Decision-support methodology to manage the efficiency of ground movements in a critical environment

Ricardo Mega Machado
Instituto Superior Técnico, Portugal
ricardo.machado@ist.utl.pt

Abstract

In complex environments, as in the case of an airport, decision-makers need to be continuously informed about ongoing operations. Such service requires the implementation of a control mechanism capable of providing an effective support at managing traffic ground movements in real-time. Nowadays, technology became available for the implementation of different levels of location-based services aiming to ensure the safety and efficiency of airport surface traffic under all circumstances with respect to traffic density, visibility and complexity of the airport layout. The paper presents an in-depth analysis of a methodology that implements the routing and guidance requirements specified by the Advanced Surface Movement Guidance and Control System (A-SMGCS) defined by ICAO and EUROCONTROL. This includes the analysis and discussion of a site test implementation of the proposed methodology with traffic data from Lisbon airport, in addition to an algorithm to dynamically deal with conflicts and deviations from assigned routes.

Keywords: A-SMGCS, Location-based services, Routing, Guidance, Shortest path problem

1. Introduction

Improving the safety and efficiency of aircraft and vehicle movements in the European airports in all weather conditions is the main objective of the A-SMGCS strategy defined by the European Organization for the Safety of Air Navigation (EUROCONTROL) and by the International Civil Aviation Organization (ICAO). From a safety point of view, the A-SMGCS strategy is particularly relevant regarding the avoidance of incidents caused by traffic congestions or lack of synchronization. ICAO also defines the A-SMGCS as a "system providing routing, guidance, surveillance and control to aircraft and affected vehicles in order to maintain movement rates under all local weather conditions within the Aerodrome Visibility Operational Level (AVOL) whilst maintaining the required level of safety" [1].

A-SMGCS has four distinct levels of implementation, which include surveillance, automated monitoring and alerting functions, routing, and automated aircraft guidance functions, with increasing implementation complexity from level one to level four [2].

- **Surveillance** (level I): positioning of moving objects within the airside area of an airport.
- **Control** (level II): detection and resolution of safety hazards and other conflicts.
- **Routing** (level III): generation of a route for each aircraft in the movement area of an airport, allowing adapting a possible path deviation.
- **Guidance** (level IV): provision of guidance indications to allow the pilots maintaining a pre-defined path.

The implementation of the A-SMGCS requires the combined use of accurate localization techniques, fast wireless communication networks, low-cost embedded systems installed on-board the vehicles, centralized and on-board control services for surveillance and guidance, and geographical information systems enabling advanced graphical-user interfaces for mapping the airport environment. In recent years, appropriate technology became available for the implementation of the different levels of A-SMGCS. Indeed technological advances in wireless communication together with the progress in Global Navigation Satellite Systems (GNSS) and Real Time Locating Systems (RTLS) allow small electronic receivers to calculate the precise time and determine their location within a few meters [3].

In fact, the integration of location-based data collected from state-of-the-art wireless technologies with performing embedded systems enables, for instance, the transmission of data to a central system. This data includes the taxiing aircrafts identification, position and speed that is integrated with flight information to automatically determine the route from the current position to the corresponding parking place [3]. During periods of low visibility, such location-based services would enable airport stakeholders to obtain valuable information about on-going surface movements without having to exclusively rely on radar data and radio communications to identify conflicts.

Along the last decades, the growth in air traffic caused an increase of the workload of both field and control personnel, which have to manage and coordinate ground movements to achieve tied schedules. Such complexity leads to the lack of resources and very often the pressure imposed by airline companies to reduce turnaround times, causes safety breaches, in particular those derived from human error. Despite of such stressing conditions, surface operations mostly rely on the principle of "see and be seen" to maintain a safety spacing between vehicles and aircrafts or even to identify intersections. However, relying on visual observations becomes complex and often misleading, particularly at rush hours when many operations must be managed simultaneously, sometimes under low visibility conditions [2].

Nowadays, only a limited set of airports are equipped with the first two levels of A-SMGCS, for instance Frankfurt Main (Germany) or Charles De Gaulle (France) have invested in new electronic means to provide accurate and reliable positioning information [4]. However, levels III and IV are only predicted to be implemented from 2015 onwards [5]. This paper presents a solution that was implemented following the methodology proposed within an MSc thesis for creating a system capable to provide location-based services in compliance with the A-SMGCS requirements for levels III and IV. The methodological approach takes a graphical representation of all the airport taxiway segments, creating a graph with nodes and arcs, where each taxiway segment is represented as an arc and segments intersection as a node. A routing algorithm takes all the static information to generate the shortest path between every pair of nodes. The conflicts resulting from dynamic information (e.g. conflicts derived from uncoordinated aircraft surface movements) are solved by a Mixed-Linear Integer Programming (MILP) solution that addresses the dynamic behavior of the ground movements by re-routing the mobiles through alternatives or simply by defining a set of holding points.

The remaining of the paper is organized as follows. Section 2 presents the state-of-the-art concerning the location technologies and the static and dynamic algorithms. Section 3 starts by presenting the architectural methodology, followed by a description of the implementation process. Section 4 presents the business context used to evaluate the solution. The results of the site test are presented in Section 5. Finally, conclusions and future work are discussed in Section 6.

