Aircraft Landing Planning: Past, Present and Future

Marie-Sklaerder Vié¹, Nicolas Zufferey¹, Roel Leus²

¹ Geneva School of Economics and Management, GSEM - University of Geneva, Switzerland
{marie-sklaerder.vie,n.zufferey}@unige.ch
² Faculty of Economics and Business, KU Leuven, Belgium, roel.leus@kuleuven.be

Key-words: Aircraft Landing Planning, Uncertainty, Scheduling, Optimization

Abstract: Aircraft Landing Planning is a key issue for airports, as the runway capacity is their main limitation. In this paper, solution methods proposed in the literature are discussed, as well as current arrival traffic flow management practices. Based on this, the main missing features are identified and allow us designing the most promising directions for future research.

1 Introduction

From take-off to landing, planes are subjects to many uncertainties (e.g., wind, traffic). Moreover, the runway capacity of an airport is the bottleneck capacity of the landing process. Therefore, the landing time of each plane arriving at an airport has to be adjusted (e.g., delay) all along its flight trajectory to use the runway capacity at best despite uncertainty. One way to delay an airplane is to make it perform holding stack patterns or vectoring in the vicinity of the airport (approach area) before its landing, but this creates bigger fuel consumption and noise pollution. Another way is to decrease (sometimes increase) the speed of the plane during its cruise (or make it do a detour using path stretching), but due to larger uncertainties at longer distance from destination, this may create unnecessary additional delays. Also, the fewer number of modifications there are on a landing time, the better it is for the air traffic controllers’ (ATC) workload, as they have other priorities (mainly ensuring safety and avoiding collisions).

The goal of Aircraft Landing Planning (ALP) is to optimize the landing times of the planes arriving at an airport runway. For this purpose, the planes to delay have to be selected, as well as how and when to delay them. Safety constraints imposing threshold distances between planes have to be satisfied, obviously.

ALP is considered here without the Aircraft Take-off Problem. Indeed, the current time-horizon of ALP is about one hour prior to landing, hence most flights are already airborne, except the so-called pop-up flights (which represents a minor part of the flights). In addition, as shown in [47], pre-planning (if the accuracy of departure time is of approximately five minutes, which is a very small uncertainty in practice) that takes into account pop-up flights gives an overall performance that is worse than if those flights are only considered once airborne. The reader is referred to [14, 30] for a literature survey and general considerations on scheduling under uncertainty.

This study is partially financed by EUROCONTROL the European Organization for the Safety of Air Navigation through the SESAR2020 programme. In the view of constructing a unified European sky, a decision-making framework developed for EUROCONTROL is presented in [24, 25]. It considers multiple objectives (i.e., capacity, safety, cost-effectiveness, flexibility, environment), and potential disagreements among stakeholder regarding the impact of the proposed system enhancements. Current typical tactical arrival management procedures are explained in Section 2. A review on the solution methods proposed for ALP is presented in Section 3 up to year 2011, relying on an existing literature review proposed in [12]. The contributions of this work are the following. First, at the end of Section 3, we highlight the
weaknesses of the works up to year 2011 according to the following criteria: instance size, computing time, integration of random events. Based on that, in Section 4, we bridge the gap of the literature review from 2011 up to now, but with a focus on the studies that are able to account for dynamic features. Finally, with respect to relevance and impact in practice, and in contrast with most of the existing publications, promising problems and future directions are identified in Section 5. A conclusion ends the paper.

2 Current typical arrival management practices

ALP can be divided into different stages. First, an initial schedule is created prior to departure, then it is modified during the cruise, and finally it is frozen (landing phase). The initial schedule is usually based on a First-Come First-Serve (FCFS) sequence [21]: planes are sorted by increasing estimated landing times and scheduled accordingly. This sequence involves all the planes that are in the planner’s range (about 30-40 minutes before landing), and it is updated each time a new plane enters this range. However, it is showed in [16] that FCFS usually does not lead to optimal sequences with respect to throughput or average delay.

Each flight can be divided into two phases within the Arrival Management (AMAN) horizon: (1) cruise (from the beginning of the AMAN planning horizon to 10-15 minutes before landing); (2) approach toward the airport (the last 10-15 minutes in non-congested conditions). Two typical approaches are used to manage the arrival flows: the buffer approach (used for instance in London) allows the largest modifications mainly during phase (2), the updating approach (employed for example in Paris) allows modifications during both phases. Modifications during phase (1) consists in performing speed adjustments and/or implementing detours on the planned trajectory. These changes are made en-route by an air traffic controller (ATC), following advisories provided by an AMAN tool. Regarding the possible modifications during phase (2), each flight arrives as expected in the approach area of the airport, and then can be delayed, as required to keep safe distances and a good runway throughput by using different means like holding patterns (“stacks”), vectoring (“detour”), point-merge or a combination of these means. In this paper, the updating approach is mainly investigated as it better captures the focus of extended AMAN operations, solution being validated in the SESAR 2020 program.

