Deutscher Luft- und Raumfahrtkongress 2008
DocumentID: 81260

ONLINE PASSENGER MOVEMENT FORECAST IN AIRPORT TERMINALS

S. Richter, EADS Deutschland GmbH, Germany
M. Kedikova, EADS Deutschland GmbH, Germany

Abstract

The integration of emerging technologies in existing airport infrastructures will enable the determination of the passenger's position inside the airport terminal within the upcoming years. The accuracy will vary with the implemented technologies.

With the existence of frequent traveller programs from various airlines and travel agencies there is already a detailed collection of personalised data about the single passenger. Is the first step of seamless data exchange between the different stakeholders taken a large number of new applications and services will be possible. Online check-in possibilities and electronic tickets are comfortable services for the passenger, but bring uncertainties for the airlines and airports. The tracking of passengers in the handling process increases the transparency of the passenger for the involved stakeholders. An additional online passenger movement forecast helps with the identification of resource bottlenecks and prevents delays due to no-/late-show passengers.

At the beginning there was an evaluation period for choosing the algorithms. These algorithms for the shortest-path-problem have to meet several specific requirements. After evaluation and identification of the algorithms the efficiency was tested. The results lead to the assumption that even simple algorithms (e. g. breadth-first search) apply to the requirements under special circumstances of small and unbranched airports. The main focus for the application is on large complex airports with several levels, many connecting passages and countless possibilities to spend the spare time. Here lies the need for algorithms that can handle the variable ambient and person specific parameters to obtain the best forecast results.

1. THE PROBLEM

1.1. Initial situation

In the past centuries the need and the willingness to travel has increased. The reasons for that can be found in the globalisation of the world markets and the decreasing costs for travel.

New information technologies lead to the possibility that even intercontinental flights can be booked online from home within minutes. These new services are continuously extended with even more features like online check-in from home or on the way to the airport via mobile phones.

These value added services help the airlines to differentiate and strengthen their position in the marked as well as use their resources more efficiently, but not without any side effects. Existing process flows and the lack of data exchange between the stakeholders (e. g. airlines, airports) narrow the view on their customers using these new services.

The example of an online checked-in passenger travelling without luggage indicates, that the airline will be informed about his presents the moment he passed the gate. On regional flights this is a standard scenario. If such a person becomes a no- or late-show passenger the options of the airline are limited to prevent a delay. Usually the passenger will be paged in an early stage and if it is a valuable customer short delays are acceptable. This leads to additional costs for the airline and interferes with the resources planning of the airport, by blocking the gate longer than expected.

If circumstances in the airport are the reason for a delay in most cases the baggage of the passenger is already loaded into the aircraft. Regulations by the authorities permit the transportation of baggage without the associated passenger. In case of a late-show passenger a complex unloading process will be initiated [1].

1.2. Solutions

Many European research projects dealing with the possibilities of passenger recognition and tracking within the airport terminal in terms of process tracking (e. g. check-in, baggage drop, security check) as well as position tracking [2], [3].

The process tracking requires a seamless data exchange between the airport and airline systems. Position tracking in most cases is based on an additional complex infrastructure installation. The new information generated by both methods increase the transparency of the passengers for all involved stakeholders.

With the needed data available the movement of the single passenger can be forecasted. The airline has now the possibility to identify potential late passengers and take countermeasures before they become critical for the flight.

Also airports are now able to identify bottlenecks on a short-term basis and reassign staff to the critical areas.

2. REQUIREMENTS

The passenger movement forecast has to cope with several requirements and parameters.

This parameter set can be split two categories. One part is handling the infrastructure properties of the airport terminal and the second part represents the individual characteristics of the passenger.

2.1. The airport terminal as graph

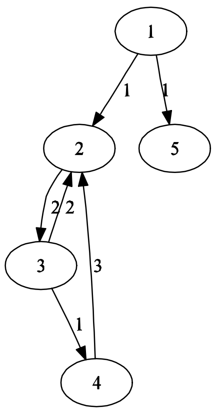
The model for a precise forecast has to include all infrastructure parameters which have an impact on the walking time of persons inside the terminal building.