2. Related work

Nowadays, the monitoring of airport procedures can be automated through the use of Location Based Services (LBS). Such services and related location technologies have evolved to a point that enables the implementation of business processes requiring a continuous surveillance of moving objects not only with a good precision but most of all with a higher accuracy to avoid false alerts. Such LBS start to be seen by airport stakeholders as a strategic tool helping them to better manage ground operations, minimizing safety hazards while enhancing operational efficiency. The following sections outline some technological issues to support the surveillance of ground movements in a critical infrastructure, as it is the case of an airport.

2.1. Location Based Services

In an airport environment, surveillance of moving objects can be handled by location-based services. It requires the coordination of multiple components, namely onboard units (e.g., vehicles) equipped with location-based technologies capable at transmitting their position continuously. Such onboard units are known as cooperative devices. The existence of a communication infrastructure responsible to transmit, in real-time, the reported position to a central system, together with a main application that centralizes the processing of all location-based information to detect any safety hazard situation or infringements to business rules. The typical network for the implementation of LBS is presented at Figure 1. The segmentation into a set of technological components is recommended not only to support the autonomy of the work to be performed by each component but most of all to cope with scalability requirements (e.g. in large airports).

A LBS is defined by the international Open Geospatial Consortium (OGC) as a "wireless-IP service that uses geographic information to serve a mobile user" [6]. LBS systems provide services that are based on the current location of the monitored object. Besides positioning data, the service can operate with additional parameters such as the destination, circulation direction or speed limit [3].

The Positioning component represented in Figure 1 is responsible for collecting positioning data related to moving objects, which correspond to end-users (e.g. vehicle or aircraft) that are moving within the monitored area. Their real-time positioning is captured through location technologies, which are discussed in Section 2.2.

The Communication Network component is responsible to perform the validation and data fusion of all positioning data transmitted by each device through a wireless network or a TCP/IP data link. The data fusion consists in transforming heterogeneous positioning data into a standard format

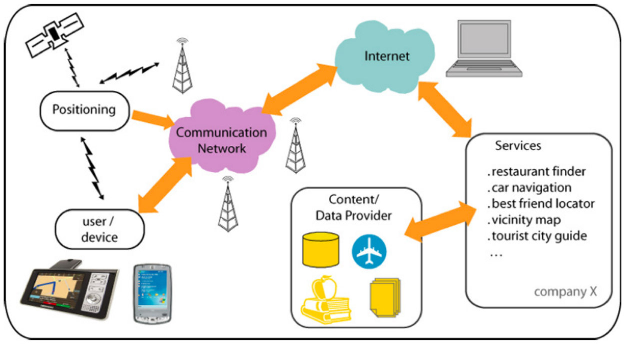

Figure 1: LBS components and their interaction [7].

that the application server understands [3].

The Service Provider component is responsible for computing the positioning data against pre-defined business rules, as well as performing the data fusion with business data collected from existing Content and Data Providers, for instance mobile terminals, flight scheduling, operational tasks assigned to field workers and airport status. At the Application Server, within the Service Provider, the location of each object is presented as a moving point feature over a map-based layout, defined by a set of overlapped thematic layers which characterize the current status of the airport. The airport spatial context and point features are managed by a Geographical Information System (GIS) engine, described in more detail in Section 2.3.

2.2. Location technologies

There are two main categories of moving objects: cooperative and non-cooperative. Targets are considered cooperative when they are equipped with a transponder that is able to communicate its precise location to the Service Provider, while the monitoring of non-cooperative objects requires the ability to detect any mobile within the operational area without the collaboration of the target [1]. The positioning of those objects requires the ability to work effectively in both indoor and outdoor environments. An indoor environment relates to a place where the satellite signal can not be reached (e.g. inside a building), while an outdoor environment is associated to any place where it is possible to maintain a continuous line-of-sight to the satellite. The transition between these two environments is another critical situation that might require location technology redundancies (e.g., GPS, RFID and Wi-Fi) [8].

Table 1 summarizes the technologies used to test and validate the quality of the proposed solution for both environments taking into considerations their accuracy, reliability and range. Although Radio Frequency Identification (RFID) is the more accurate technology, it only provides a very short communication range. However, it is the most promising one for indoor environments. On the other hand, Automatic Dependent Surveillance Broadcast (ADS-B) and Global Positioning System (GPS) seems to be the most suitable technologies for the detection of aircrafts and vehicles, respectively. Besides their high reliability and large coverage range, these outdoor technologies achieve an accurate and cost-effective positioning. On the other hand, Surface Movement Radar (SMR) is an expensive solution that allows to capture every moving object in a large coverage area, with a medium level of reliability.

2.3. Geographical Information Systems

The previous sections described how LBS provide the positioning in real-time about every moving object within the airside of an airport. This information is required to improve the situational awareness of airport stakeholders at the control centre, keeping them well informed about on-going operations through a map-based display with GIS functionalities enabling dynamic interactions with the map features.

This information is managed by a GIS engine that provides an abstraction of the layout complexity as a set of independent thematic layers, each one representing a specific operational area or providing particular features. These layers include taxiways, runways, stand parking areas, as well

Table 1: Location technologies performance.