Any modification must keep the landing time of an aircraft between its earliest and latest possible landing times, which depend on technical and operational constraints (e.g., possible speed, fuel limitation, connecting flights). Also, ATCs may limit the shifting of a plane, within the FCFS sequence, to a maximum number of positions (CPS: Constrained Position Shifting [18, 46]), or to a maximum time (CTS: Constrained Time Shifting [35]), for not deviating too much from the initial schedule.

3 Summary and analysis of the literature review up to 2011

A comprehensive review covering the period up to 2011 can be found in [12], and it is discussed in this section. The main objective functions can be divided into four categories, all well-represented in the literature.

(F1) Safety and efficiency objectives. Maximize: runway throughput, fairness among the aircrafts. Minimize: approach time before landing, ATCs’ workload, aircraft taxi-out time, arrival delays.

(F2) Airline’s objectives. Maximize: punctuality with respect to landing times in published timetables, adherence to airline priorities within their own flights, connectivity between flights. Minimize: operation costs (mainly fuel costs), total passenger delays.

(F3) Airport objectives. Maximize: punctuality according to the operating schedule. Minimize: the need for gate changes due to delays.

(F4) Government objectives. Minimize: environmental effects (i.e., noise and air pollution).
Below, we first discuss the main exact approaches that were adapted to ALP. Next, we discuss the main proposed solution methods (mostly heuristics and metaheuristics). Finally, we highlight the main weaknesses of these works.

Dynamic programming was widely used for minimizing the sum of landing costs of an arriving schedule, which is equivalent to minimizing landing times first, and fuel cost second (i.e., $F_1$ and $F_2$ objectives), under various constraints (precedence, time-window) [3, 4, 17, 34]. A dynamic programming algorithm for maximizing the smallest spacing between planes upon landing (a $F_1$-objective) is proposed in [6]. Branch-and-bound is a common algorithm to solve ALP. In [1], it minimizes the speed adjustments of the aircraft (a $F_1$-objective). In [8, 20], it minimizes deviations from target landing times ($F_2$ and $F_3$ objectives). A branch-and-cut algorithm maximizing the smallest time between two consecutive landings (a $F_1$-objective) is given in [2], with a time-window constraint for each landing. With the same objective function, a branch-and-price exact algorithm with the use of column generation is developed in [48].

Some heuristics were designed for ALP, such as descent local search: in [19] with a permutation neighborhood (the position of two consecutive planes in the landing sequence are permuted) within a CPS-constrained environment, or in [42] with the well-known neighborhood structures shift and swap. Among metaheuristics, one can cite Multi-objective Neighborhood Search Differential Evolution [43] or ant colony optimization [11]. However, genetic algorithms are the most popular [10, 16, 31, 32, 38, 49]. Finally, ALP can be modeled as a queuing system with the incoming aircraft being the customer [5], and approximation algorithms for minimizing the sum of all landing times and the last landing time were also proposed [7].

On the one hand, exact methods consider the static case, involving $n$ planes about to land within their time-windows, but without any planning horizon. They can be used for instances with up to 50 aircrafts, often allowing shifting the position of an aircraft by at most three ranks (with respect to the initially planned sequence). The computing times for these instances usually range from 10 to 60 minutes, which is too long considering that an airport can have a plane landing/taking-off per minute. The only exact methods that run in reasonable time are those with CPS between 1 and 3 positions, which is very low in a dynamic case when a plane can often encounter some minutes of delay. On the other hand, the vast majority of (meta)heuristics also tackle the static case, and some of them can consider up to 500 planes. However, most of these methods only solve the problem up to 50 planes (within seconds). (Meta)heuristics have unreasonable computing times (i.e., more than 10 minutes) above 50 planes, except if a 4-CPS constraint [31, 32] is imposed. Unfortunately, such a limitation is too strong in practice. Although some metaheuristics have reasonable computing times for small instances, the robustness of the provided solutions is never analyzed. This is a major drawback in a dynamic context, as a random event might lead to an updated solution (using the metaheuristic on hand) that has a totally different structure. This is obviously not acceptable from a practical standpoint.