These parameters include the possibility of having different route options from one starting point to a random endpoint. It is assumed that the different levels within the terminal building are spreading along the x- and y-axis of a Cartesian coordinate system and the z-axis is the vertical link between these levels. Usually a level change can be done by using one of the following three options: normal stairs, escalators or elevators.

The forecast of the passenger is enabled due to the knowledge of its actual position. Also the knowledge about its potential destination is essential for the estimation of the most likely route and the needed walking time. The localisation of the people can be realised in different ways. An exact description of a technical solution for tracking passengers is not subject of this research study. To reach a high compatibility of the forecast model with the different tracking systems the airport terminal is split into zones. The number and size of the zones depend on the tracking solution. It is assumed that points of interest are the basis for a zone. The actual position of a detected passenger is reported as a centre of a particular zone. The moving direction of the person is not relevant for such an implementation.

The zone concept as a baseline for a passenger movement forecast favours the usage of algorithms from the graph theory. The terminal zones are mapped to the nodes of the graph. The edges represent the possible ways between the dedicated zones. Some of the ways are one-way (e. g.: security check) which leads to the usage of a directed graph (digraph) with weights on the edges (PIC 1) [4]. The costs of an edge represent the time a person needs from one node to the next.

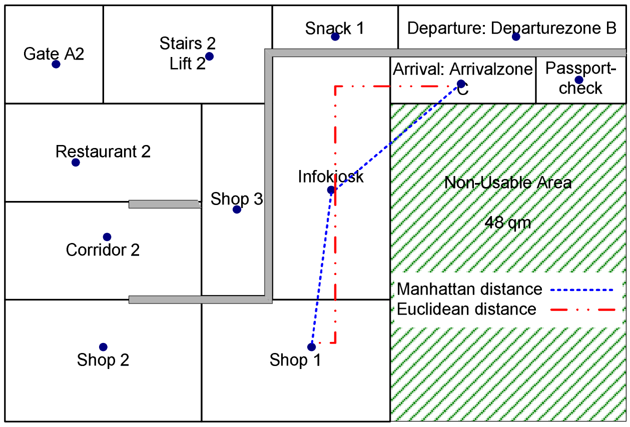
There are several ways to determine the costs of an edge. The empirical method by walking along each edge of the graph and measuring the time is the most costly. The second and third solution involves the Cartesian coordinate system, which is the baseline for the position of the nodes.



PIC 1. Directed graph with costs on the edges

The Euclidean distance between two nodes can be converted into the walking time by talking the average walking speed v_g of a normal healthy human (1,34 m/s) into account [5]. If the nodes are not in the line of sight undercuts can happen and a very optimistic collection of data is the result. An alternative is the usage of the Manhattan distance. The distance of way between two nodes is measured along the axis of the Cartesian coordinate system. This results in values closer to the real walking ways of passengers (PIC 2) [6].

Vertical ways represent a speciality within the terminal building. The level change can only take place on special places where stairs, escalators or elevators are installed. These edges are between two nodes without dedicated zones. The nodes of stairs and elevators are connected with two edges for the different moving directions each. Escalator nodes are only connected to one edge since the movement is one-way in most cases.



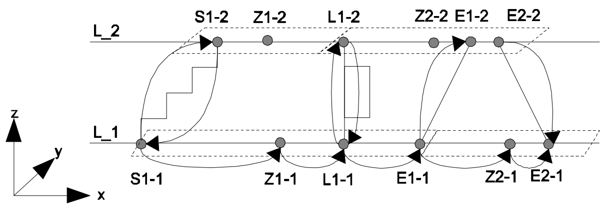
PIC 2. Distance measuring methods for cost determination of the edges

The difference for a level change can be described as follows:

$$dif(z) = z_{vj} - z_{vi} \neq 0 \quad (1)$$
$$\text{for i, j} \in [0, \text{number of nodes}],\ i \neq j$$

Vertical ways are identified by the difference and treated separately. **Fehler! Verweisquelle konnte nicht gefunden werden.** illustrates this case.