Technologies		Indoor	Outdoor	Accuracy	Reliability	Range	Cost
Non-cooperative technologies	Magnetic sensing [9]		✓	< 10m	High	< 20m	Low
	Image-based [10]		✓	–	Medium	–	Medium
	SMR [11]		✓	< 5m	Medium	2-3km	High
Cooperative technologies	GPS [12]	✓[2]	✓	5-10m	High	[3]	Med./High
	RFID [4]	✓		5cm-5m	High	3cm-10m	Low
	Wi-Fi [13]	✓	✓	2-100m	Medium	50-100m	Medium
	ADS-B [14]		✓	–	High	≈ 3km	Medium
	MLAT [15]		✓	6-60m	High	–	Medium

as traffic information labels that are refreshed every second. The airport layout is represented as a set of overlapped operational areas, each one with a specific meaning and metadata that are relevant to characterize events within the exact spatio-temporal instance they occur. This means that the features of each layer are associated with a set of metadata. For instance, taxiways have associated, for each segment, a length, traffic circulation rules and a specific speed limit [3]. This information allows generating a weighted graph that corresponds to a network of segments, where each arc has a weight associated. This weight is the length of the segment linking two nodes. The distance between every pair of nodes is stored in a two-dimensional adjacency matrix, each cell representing the length between two nodes [16].

2.4. Routing algorithms

The metadata provided by GIS allows generating a weighted graph that is mainly used by the routing algorithms to solve the shortest path problem. This consists at finding the best path between a source node and a destination node. Metadata are quite relevant at this process because they enable, for instance, accurate definition of the cartographic position of each sub-element of the segment, the length of the segment and the circulation rules associated to a specific segment [3].

2.4.1 The shortest path problem

There are several algorithms providing static route planning. Within the scope of this paper, three algorithms were considered.

- **Dijkstra** solves the shortest path problem by searching the minimum length from a given source node to all the other nodes in the network. The algorithm uses two vectors: one for the set of unprocessed nodes and a second vector with the shortest path distance found between the origin and each of the other nodes. Starting from the original node, each unprocessed node is processed individually in order to select the immediate successor that is closer to the current node. At the end, the second vector has the minimum distances values from the origin to all the other nodes in the network. So, the shortest path is determined by checking which nodes belong to the path, starting from the destination to the origin. This is possible because the length of each arc is known and the minimum distance between the origin and the other nodes is stored in the second vector. By comparing the value to the destination and the distance to its previous immediate successors, the node before the destination is determined. The same process is repeated until the origin node is reached [17].
- **A*** follows the same principle as Dijkstra, but uses an heuristic function that estimates the more promising node that will allow to reach the destination first. While Dijkstra have to check every node of the network to make a decision about the direction to follow, A* is based on a prediction that indicates which node is closer to the destination. The better the estimation accuracy is, the better the performance of the searching algorithm is [18].
- **Floyd** was designed to provide the shortest path between each pair of nodes in a weighted network. The algorithm works in three sequential steps: first, it computes the shortest distance between each pair of nodes, then updates a route matrix that contains the intermediate nodes connecting each pair of nodes, and finally the optimal path is determined [19].

2.4.2 The *k* shortest paths problem

Assigning routes to simultaneous movements at the same time requires the ability to find alternative paths when there is a conflict. The computation of *k* shortest paths to the same destination is an efficient solution that provides a list of *k* alternative routes, where *k* is an arbitrary natural number. The following algorithms were considered in the resolution of the *k* shortest paths problem.

- **k-PathA** resolves the *k* shortest simple path problem by following a forward/backward chaining approach to generate loop free pathways from a seed node to a target node. This mechanism is based on successive expansions and reductions operations starting from the two opposite nodes. The expansion consists in searching forward from the seed for outgoing links and backwards from the target for incoming links. The added nodes from the seed are then compared with every node added from the target and, if there is a common node, the pathways are connected, evaluated and stored. This step is repeated until there are no new nodes to add or the length of the paths reached a predefined threshold. For each iteration, the number of added nodes is reduced to a user-defined value, selecting the nodes with lower distance weights from the expanded nodes. Finally, all possible combinations between pathways from the seed to the common node are generated, as well as from the common node to the target. The *k* shortest paths are then selected [20].
- **Eppstein** first applies Dijkstra in the reverse sense, searching backwards from the destination *d* to all the other vertices in the graph. The result is a shortest path tree *T* with the shortest path from any vertex to *d*. Then, following the shortest path from the origin to the destination, all deviations to the shortest path are

considered and marked as alternative routes with the correspondent sidetrack *e* value. This value determines the difference between following the shortest path and deviating to that specific edge *e*, with respect to the length. The *k* shortest paths are then generated, based on each sidetrack value [23].

- **K*** is inspired in Eppstein, but was designed to perform on-the fly[1] while being guided by heuristic functions. Whereas Eppstein uses Dijkstra in a backwards manner, K* determines the shortest path tree by applying the A* algorithm in the forward sense, from a given source *s* to a destination *d*. The heuristic is based on $\delta(u, v)$, which represents the disadvantage of taking the edge between vertices *u* and *v* as the deviation from the shortest path. However, the shortest path and $\delta(u, v)$ are not known during the search. This problem is solved by applying concurrently Dijkstra to generate shortest path solutions before A* ends its execution. Decisions on-the-fly can only rely on an evaluation function *f* that determines an estimated value for $\delta(u, v)$ [21].

2.4.3 Performance assessment

Table 2 provides an estimation of the runtime execution complexity in the worst-case for every surveyed routing algorithm, where *n* is the number of vertices, *m* the number of edges and *k* the number of shortest paths in the case of the *k* shortest paths problem. It can be noted that A* solves the shortest path problem much faster than Dijkstra since it uses an heuristic function to estimate the node that will first lead to the destination [22]. On the contrary, Floyd is mathematically more complex than Dijkstra and A* because it has to compute the shortest route from every node to all the others. However, it only requires a single execution to find all-to-all shortest paths, whereas Dijkstra and A* require the analysis of almost all the nodes to compute a single path between two nodes. Floyd is then considered the most appropriate static path planning algorithm for complex traffic environments.