The review of [12] leads to different observations. First, that many theoretical studies show a great potential improvement of the runway utilization but may not be feasible in practice, due to ignored operational constraints (e.g., minimum time before landing, precedence constraints) or unreasonable computing times. Indeed, ATCs will always prefer fast (i.e., able to generate a solution within seconds) and satisfying (with respect to the considered objective) solution methods rather than optimal but time-consuming ones. Second, the definition of the objective functions and constraints vary a lot among the articles. Indeed, the involved parties (e.g., airport, companies, customers) have conflicting interests. Therefore, selecting an appropriate and realistic objective function is a critical issue. Third, and most importantly, uncertainty occurs at different levels (e.g., weather, precision of the equipment, departure time, accuracy of information). However, very few studies take this aspect into account and actually use simulation, as [9, 36]. Indeed, most of the literature considers the static case rather than a dynamic (but realistic) environment. Interestingly, it is showed in [15] that: deterministic
algorithms are sub-optimal in a dynamic environment; a FCFS method is robust (i.e., not too sensitive to variations in problem characteristics or data quality) and it has many advantages over many existing algorithms (e.g., generated sequence understandable by ATCs, stable and easy-to-estimate delays, already in place in many airports).

4 Recent works (2011 - now) integrating dynamic features

In recent years, some work on static ALP has still been published [23, 44]. But as highlighted above, from a practical standpoint, only the approaches that are able to take into account random events are relevant. For this reason, this section focuses on the scarce literature that integrates this important feature. The three main axes are queuing theory, robust optimization, and on-line optimization. The approaches including uncertainty use Monte-Carlo simulation, and some dynamic solution methods (but without uncertainty, except the appearance of new flights) employ a rolling-horizon approach.

Queuing systems for ALP consist in a stochastic model of the waiting line of planes about to land [27]. It uses the concept of Controlled Time of Arrivals relying on a traffic window (ranging from the expected arrival time \( t \) to \( t \) plus a tolerance value, which is zero if congestion) assigned to each aircraft along its trajectory (even if it sometimes misses it). Each traffic window is computed prior to departure and updated during the flight, in order to maximize the runway utilization. With queuing theory, the propagated delay triggered by a late plane can be quantified [26, 37]. This leads to the observation that when queuing delays are absorbed in high altitudes, fuel burn is minimized for individual flights. But due to trajectory prediction errors, there is a risk that lost landing slots propagate back to the remaining aircrafts, which increases the total delay, and hence the total fuel burnt. This means that queuing delays have to be distributed between high and low altitudes in order to find a fuel minimizing trade-off strategy.

A different approach consists in making static but robust algorithms. For example, a specific optimization method for the pre-tactical planning phase (i.e., between 30 minutes and several hours from departure) that reduces complexity by omitting unnecessary information is developed in [29]. More precisely, instead of determining arrival/departure times to the minute in this phase, several aircraft are assigned to the same time window of a given size. Mathematically, this corresponds to a generalized assignment problem on a bipartite graph. Next, the impact of uncertainties on these deterministic solutions is investigated, which leads to deciding to include these uncertainties or not (in the model). It is done in [28] by incorporating a predefined uncertainty set of variables into the model. This additional set leads to adding side constraints to the problem, in order to protect the model against data uncertainty, and hence it guarantees some robustness. The goal of robust plans is to reduce rescheduling when changes appear due to uncertainty in the planning, but in counterpart, it usually results in an efficiency reduction. Generally speaking, stochastic optimization aims to provide good solutions on average, whereas robust optimization immunizes against worst-case scenarios (often not the single worst-case scenario among all the options, but the worst scenario having at least a certain probability to occur). Therefore, robust solutions may give worse solutions, on average, than stochastic optimization. But, as it appears, the makespan (i.e., the arrival time of the last plane) of the robust solution is equal or slightly better, when compared to the solution of the nominal case (i.e., the model without the side constraints involving the uncertainty set of variables). This is due to the fact that the robust solution reduces the number of planning changes (e.g., detours) significantly, as it reacts better to uncertainty.

Another way to switch from the static to the dynamic case is to add on-line optimization to a static method. For instance, a multi-objective formulation of the problem is proposed in [13], taking into account runway throughput, earliness and tardiness, and fuel cost arising for maneuvers and additional flight time. In addition to the classical FCFS heuristic, the static method is solved with dynamic programming, iterated descent and simulated annealing.
A rolling horizon approach is used, where the dynamic problem is tackled by periodically updating the previous schedule with an iterated descent or a dynamic programming approach. Improvements (versus FCFS) are encountered when updating the schedule every five minutes. However, uncertainties are not taken into account in this method, except the new approaching planes.

5 Promising problems and future directions of the field

In the light of the weaknesses of the existing literature, various promising research directions are proposed below.

