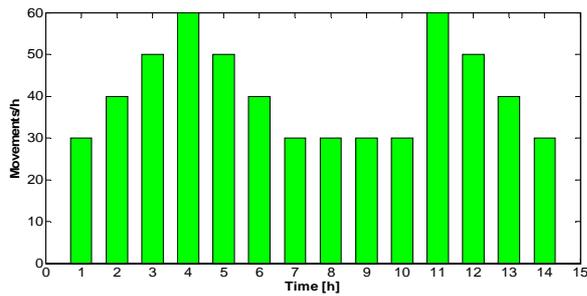


Fig. 10: Scheduled movements on a hourly basis

4.3 Results

The simulation results show the model’s capability to qualitatively capture the dynamic system properties with respect to the distributions of the taxi times, capacity and delay. The requests were processed by a First-In-First-Out schedule, i.e. there was no prioritization of arrivals etc. In addition the gate capacity was not taken into account.

In Fig. 11 the distributions of taxi-out and taxi-in times are shown. As mentioned above runway congestion mainly influences the taxi-out time resulting in longer taxi times for departing aircraft.

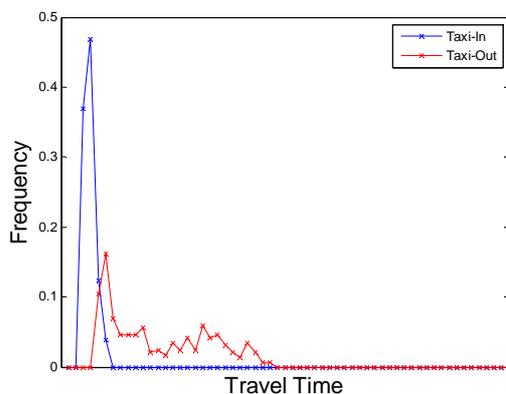


Fig. 11: Relative frequencies of taxi-out and taxi-in times (570 movements)

The corresponding accumulated delay on an hourly basis is shown in the upper plot of Fig. 12. As described in 2.1, delay is the difference of the impeded taxi time and the unimpeded taxi time. The

impeded and unimpeded times are calculated by the SPPTW- and SPP-algorithm respectively. The scheduled movements and the actual movements are also depicted. The green bars represent the scheduled movements according to the flight plan, the red line are the movements actually processed. Due to the mentioned separation criteria for arriving and departing aircraft the single runway system can only handle a specific amount of movements per hour. According to [9] the number of operations per hour is between 50-59 movements, depending on the mix of aircraft types. Thus in periods when more movements are demanded, the over-demand causes delay.

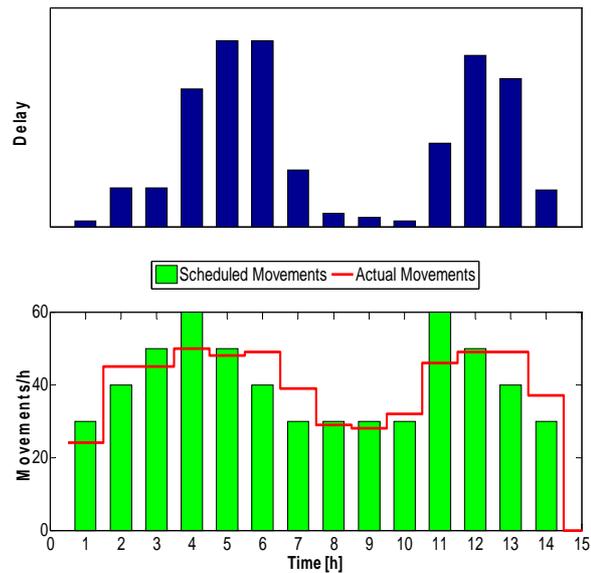


Fig. 12: Delay and Capacity

It has to be pointed out that beside the effect caused by congestion due to limited infrastructural resources other factors affect the flow of aircraft, such as the communication process between flights and the controller, which are not taken into account by the model presented. The Virginia Tech Airport Simulation Model (VTASIM) includes such effects ([14]). VTASIM is a microscopic hybrid airport simulation model. It is denoted as hybrid because aircraft movements are represented by a discrete-time simulation model, whereas the communication process is represented as discrete-event simulation model. Nevertheless the taxiing of aircraft is also modeled by determining a time-dependent shortest path (TDSP) for each aircraft as it is shown in the process-chart of Fig. 13.