On the other hand, the *k* shortest paths problem introduces the variable *k* that influences the runtime complexity. Comparing K* and Eppstein, the most advantageous algorithm is Eppstein in terms of runtime complexity. The on-the-fly search of K* multiplies the complexity by a factor of *k* in the $n\log(n)$ portion [21]. On the contrary, the assessment made to k-pathA in [20] was based on a distributed network where nodes communicate with each other by message passing. Although they do not provide the runtime complexity of the algorithm, it can be concluded that the algorithm scales linearly in large networks but is computationally heavy to smaller ones. Thus, Eppstein is considered the faster algorithm providing *k* shortest paths.

2.5. Dynamic management of traffic environments

The dynamic, or time-dependent, management of traffic environments requires the ability to schedule different movements taking into account the current traffic situation. These movements are based on continuous routing decisions along the time. The dynamic nature of the problem implies a regular planning to deal with unexpected situations, such as deviations from assigned paths. This section considers two different models with dynamic features, capable of scheduling and coordinating concurrent movements within a traffic environment:

- **Petri Net** is a particular representation of the weighted graph, modeling the system with three different components: places, transitions and arcs. A place represents a discrete element, such as a taxiway segment in the airport movement surface. The transition, as the name implies, is the passage from one input place to an output place. The transition is fired when a set of conditions are satisfied. The arcs are the connecting links between places and transitions. The Colored Timed Petri Net (CTPN) is an improved Petri Net model, which provides a dynamic adaptation based on a set of constraints, where the state of the system is continuously verified in order to predict conflicts and

[1] On-the-fly represents an activity that has to be dynamically adapted to operational changes during its execution, without requiring the graph to be explicitly available on main memory [21].

Table 2: Routing algorithms performance.

Algorithms		Runtime complexity	Contribution
Shortest path algorithms	Dijkstra [17]	$\mathcal{O}(n^2)$	One-to-all shortest path
	A* [21]	$m + n\log(n)$	Dijkstra improved with an heuristic function
	Floyd [19].	$\mathcal{O}(n^3)$	All-to-all shortest path
k shortest paths algorithms	k-PathA [20]	–	Forward/backward chaining approach
	Eppstein [23]	$m + n\log(n) + k$	Reverse Dijkstra search and shortest path tree
	K* [21]	$m + kn\log(kn)$	On-the-fly search based on heuristic functions

provide a resolution. A constraint is defined as a condition that must be satisfied in order to fire the transition. For instance, an aircraft is not allowed to cross a transition until all the conditions are satisfied. In the case there is a conflict, the aircraft is retained a delay time that will assure the synchronization between all movements. Otherwise, the transition is fired and the token (i.e., aircraft) passes from the input place to the output place [24].

– **MILP** represents a traffic movement surface as a space-time network, where the occupancy of each node changes over the time. This model is divided in two main steps: first assigns an individual ideal route for each aircraft and then solves the conflicts caused by the uncoordinated paths. The first step allows to schedule, periodically, all movements along the time. Such movements may involve conflicts, namely when there is more than one target using the same link at the same time or if there are two targets crossing each other. So, the conflicting routes are re-routed to an alternative path to the same destination or a delay is applied to avoid such conflict [16].

In the Petri Net approach, the algorithm coordinates the movements by defining constraints that have to be validated whenever an operation takes place, whereas the MILP solution solves the conflicts by assigning alternatives routes or by defining a set of holding points. MILP is the preferred dynamic path-planning algorithm since it provides a continuous guidance to moving objects with a simple and accurate mechanism.

3. Methodological proposal

This section addresses the proposed methodology to implement levels III and IV of A-SMGCS. The main contribution consists at extending an existing A-SMGCS platform, named A-Guidance, that already provides real-time positioning of aircrafts and vehicles. The A-Guidance also includes an alert mechanism to inform decision-makers and vehicle drivers about safety incursions or business rules infringements. This system is experimentally deployed in two airports in Portugal, where the ground movements are represented in real-time on a GIS display to airport stakeholders in order to improve their ability to manage surface movements [3]. Nevertheless, the system does not provide routing and guidance functionalities. The proposed methodology is therefore focused at implementing such functionalities oriented to the guidance of aircrafts, within the Application Server of the LBS infrastructure presented in Section 2.1.

3.1 Architecture

The methodological architecture is presented in Figure 2. It is divided into three main components, namely Surveillance, Routing and Guidance that actively interact with the Application Server. A-SMGCS control functions related to guidance procedures were embedded in the guidance component, where guidance conflicts are detected and resolved. As shown, the routing component manages the input data to be able to generate the weighted graph and compute the routes in runtime. The required input data are mainly:

– Line segments extracted from the polygon shapes of the airport surface.

– Operational data with information about flight schedules.

– Business rules that mostly define the circulation rules.