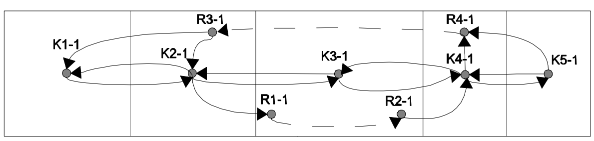
Nodes marked "S" represent stairs, "L" lifts and "E" escalators with a separate escalator for up- and downward movements. Nodes marked with a "Z" indicate all other nodes belonging to a zone. The arrows show the edges of the vertical movements and for completeness some of the horizontal edges.



PIC 3. Illustration of vertical ways

For the estimation of the costs the vertical edges have to be treated differently. Escalators and elevators have a different velocity than the normal walking speed v_g. Also the usage of stairs results in a different velocity of the human being.

Another speciality are moving walkways. These are horizontal edges, which are treated like vertical ones.



PIC 4. Illustration of moving walkways as edges

Like describe in [7] the velocity of a moving walkway can be up to 7 km/h (1,94 m/s). Therefore they are a fast alternative to normal corridors. PIC 4 illustrates the incorporation of moving walkways into the graph. The nodes “R1-1“, “R2-1“, “R3-1“ and “R4-1“ represent the moving walkways. Nodes marked with a “K“ indicate all other kinds of nodes.

2.2. Input parameters

The input parameters can be separated in two categories, passenger related or airport related.

2.2.1. Passenger related input parameters

To the passenger related input parameters belong values which represent the individual needs of the single passenger. Some of the parameters are linked to the chosen flight and its destination, e. g.:

- *Schengenstatus* – the passenger travels within the Schengen area [8]
- *Gatenumber* – the gate represents the default destination of the passenger within the terminal
- *Gateopeningtime* – the time when boarding of the aircraft starts
- *Gateclosuretime* – the time when the gate will be closed for departure.

Other parameters representing the individual characteristics of a person including physical constrains and personal behaviour (partially described in [9]):

- *Age* – age of the passenger
- *Gender* – gender of the passenger
- *Moving limitations* – three groups are linked to moving limitations (*normal, no stairs, lift only*)
- *Route* – the shortest or most likeliest way of the passenger to his destination
- *Destination* – alternative destination besides the gate selected by the passenger itself
- *Actual position* – present zone of the passenger
- *Passenger process status* – the passenger has to pass several stations within the airport to be able to travel by plane [10]. These stations can be listed as process steps:
  - Check-in
  - Baggage drop
  - Security check
  - Passport check
  - Customs
  - Boarding.

The above lists do not represent the complete set of parameters, but include the major characteristics as a baseline for the calculations. The extension of these lists will lead to a more precise and reliable forecast. Nevertheless no personalised data is used, yet.

2.2.2. Airport related input parameters

To the airport related input parameters belong values which represent the local peculiarities of the airport infrastructure, e. g.:

- N_{osl} – Number of open security lanes per area
- N_{opl} – Number of open passport check lanes per area
- N_{ps} – Number of passengers in security area
- N_{pp} – Number of passengers in passport check area

The list is also extendable at any time to increase the quality of the results of the calculation.

2.2.3. Airport graph

Now, the infrastructural characteristics and the input parameters are defined, so that a sufficient graph for the calculations can be set up.
The structure and the weights of the graph are influenced by the airport related and passenger related parameters as well as the Schengen status and the moving limitations.
The airport related input parameters have an impact on the waiting time t_a at nodes of the category “Security” and “Passport”. At these nodes of the “Rank 3” t_a is calculated as follows:

$$\Omega_t\left(v_{rang3}\right) = t_a = \frac{\left(N_p \cdot t_p\right)}{N_o}. \tag{2}$$

Depending on the area (security or passport check) the passenger is at the time, the variables v_{rang3}, N_p and N_o are mapped to the following parameters:

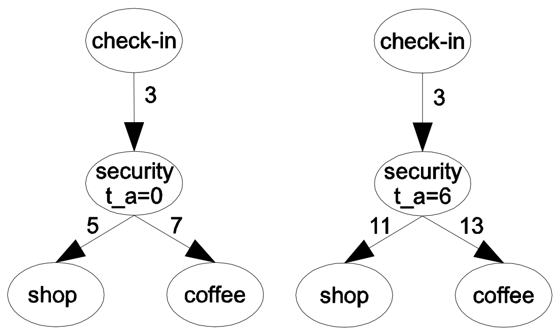
$\Omega_{name}\left(v_{rang3}\right)$	N_p	N_o
“Security“	N_{ps}	N_{osl}
“Passport“	N_{pp}	N_{opl}

TAB 1. Different parameters for calculation of waiting time of a node v_{rang3}

The value t_p indicates the mean processing time for a passenger. The used shortest-path algorithms only take costs at the edges into account while calculating the length of the path. Costs of the nodes have to be transferred to the edges which connect these nodes to other nodes in the graph. Equation (3) shows this relation:

$$\begin{aligned} &\omega_{new}\left(v_{rang3} v_i\right) = \omega\left(v_{rang3} v_i\right) + t_a \\ &v_i, v_{rang3} \in V,\ i \in \left[0, \text{number of nodes}\right] \end{aligned} \tag{3}$$

The queues at the bottlenecks are subject to a constant change. Therefore the costs of the edges with their origin in v_{sec} (node “Security“) and v_{pas} (node ”Passport“) have to be adjusted with every calculation run. PIC 5 illustrates this case.



Graph with standard costs **Graph with increased costs**

PIC 5. Impact of t_a on the costs of the edges

In case of handicaps some passengers might not be able to take stairs or even escalators. The system takes the moving limitations of the passengers into account and calculates a route via possible ways for the specific passenger. To prevent the calculation of a "wrong" route the costs of the not passable edges are raised. In case of a person who can not walk on stairs all edges from the node type "Stairs" are scaled by a large factor. If someone can only take elevators all edges originating in a node of the types "Stairs", "Escalators" or "Moving walkways" are scaled by a large factor. For the algorithm these limitations are treated close to the ones shown in equation (3) and therefore will not be described in further detail.
Another possibility to cope with limitations without changing the costs of the edges is to go for three different graphs. One graph covering all possibilities, a second one without the stairs as nodes and a third one which includes only elevators for level changes and no moving walkways.
The Schengen status of a passenger indicates if the journey of this traveller is within the Schengen area. Passengers with such a destination are not allowed to enter, e. g. duty free areas within the airport. One solution to treat this problem is to compile a list of nodes which are not allowed. If a "Schengen"-passenger selects a destination which is part of this list, an implementation of the algorithm in a real systems would inform the passenger about it and denies the further route calculation to this node.

3. EVALUATED ALGORITHMS FOR THE SHORTEST-PATH PROBLEM

With a potential system two goals for the route forecast can be met. On one hand the route for the single passenger can be predicted and on the other hand the passenger could be informed about the optimal way to his destination. Therefore the algorithms are split into two groups:

- Algorithms, for the *fastest* path and
- Algorithms, for the *most likely* path

The fastest path' are dedicated to passengers who are interested in a guidance service and accept incoming notifications. For all the other travellers routes will be calculated which represents the most likely way of the single passenger. For demonstrational purposes of a feature of the algorithms only it is assumed that women most likely take routes via shops and men prefer routes via restaurants.
To increase the usability of such a solution the passenger has the possibility to select a different destination than the departure gate. If this is the case the algorithm calculates the route until the destination and the route from the destination to the departure gate. Since the selected algorithms allow only one staring point the calculations have to run twice – once from the actual position (s) to the destination (d) and a second time from the destination (d) to the departure gate (e).
In this case it is important that the overall walking time t_{ges} as a sum of the time until the destination t_d and the time from the destination to the departure gate t_e does not exceed the time t_b until the boarding begins.