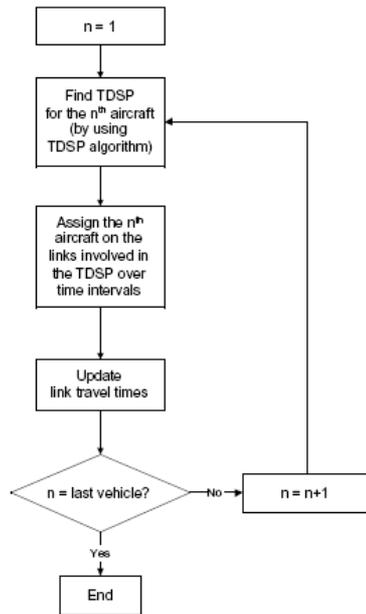


Fig. 13: Process-chart of dynamic taxiing route planning in VTASIM ([14])

5 Outlook

Results show that the system's dynamic behavior is captured qualitatively by the model. A quantitative validation is planned using recorded flight data from specific airports or by benchmarking with more detailed and sophisticated simulation tools or models.

Also the estimation of travel time in accelerated motion is important, especially for the computation of emissions. A way to consider this effect is presented in [15]. In combination to deceleration and acceleration due to conflict avoidance, deceleration due to turns should also be considered. In [5] a way to estimate this effect by the definition of turn costs is presented.

6 References

- [1] Volker Gollnick. Time and Energy Efficient Transport – A Proposal for an Air Transportation system Assessment. 6th MathMod-Conference Conference Proceedings, February 2009.
- [2] Rolf H. Möhring, Ekkehard Köhler, Ewgenij Gawrilow, Björn Stenzel. Conflict-free Real-Time AGV Routing. Operations research Proceedings 2004, p.18-24. Springer Berlin Heidelberg, 2005.
- [3] Ewgenij Gawrilow, Ekkehard Köhler, Rolf H. Möhring, Björn Stenzel. Dynamic Routing of Automated Guided Vehicles in Real-time. Mathematics – Key Technology for the Future Joint Projects Between Universities and Industry 2004-2007, p.165-177. Springer Berlin Heidelberg, 2008.
- [4] Björn Stenzel. Online Disjoint Vehicle Routing with Application to AGV Routing. PhD thesis, Technische Universität Berlin, 2008.
- [5] Torsten Busacker. Steigerung der Flughafen-Kapazität durch Modellierung und Optimierung von Flughafen-Boden-Rollverkehr – Ein Beitrag zu einem künftigen Rollführungssystem (A-SMGCS). PhD thesis, Technische Universität Berlin, 2005.
- [6] Ling Qiu, Wen-Jing Hsu, Shell-Ying Huang, Han Wang. Scheduling and routing algorithms for AGVs: a survey. International Journal of Production Research 2002, p. 745-760. Taylor&Francis 2002.
- [7] F. Poldy. Airport Runway Capacity and Delay: Some Models for Planners and Managers. Australian Government Publishing Service, Canberra 1982
- [8] Nicolas Pujet. Modeling and control of the departure process of congested airports. PhD thesis, Massachusetts Institute of Technology, 1999.
- [9] Robert Horonjeff, Francis X. McKelvey. Planning & Design of Airports, 4th Edition. McGraw-Hill, 1994.
- [10] Kari Andersson, Francis Carr, Eric Feron, William D. Hall. Analysis and Modelling of Ground Operations at Hub Airports. 3rd USA/Europe Air Traffic Management R&D Seminar. Napoli, 13-16 June 2000.
- [11] Husni Idris, John-Paul Clarke, Rani Bhuvanga. Queuing Model for Taxi-Out Time Estimation. ATC Quarterly, 2001.
- [12] <http://www.dlr.de/esug/>
- [13] <http://www.math.tu-berlin.de/coga/projects/traffic/agvrouting/>
- [14] Hojong Baik, Hanif D. Sherali, Antonio A. Trani. Time-Dependent Network Assignment Strategy for Taxiway Routing at Airports. Transportation Research Record: Journal of the Transportation Research Board, Volume 1788, p.70-75. Transportation Research Board of the National Academies, 2002.
- [15] Hyo Young Bae, Ri Choe, Taejin Park, Kwang Ryel Ryu. Travel Time estimation and Deadlock-free Routing of an AGV system. Dynamics in Logistics, p. 77-84. Springer Berlin Heidelberg, 2008.