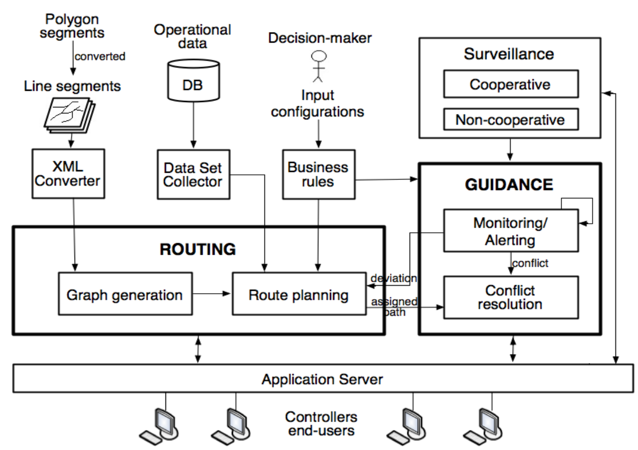

Figure 2: Methodological architecture.

The proposed routes are monitored by the Guidance component that is responsible for avoiding conflicts between aircrafts crossing the same taxiway intersection, a persecution or a frontal collision [25]. Such situations will automatically trigger an alert procedure that must be resolved by the controllers, which have a set of alternative options provided by the guidance function. For instance, when a pilot fails a runway exit, the system automatically recalculates an alternative path to the same destination (e.g., stand position).

Floyd was the implemented routing algorithm, since it provides the shortest path between every pair of nodes at any time without additional computation. This is particularly useful when there is a real-time requirement to compute routes from anywhere to everywhere within a very short period. Therefore, Floyd is the most powerful algorithm because, although it spends more time computing all paths at the beginning of the execution, it allows determining a path extremely fast during the execution.

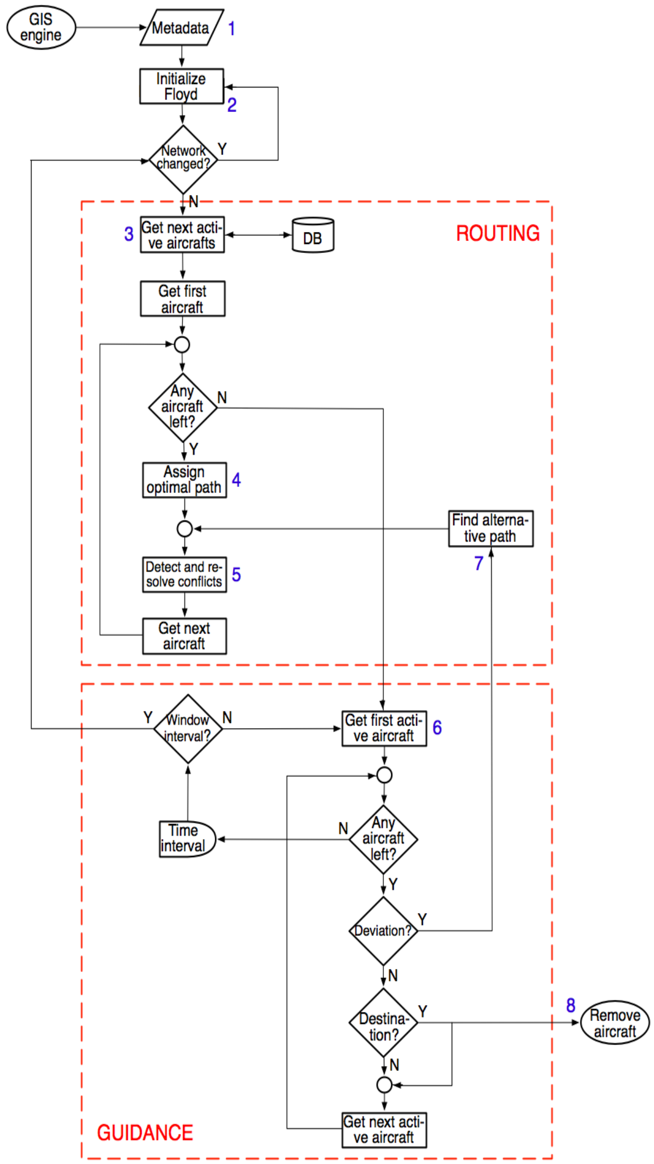

Figure 3: Proposed solution flowchart.

In the case there is a tight turn[2] in the generated route or a conflict between two moving objects, the system applies a mechanism to compute an alternative path to the same destination. It was inspired in K*, but uses the A* algorithm in the backward sense by applying an heuristic decision in each iteration. The decision consists in selecting the closest node to the origin, starting from the destination. This means that, among the linked nodes, the node that is less distant to the origin is chosen and belongs to the path. The process is repeated until the origin is reached, where the alternative path corresponds to the reverse route of the computed path. This mechanism is applied to the k nodes directly linked to the destination, generating k distinct alternative paths sorted by their respective length.

On the other hand, the guidance function was implemented based on the MILP model because it allows scheduling the surface movements in a very simple and effective way. Periodically, the guidance function assigns a path to each aircraft, while avoiding conflicts with the other movements. Then, it monitors the path followed by each pilot with the location data provided by the surveillance component and, if a conflict is detected, it reorganizes the predefined paths.

3.2 Implementation flowchart

The flowchart represented in Figure 3 highlights the main steps of the implementation of the proposed methodology, which are described below.

- **Step 1:** Extraction of the cartographic information from GIS as the input data to the generation of the weighted graph.

- **Step 2:** Generation of the adjacency matrix, based on the input metadata. Initialization of Floyd matrices with routing information.