(4) $t_{ges} = t_d + t_e > t_b$

If the inequation (4) is true, optional the route from the actual position until the departure gate can be transmitted. The algorithms evaluated for the system are breadth-first search, Dijkstra- und A*-algorithm [4], [11], [12].
Passengers at the airport have a designated destination, which is why there is only the way to this destination needed. This leads to the following abort rules:

- For the breadth-first search as soon as the end-node is reached
- For Dijkstra- und A*-Algorithm: if all ways with $d(v_s, v_i) \le d(v_s, v_e)$ for $i \ne e$ are investigated and v_i and v_e are expanded nodes

3.1. Breadth-first search

The breadth-first search is a method to search through all nodes within a graph. It is used for example to find the minimum spanning tree or to find the shortest path from one node to all the others within a graph [13]. The disadvantage of this method is that an optimal solution can only be found by using a graph without weights.
This statement is not conforming to the previous mentioned assumptions. Nevertheless there are special cases of airport terminals which allow the usage of the breadth-first search for a movement forecast application. The requirements for such a terminal would be a small building with one security and one passport area and the possibility to split the terminal into equal zones. This results in equal costs for all edges. After the breadth-first search the way is scaled with the edge costs $\omega(e)$ and the waiting time at the security $\Omega_t(v_{sec})$ will be added. In case of a passenger need to go through a passport check the waiting time for this check $\Omega_t(v_{pas})$ will be added, too. This results in the following equation for the length of the way:

(5) $\omega(W) = \omega(e) \cdot n + \Omega\ (v_{\sec}) + \Omega\ (v_{pas}),$

whereas n represented the level of the destination node.

3.2. Dijkstra

The Dijkstra algorithm is one of the famous algorithms in the graph theory. It finds the shortest path between a fixed node and all other nodes in a weighted graph (G, ω) or network (D,w) [11]. This kind of problem is to be known as *single-source-shortest-paths* [14]. The weights of the edges are defined as follows:

(6) $\omega(v_i v_j) \ge 0, i, j \in [0, k].$

The time complexity of the Dijkstra-algorithm is [15]:

(7) $O(|V| \log(|V|)).$

Except from the abort rule no further changes or extensions were made while the implementation of the Dijkstra-algorithm.

3.3. A*

Like the Dijkstra-algorithm the A*-algorithm is based on the breadth-first search and is used for graphs with positive weights. The A*-algorithm is extended by a heuristic to steer the search in the direction of the destination node [16].

The time complexity of the A*-algorithm is [17]:

(8) $O((|E|+|V|)\log(|V|))$.

The heuristic $h(v_i)$ indicates the estimated costs from a node v_i to the destination node. Closely related to the Dijkstra-algorithm the function $f(v_i)=g(v_i)+h(v_i)$ is used to decide which node will be visited next. Whereas $g(v_i)$ indicates the length of the way until v_i. The node with the lowest value of $f(v_i)$ will be treated as the cheapest way and the algorithm investigates it next. If $h(v_i)=0$ A*-algorithm will function the same as the Dijkstra-algorithm.
Further on three heuristics will be introduced, which were used together with the A*-algorithm:

- *Level*-heuristic
- *Distance*-heuristic
- *Gender*-heuristic.

3.3.1. Level-heuristic

The level-heuristic is based on the difference between the level numbers. It is assumed the passengers prefer to stay on the same level if possible and that they avoid unnecessary level changes.
With the function $L: V \rightarrow N$ to every node the level of its location is assigned. The heuristic $h(v_i)$ represents the difference between the level numbers of the investigated node v_i and the destination node v_e:

(9) $$h(v_i)=|L(v_e)-L(v_i)|, \quad \text{for } i,e \in [1, \text{number of nodes}].$$

The direction of the movement during the level change has no impact on the heuristic. This is the reason for using the absolute value of the difference. The routes via nodes with $h(v_i)=0$ are in favour, because they are on the same level like v_e. Routes with $h(v_i)>0$ are path' that include at least one level change. The level-heuristic has a great potential to calculate the most likely way of a passenger.

3.3.2. Distance-heuristic

The distance-heuristic is an estimation of the distance left to the destination node. The Floyd-Warshall-algorithm is used on the airport graph with its original weights [18], [19]. A matrix is created which includes all shortest path' between all nodes in the graph. As a heuristic the column of the matrix with the distances to the destination node is used. The closer the node to the destination the smaller the value for $h(v_i)$. It is $h(v_e)=0$. The distance-heuristic is valuable for calculation of the fastest path.