(1) Consideration of uncertainty. First and obviously, as most variables in ALP are stochastic, a quick, accurate and dedicated simulation tool is mandatory to evaluate the true quality of a solution. The Monte-Carlo simulation that is usually proposed for ALP is probably not the best tool according to these criteria. Generally, the existing literature does not take uncertainty into account. A scarce literature proposes either robust solution within a static approach, or a dynamic solution method but without any robustness guarantee. An interesting approach could integrate robustness in the on-line process (to avoid a prohibitive number of rescheduling actions) [33]. This can be done on the one hand by providing a robust initial solution with static optimization, and on the other hand by performing robust changes in the on-line optimization. As in the inventory management literature [41], a relevant way to do that could be the design and use of a robust estimator, which could approximate, analytically, the quality of a solution (i.e., without any simulation, which is the most time-consuming feature), while taking uncertainty into account.

(2) Appropriate solution methods. The existing metaheuristics (generally genetic and ant colony algorithms) do not seem the most appropriate for ALP. Indeed, they need a long learning phase, and even though they can provide feasible solutions at any time in the computation, there is little chance that the quality of the solutions is good after only a few seconds of computation. Surprisingly, tabu search was not proposed. However, it presents good characteristics for ALP as it allows incremental computation (i.e., only the variation of the objective function is computed), restricted neighborhoods (i.e., not all the neighbor solutions have to be evaluated in each iteration), and an efficient but simple mechanism to escape from local optima. In addition, it was showed in [50] that it has a good tradeoff between quality, speed and robustness, for various problem structures.

(3) Lexicographic optimization. All the solution methods listed above are limited to either a single objective which does not cover the richness of ALP, or a multi-objective approach which opens the door to conflicting discussions on the importance of some objectives versus others (and this discussion usually occurs after the optimization phase). In line with the production-scheduling literature [45] or with some supply-chain-management studies [39], a relevant approach would be the use of lexicographic optimization, for which the sensitive discussions on the priority of objectives happen before triggering the optimization phase. For instance, the following ranking of objectives (to be minimized) can cover various dimensions of ALP: (1) delays; (2) number of rescheduling actions; (3) fuel cost; (4) noise pollution. Indeed, in such a case, all dimensions $F_1$ to $F_4$ are fully covered. An important advantage of this lexicographic approach is that if is there is not enough computing time to deal with all three objectives, at least the first has a chance to be efficiently tackled.

(4) Instance calibration. An instance has to be designed in a realistic way. In most of the literature, instances with hundreds of flights are tackled (with metaheuristics), as well as instances with around 50 flights (with exact methods or metaheuristics). However, everyday and for many important European airports, the demand follows patterns with marked peaks (e.g., hub periods of three hours) and then lower traffic levels [22]. For these reasons, a relevant instance can be defined by a time horizon of three hours (i.e., covering a peak) and a single
landing runway. Knowing that the time between two consecutive aircraft belongs to interval [90, 120] seconds, the most relevant instance size ranges from 90 to 120 flights. Surprisingly, this range is usually not considered in the literature.

(5) Planning horizon. In a dynamic case, an important optimization would be the size of the considered rolling horizon $H$, as it can have a crucial impact on the quality of the obtained solution [40]. A tradeoff has to be found between (1) waiting as much as possible before delaying a flight (it minimizes the uncertainties, but increases the noise pollution and fuel costs), and (2) anticipating and delaying a flight as early as possible (more risky approach if too many uncertainties, but more efficient otherwise). The challenge is that the larger $H$ is, the more flexibility we have in the optimization, but the more uncertainty we have too (mainly because of pop-up flights).

6 Conclusion

A lot of work has been done on Aircraft Landing Planning (ALP), from exact methods (mainly branch-and-bounds) to metaheuristics (e.g., numerous genetic algorithms). These methods, however, entail two main weaknesses. First, they are usually static (i.e., assume that all data is well-known), whereas the problem presents many uncertainties (e.g., weather, traffic, interaction with other flights). Some recent articles try to tackle this issue, but there is still a big gap to bridge in order to properly include these uncertainties. Second, the existing approaches generally do not match the quickness of the decision environment in which the decision makers have to work. Moreover, the number of changes resulting from uncertainties should remain low, otherwise the workload of the Air Traffic Controllers will be increased too much. Instead of these static and theoretical solution methods, future work should propose realistic and dynamic solution methods, with a short computing time, and a strong effort on robustness, as uncertainties (which can be quantified) must be taken into account. These future dynamic algorithms should investigate the optimal length of the considered time horizon, with the help of fine-tuned simulation tools, on properly calibrated instances. Last but not least, lexicographic optimization is recommended for covering the needs of the various involved stakeholder (e.g., airlines, airports, governments). Overall, the above-identified research avenues are likely to lead to an augmentation of the airports capacities and to a reduced environmental footprint. It would be even more efficient if ALP is integrated with the Take-Off Problem, or with the Route Scheduling Problem of aircraft between runway and taxiway.

Acknowledgements

The authors would like to thank Mr. Raphaël Christien, from EUROCONTROL, for his availability and advices.

References