- **Step 3:** Query to the flights database to request next active aircrafts.

- **Step 4:** Assignment of optimal paths to each active aircraft, without considering the other movements.

- **Step 5:** Detection and resolution of conflicts between movements scheduled in Step 4.

- **Step 6:** Get each active aircraft to monitor its movement along the assigned path.

- **Step 7:** An alternative route must be computed due to the detection of a deviation from the assigned path.

- **Step 8:** Destination was reached, the aircraft is removed from the system.

The process of assigning routes is repeated every window interval, which represents the periodic scheduling time (e.g. 1 minute). Within every window interval, the guidance function runs periodically at each time interval, which corresponds to the pace of the simulation (e.g. 1 second). Thus, the guidance function performs a continuous monitoring of active aircrafts movements, for instance every second, and the routing function assigns a route to every new active aircraft, for instance every minute.

[2] A tight turn cannot be performed by an aircraft.

4. Case study

The case study focuses on the airport domain, more precisely on the movement area that is composed by the manoeuvring area and a restricted area called apron. The manoeuvring area is used by aircrafts to take-off, land or travel along their path. This area is organized by a set of taxiways and roadways used by aircrafts and airport vehicles, respectively. The apron area is the operational area of the airport used for parking aircrafts, boarding passengers, and where most of ground handling activities occur. Aircrafts have to follow a route from the runway to the stand area (or vice versa) [1]. Thus, there is a need to guide the aircraft along the route, while considering the location of other moving objects (people, vehicles and other aircrafts). Such need requires the application of the methodology described in the previous section, in order to take advantage from an LBS infrastructure to provide routing and guidance functionalities. The case study operates with data collected from the A-Guidance system that is installed at Lisbon airport, extending its capabilities to cope with levels III and IV of the A-SMGCS requirements.

The A-Guidance software was developed by the INESC-Inovação (INOV) team, in collaboration with ANA-Aeroportos, the main airport management authority in Portugal. This system is deployed at Porto and Lisbon airports, where several experimental tests have been performed in order to validate the safety and functional requirements of an A-SMGCS implementation. The current implementation relies on a LBS infrastructure where vehicles are equipped with GPS/EGNOS receivers to transmit their position through a wireless network covering almost the entire movement area of the airport. Lisbon airport is equipped with a SMR enabling aircrafts to be detected.

An airport is a very complex and regulated environment, where there are a set of business rules defining, for instance, circulation rules and conflict resolution rules. On the one hand, the circulation rules comprising for instance speed limits, traffic circulation rules, together with traffic signalling. Another rule is that aircrafts cannot move backwards or perform a tight turn (e.g. with an angle below 100°), forcing to re-route the aircraft through alternatives [16]. Aircrafts must also maintain a safety distance from every obstacle, in order to avoid collisions. Conflict resolution requires the adoption of some rules related to aircrafts categories, the bigger the aircraft the higher the priority. In other words, the system must decide which aircraft has the highest priority. For instance, despite bigger aircrafts have circulation priority against smaller ones; departing aircrafts have circulation priority against any arriving aircraft. When a conflict takes place, the aircraft with lowest priority has two options: wait in a holding position until the other aircraft passes or find an alternative path to its target destination. At the current version the option that is less time consuming is selected.

5. Test and evaluation

The evaluation of the proposed methodology was based on a simulation environment applied to the Lisbon airport test bed, containing two runways and about 306 segments, including taxiways and stands. The routing and guidance functionalities were tested in this airport, with simulated flights and aircrafts. However, several tests were also performed with real operational data. The results are presented in this section, evaluating how the system performs and how the ground movements efficiency was improved.

5.1. A-SMGCS requirements

In order to evaluate the solution, the most relevant A-SMGCS requirements defined by ICAO were considered and are presented in Table 3 [2].

5.2. Simulation environment

A test scenario with 7 aircrafts tested the routing and guidance functionalities, where there are different categories and types of aircrafts taxiing at the same time. Table 4 presents the results obtained from this test, namely the

Table 3: A-SMGCS routing and guidance requirements [2].

Type	Requirements
Operational	• O1: Be able to compute a route for each authorized moving object within the movement area. • O2: Allow for a change of destination or route at any time. • O3: Minimize the length of computed paths. • O4: Minimize the conflicts. • O5: Provide guidance for every authorized moving object, for any assigned route. • O6: Provide clear instructions to pilots to help them following their path.
Performance	• P1: The initial route should be computed in less than 10 seconds. • P2: Recompute the path for a moving object should not exceed 1 second.

Table 4: Test scenario results.

Aircraft	Category	Type	Start time	Taxi distance	Wait time	Conflicts
5	Medium	Departing	0:00:00	1390m	0s	0
1	Small	Departing	0:00:10	1732m	18s	1
7	Large	Departing	0:00:10	2167m	0s	1
3	Medium	Departing	0:00:30	2562m	19s	2
4	Large	Departing	0:01:20	1492m	0s	2
8	Small	Arriving	0:01:30	1385m	4s	1
2	Small	Arriving	0:01:35	1457m	20s	1

taxi distance, the wait time and the number of conflicts. The taxi distance represents the length of the path followed by the aircraft and the wait time is the time spent in a holding point. The number of conflicts refers the number of conflicts detected and resolved. There are three categories of aircrafts size: small, medium and large. Also, there are two types of aircrafts: arriving and departing. As shown in Table 4, 6 aircrafts were involved in 4 conflicts (8 conflicts, 2 aircrafts for each). These conflicts forced the delay of 4 aircrafts, in order to avoid frontal collisions, persecutions and intersections. For instance, the main injured aircrafts were the aircrafts 1, 2 and 3 with a wait time of 18s, 20s and 19s respectively. On the contrary, aircrafts with higher priority (4, 5 and 7) were not affected. The system was then able to coordinate the movements efficiently, giving priority to the largest and departing aircrafts, while the smaller and arriving ones had to wait a few seconds.