3.3.3. Gender-heuristic

The gender-heuristic is valuable for the calculation of the most likely way of a passenger. For demonstrational purposes only the following assumptions have been made to show the functionality of such a heuristic:

- Women spend more spare time in shops,
- Men spend more spare time in restaurants,

This is the reason why the calculated route to the gate will go past these zones accordingly. The chosen method allows selecting all different kinds of preferred areas within the airport terminal as a dependency of the passenger's gender. The heuristic itself is based on the distance-heuristic and some of the heuristic values in the graph are scaled with $a, a \in (0,1)$:

- If the passenger is a woman, the values of $h(v_i)$ to be scaled are of the type "Shop"
- If the passenger is a man, the values of $h(v_i)$ to be scaled are of the type "Restaurant".

This leads to a decrease of the estimation values of these nodes and the route prediction is steered in the direction of these nodes.

4. OUTPUT PARAMETERS

The implementation of the introduced algorithms can generate a large variety of values dedicated to a single passenger. In the following paragraph the most valuable parameters are listed:

- *Time to gate* – calculated time the passenger needs to reach the gate
- *Time to destination* – calculated time the passenger needs to reach a destination of its own choice
- *Waypoints* – list of nodes along the way to the passengers destination
- *Passenger status* – a parameter that indicates time critical levels of a certain passenger, e. g. different alert levels about potential late-show passengers
- *Delay cause* – different reasons for a passenger of being late:
  - *Security check* – long queuing time at security
  - *Passport check* – long queuing time at passport
  - *Passengers' fault* – the passenger itself is the reason for being late.

If a passenger has not additional destination selected the parameter "destination" equals the departure gate.

5. ANALYSIS AND RESULTS

As a conclusion an overview will show the quality of the tested algorithms for the given problem. In addition different test runs were performed to give statements about the runtime of the algorithms.
To compare the efficiency of two heuristics or the performance of two algorithms an appropriate function is needed. The performance of the shortest-path search can be measured with the penetrance [12], [20].

(10) $P=\frac{L}{T}$

The level of the destination node is represented by L and the number of investigated nodes excluding the starting node is T. The interval of the penetrance is $[0,1]$. The closer the value to 1 the narrower the search graph is. If the search tree is wide, meaning the algorithm had to investigate many nodes the value of the penetrance is closer to 0.
The penetrance was calculated for the 5 different investigated algorithms, breadth-first search, Dijkstra-algorithm and A*-algorithm with level-heuristic, distance-

heuristic and gender-heuristic. The algorithms were used on a graph with 27 nodes and 63 edges. The costs of the edges are positive integral numbers $\omega(e) \in \mathrm{N}$. Five passenger datasets with the following attributes were used for testing:

1) Man, no moving limitation, 39 years old, actual position: check-in, departure gate A2, destination of its own choice: info-kiosk
2) Woman, no moving limitation, 39 years old, actual position: check-in, departure gate A2, destination of its own choice: info-kiosk
3) Man, no moving limitation, 32 years old, actual position: Restaurant 1, departure gate B11, no other destination
4) Man, in wheelchair, 32 years old, actual position: Restaurant 1, departure gate B11, no other destination
5) Man, with crutches, 32 years old actual position: Restaurant 1, departure gate B11, no other destination.

With these five test cases the usability of the algorithms and their implementation characteristics for a potential real system were tested. In the first and second case the passengers have two routes because of their own chosen destination. Therefore two values for T and P are calculated. In TAB 2 the results of the test runs are listed. The abbreviation "A* FW" stands for A*-algorithm with the distance-heuristic based on a matrix of the Floyd-Warshall-algorithm.
The A*-algorithm with distance- and gender-heuristic based on the distances between the nodes deliver the best values for the penetrance. The A*-algorithm for the level-heuristic has poor penetrance values because in all five test cases a level change takes place. This leads to the "useless effort" of the heuristic to find a way in the same level without any advantages. For all penetrance values is $P \geq 0{,}5$, because of the narrow graph.