Furthermore, the difference between optimal and alternative paths is represented in Figure 4, where the bold line is the shortest path from point A to point B and the dark line represents the alternative path computed by the routing algorithm. Other examples are presented in Figure 5, where the first aircraft is relative to the Figure 4. These values prove that some alternative paths are almost twice longer than the optimal paths, which is the case of the aircrafts 1, 6 and 7. On the contrary, for aircrafts 2, 3, 4 and 5, the routing algorithm was able to find little longer alternative paths. This is particularly important when considering a deviation from the assigned path caused by a pilot mistake, allowing finding alternative paths with almost the same length than the optimal one.

Finally, the performance of the routing function was evaluated with respect to the execution times, depending on the number of segments of the computed path, for the first iteration of the routing algorithm. In other words, the path computation complexity was evaluated by measuring the execution times for paths with different sizes (number of segments).

The results are represented in Figure 6, where the execution time increases moderately with the number of segments of the computed path. As shown, a path with 5 segments was computed in 20 ms. The same execution time was spent to compute a path with 15 segments and almost 20 segments. Even a path with 35 segments only spent 30 ms to be computed. So, it can be concluded that the algorithm is extremely fast and scalable, computing paths with 1 to 42 segments in less than 30 ms.

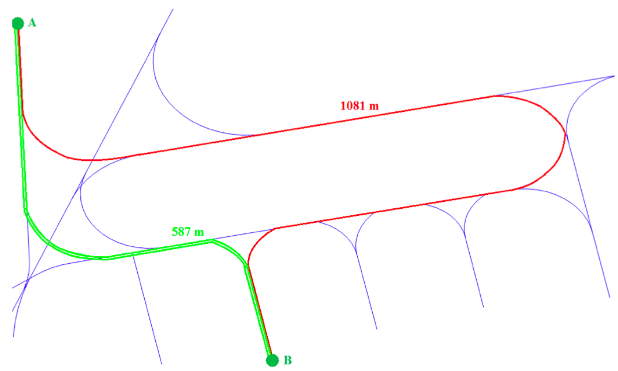

Figure 4: Optimal (bold) and alternative (dark) paths.

Aircraft	Taxi distance	
	Optimal	Alternative
1	587m	1081m
2	1454m	2223m
3	2113m	2562m
4	1020m	1134m
5	1952m	2002m
6	911m	1687m
7	545m	989m

Figure 5: Optimal and alternative paths length.

5.3. Operational environment

The operational environment comprises the real historical data collected from February 2011 in the Lisbon air-

Figure 6: Execution times for paths of different lengths.

port, including the geographic coordinates of the path followed by each pilot. This data contains several gaps caused by interferences or other communication difficulties, for instance positions out of the taxiway guidance lines. These gaps force to implement a function that determines the real sequence of segments followed by the pilot, in order to enhance the guidance capability to detect conflicts and provide guidance assistance to the pilots.

From the data collected at the airport, relative to one operational day, about 23% were profitable. This means that the path followed by the pilot was identified and correctly monitored. The data presented in Table 5 represents the situations where the current version of A-Guidance proposed a shorter path, comparing to the one followed by the pilot. For instance, the difference between the real data and the A-Guidance proposal is about 1099m for TAP741 and 477m for KLM1692, which would have saved several seconds in the taxiing time.

Table 5: Length of paths computed by A-Guidance, comparing to real movements in the Lisbon airport.

Aircraft	Taxi distance	
	Real data	A-Guidance
TAP741	2380m	1281m
TAP668	1625m	1601m
KLM1692	2069m	1592m
IBE31EW	1898m	1625m

5.4. Compliance with the A-SMGCS requirements

The test results proved that the operational A-SMGCS requirements were meet, since the optimal path (O3) was assigned to all aircrafts (O1), minimizing the conflicts with the other movements (O4). The pilots were also provided with an automated taxiway lighting system (O5) that turns on the lights relative to the segments n+1, meaning that a pilot is always guided by the lights of the current segment and the following one (O6). The guidance function also copes with path deviations, providing an alternative path in real time (O2).

On the other hand, the performance A-SMGCS requirements were also accomplished, since the routes are all computed within 30 ms (P1-P2). Furthermore, Floyd spends approximately 9 seconds to initialize the matrices with the routing information. However, a mechanism was implemented to store the matrices in a file, in order to speed up the process. Then, Floyd only spends 9 seconds when the airport layout suffers a modification that forces to recompute the matrices. Otherwise, it only spends 70 ms reading from the file.

6. Conclusions and future work

The business case used to successfully test the proposed methodology was based on a LBS infrastructure supported by the A-Guidance system that copes with level I and level II of the A-SMGCS. The paper described how the methodology enabled the extension of the A-Guidance capabilities to provide routing and guidance functionalities. In fact, the integration of location-based data collected from wireless technologies with performing embedded systems enabled, for instance, to provide the required spatio-temporal context to the controllers. Such information improved the controllers situational awareness, providing them with decision-support mechanisms to enhance the management of traffic ground movements more efficiently.