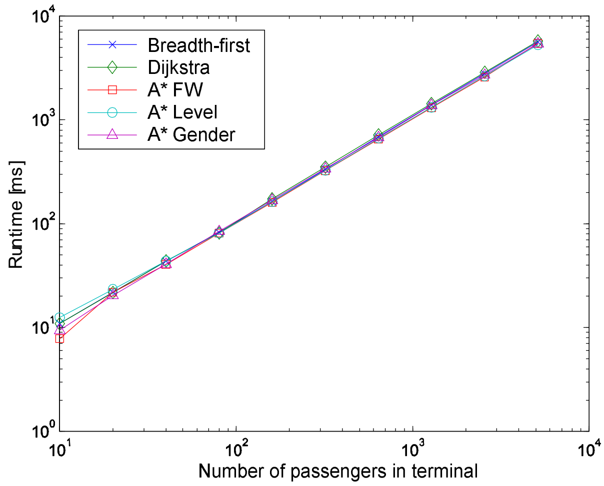
	Breadth-first search	Dijkstra	A* FW	A* level	A* gender
1. case	P_1≈0,64 P_2≈0,54	P_1≈0,64 P_2≈0,54	P_1≈0,78 P_2≈0,88	P_1≈0,64 P_2≈0,54	P_1≈0,82 P_2=0,7
2. case	P_1≈0,64 P_2≈0,54	P_1≈0,64 P_2≈0,54	P_1≈0,78 P_2≈0,88	P_1≈0,64 P_2≈0,54	P_1≈0,78 P_2≈0,88
3. case	P=0,5	P=0,5	P≈0,67	P=0,5	P≈0,73
4. case	P=0,5	P=0,5	P≈0,73	P=0,5	P≈0,67
5. case	P=0,5	P=0,5	P≈0,73	P=0,5	P≈0,67

TAB 2. Penetrance

The different values for P in the test cases 1 and 2 demonstrate the functionality of the gender heuristic. The following conclusions are taken from the test results:

- For the calculation of the fastest route the A*-algorithm with distance-heuristic has the greatest potential
- For calculations of the most likeliest route the A*-Algorithm with gender-heuristic has the greatest potential

In both cases the heuristic is based on the Floyd-Warshall-algorithm which can also be used as a baseline for other heuristics.
The used graph with 27 nodes and 63 edges is a small graph for a real airport. This was the reason to run some tests on a larger graph to prove the usability of the system for a real environment.
The runtime measurements took place on two graphs, one with 50 nodes and 109 edges and a second one with 100 nodes 227 edges. The costs of the edges are positive integral numbers $\omega(e) \in \mathrm{N}$.
Additionally 10 passenger datasets were generated with attention to different combinations of parameters.
The first test measures the runtime of the program by an increasing number of passengers in the terminal. The number of passengers was doubled each time beginning with 10 up to 5120 datasets. This figure represents the average number of passengers being in the terminal at the same time on a middle sized airport[1].
The ratio between runtime and number of passengers varies between 0,78 and 1,12. The result is a close to linear gradient of the curve through the measurement points (PIC 6).

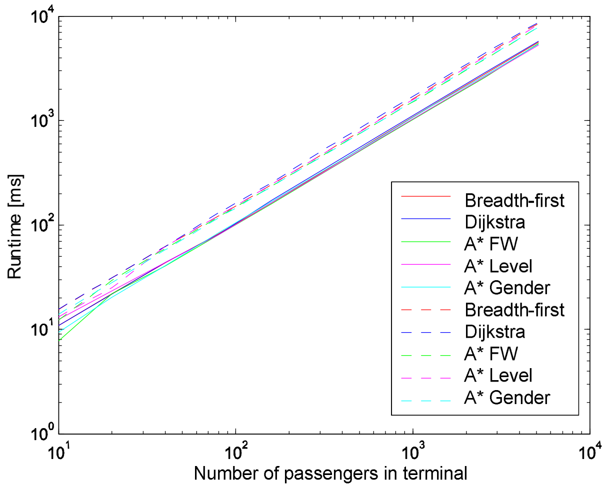


PIC 6. Runtime with increasing passenger numbers

In addition the influence of the size of the graph on the runtime was tested. The used graph had twice the number of nodes and more than twice the number of edges compared to the first runs. The same passenger datasets were used.
The tendency is nearly the same like with the graph with 50 nodes. The ratio between the runtime and the number of passengers varies between 1,24 and 1,74. The close to linear gradient while increasing the passenger number can be seen as well.
PIC 7 illustrates the behaviour. The continuous lines represent the runtime measurements for the graph with 50