The results comply with the operational and performance A-SMGCS requirements, enabling the routing function to assign a path to every aircraft while minimizing the taxi distance as well as minimizing the risk of conflicts with other moving objects. Furthermore, the guidance function copes with dynamic changes to routing, enabling, for instance, to interact with the lighting system to guide the pilot from the current position to the assigned stand or runway. On the other hand, the performance goals were achieved since the routes are computed extremely fast and the system responds accurately to emergency situations.

The future work focuses on extending the proposed methodology to provide guidance assistance to airport vehicles. In this case the routing algorithm becomes more complex because vehicle drivers can have unpredictable driving behaviors. Such behaviors are extremely difficult to predict and introduces additional variables to the algorithm that might compromise its performance. Research is being done to respond to those challenges, namely to consider both forward and backward vehicle movements, tight turns, traffic rules and circulation priorities based on the type of vehicles as well as between vehicles and aircrafts.

References

[1] EUROCONTROL, *Definition of A-SMGCS implementation levels*, Edition 1.2, 2010.

[2] ICAO, *Advanced Surface Movement Guidance and Control Systems (A-SMGCS) Manual*, Doc 9830, 2004.

[3] Casaca, A., et al., *A platform to increase the safety of ground movements in the airside area of airports*, in P. Larauge, Airports: Performance, Risks, and Problems, chapter 1, pp. 1-31, 2009.

[4] EMMA Project Partners, *State of the art in A-SMGCS*, Doc. D1.1.1 Version 1.0, 2005.

[5] Rees M., *The EUROCONTROL surveillance strategy*, Digital Communications - Enhanced Surveillance of Aircraft and Vehicles, Tyrrhenian International Workshop, 2008.

[6] Steiniger S., Neun M., Edwardes A., *Foundations of Location Based Services*, Project CartouCHe, Lecture Notes on LBS, Lesson 1, V. 1.0, Department of Geography, University of Zurich, 2006.

[7] S. Steiniger, M. Neun, A. Edwardes, *Foundations of Location Based Services*, Project CartouCHe, lecture notes on LBS, lesson 1, V. 1.0, Department of Geography, University of Zürich, 2006.

[8] Qi L., *Research on Intelligent Transportation System technologies and applications*, Power Electronics and Intelligent Transportation System (PEITS 08), 2008.

[9] Gao H., et al., *Safe airport operation based on innovative magnetic detector system*, Intelligent Transport Systems (IET), vol.3, no.2, pp. 236-244, 2009.

[10] Besada J., et al., *Image-based automatic surveillance for airport surface*, 4th International Conference on Information Fusion, Montreal, 2001.

[11] Lukin K., et al., *Novel concepts for Surface Movement Radar design*, Digital Communications - Enhanced Surveillance of Aircraft and Vehicles, Tyrrhenian International Workshop, 2008.

[12] Muthukrishnan K., M. Lijding and P. Havinga, *Towards smart surroundings: enabling techniques and technologies for localization*, Location- and Context-Awareness, Lecture Notes in Computer Science, Springer Berlin, vol. 3479, pp. 350-362, 2005.

[13] Roxin A., et al., *Survey of wireless geolocation techniques*, IEEE Globecom Workshops, 2007.

[14] Australian Government, *ADS-B Automatic Dependent Surveillance Broadcast*, Civil Aviation Safety Authority, 2006.

[15] Miyazaki H., *Evaluation results of multilateration at Narita International Airport*, 13th IAIN World Congress and Exhibition, Stockholm, 2009.

[16] Roling P., Visser H., *Optimal airport surface traffic planning using mixed-integer linear programming*, International Journal of Aerospace Engineering, 2008.

[17] Chao Y., Hongxia W., *Developed Dijkstra shortest path search algorithm and simulation*, Computer Design and Applications (ICCDA), International Conference, vol.1, pp. V1- 116-V1-119, 2010.

[18] Yao J., et al., *Path planning for virtual human motion using improved A* algorithm*, Information Technology: New Generations (ITNG), 7th International Conference, 2010.

[19] Wan J., et al., *Route planning based on Floyd algorithm for Intelligence Transportation System*, Integration Technology (ICIT '07), IEEE International Conference, 2007.

[20] A. Ullrich, C. V. Forst, *k-PathA: k-shortest path algorithm*, International Workshop on High Performance Computational Systems Biology, HIBI '09, 2009.

[21] H. Aljazzar, S. Leue, *K*: A directed on-the-fly algorithm for finding the k shortest paths*, Technical report soft-08-03, University of Konstanz, Chair for Software Engineering, 2008.

[22] Q. Wu, *Incremental routing algorithms for dynamic transportation networks*, M.Sc Thesis, Calgary University, Calgary, Alberta, Canada, 2006.

[23] D. Eppstein, *Finding the k shortest paths* SIAM J. Computing, vol. 28, no. 2, pp. 652-673, 1998.

[24] Yu-ting W., et al., *Research on airport surface modeling and path planning algorithm based on CTPN*, Intelligent Computing and Intelligent Systems (ICIS 2009), vol.3, pp.262-266, 2009.

[25] Li Z., Zhang J., Zhang X., *A dynamic model for aircraft route optimizing in airport surface management*, Electronic Measurement and Instruments (ICEMI 09), 9th International Conference, 2009.