[1] These figures are based on a press release fort he Hamburg airport mentioning about 35000 passengers a day [http://www.airport.de/de/pressearchiv.phtml?start=6&year=2008&month=Januar&searchterm=&showdetail=7]. Taking a mean visiting duration of 2h per passenger and the opening hours of the airport into consideration in average 4000 passengers are present at the airport at the same time.

nodes and the dotted lines the graph with 100 nodes.
The A*-algorithm as well as for the penetrance and for the runtime tests delivers good results. The choice of the heuristic is strongly related to the structure of the graph. As an example for this statement the implementation of the level-heuristic can be taken. In general heuristics based on distance measurements seem to be a good implementation for airport graphs.



PIC 7. Runtime with increasing passenger numbers and 50/100 nodes

The Dijkstra-algorithm and the breadth-first search need a little bit more time for the shortest-path calculations compared to the A*-variants in most of the cases.

6. REFERENCES

[1] SITA, *Baggage Report 2008*, SITA 08-REP-001-1, 2008
[2] http://www.e-cab.org/
[3] OpTag, gefördertes EU-Projekt des FP6, Contract No. AST3-CT-2004-502858
[4] Robert Sedgewick, *Algorithmen*, Addison-Wesley, 1992
[5] D. Helbing, I. Farkas, P. Molnar, and T. Vicsek, *Simulation von Fußgängermengen in normalen Situationen und im Evakuierungsfall*. Pages 205-248 in: W. Freyer, and S. Groß (eds.) *Tourismus und Sport-Events* (FIT-Verlag, Dresden), 2002
[6] Jon Orwant, Jarkko Hietaniemi & John Macdonald, Deutsche Übersetzung von Andreas Karrer, *Algorithmen mit Perl*, 1. Auflage, April 2000
[7] Heribert Klein. *Schneller ans Ziel kommen*. Das ThyssenKrupp Magazin, (2), 2003
[8] Verordnung (EG) 562/2006 des Europäischen Parlamentes und des Rates vom 15. März 2006 über einen Gemeinschaftskodex für das Überschreiten der Grenzen durch Personen (Schengener Grenzkodex), *Amtsblatt der Europäischen Union*, 13.04.2006, L 105/1
[9] Wsewolod M. Predtetschenski, Anatoli I. Milinski. *Personenströme in Gebäuden – Berechnungsmethoden für die Projektierung,* Verlagsgesellschaft Rudolf Müller Köln-Braunsfeld, 1971
[10] SPT Interest Group, Simplifying Passenger Travel (SPT) Program, *Air Passenger Process*, June 2004
[11] C. Tietz, *Adaptive Pfadplanung mit Verfahren zur heuristischen Graphensuche*, R. Oldenbourg Verlag, 1996
[12] Judea Pearl, *Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley Publishing Company, 1984
[13] Thomas H. Cormen, Charles Leiserson, Ronald L. Rivest, Clifford Stein, *Introduction to Algorithms*, MIT Press, 2nd edition 2001
[14] E.W. Dijkstra, A note on two problems in connection with graphs, *Numerische Mathematik*, Volume 1, pp. 269-271, 1959
[15] http://www.boost.org/doc/libs/1_34_1/libs/graph/doc/index.html
[16] Pedro Felzenszwalb, David McAllester. The generalized A* architecture, *Journal of Artificial Intelligence Research*, 2007
[17] Kris Beevers and Jufeng Peng. *A* graph search within the bgl framework*, 2003.
[18] Robert W. Floyd, Algorithm 97 (SHORTEST PATH), *Communications of the ACM*, 5(6), p. 345, 1962.
[19] Stephen Warshall, A Theorem on Boolean Matrices, *Journal of the ACM*, 9(1), pp. 11-12, 1962.
[20] Nils J. Nilsson, *Principles of Artificial intelligence*, Springer-Verlag Berlin Heidelberg, New York, 1982