Routing and Scheduling Algorithm for Aircraft Ground Movement Optimization

Pushkar J. Godbole*, Abhiram G. Ranade †, Rajkumar S. Pant ‡
Indian Institute of Technology - Bombay, Powai, Mumbai, 400076, India

Extended Abstract

Due to steep increase in air-transportation demand, heavy air traffic has become very common at many airports across the world while airport facilities and operation methodologies broadly remain constant. This has led to heavy congestions at airports thus increasing ground delays and taxi times. There is therefore a need to develop a decision support tool for efficient scheduling of aircraft, particularly at busy airports.

Optimal aircraft scheduling is a well known NP-Hard problem and hence heuristics are often used to generate schedules within realistic run-times. These heuristics are designed to run fast, but often do not promise any guarantee about the solution quality. A metric to evaluate and benchmark various scheduling algorithms for their *nearness to optimality* is thus necessary. The first part of the work reported in this extended abstract has been the development of an exhaustive search algorithm that finds a *provably* best solution to a snapshot problem. The benchmarking results of two different algorithms, one by Roling et al. [3] and another by Baijal et al [1] on the CSIA, Mumbai Airport have been illustrated. Early identification of global optimum by the exhaustive search algorithm shows promise that appropriate modification to it could deliver a real-time global optimizer. The second part of the work (ongoing) is to develop such a heuristic to generate *testable* optimal or near optimal schedules within short time.

I. Introduction

The throughput of an airport in terms of the the number of aircraft handled, depends on the airport's infrastructure capacity and operational efficiency. Due to heavy increase in air transportation, many of the airports today work at the brink of their infrastructural capacity thus making efficient operations a significant determining factor.

Although substantially efficient, the process of Air Traffic Control (ATC) remains majorly manual and empirical in nature. The downside of this is particularly observed in heavy traffic hours which make the manual decision making significantly difficult to handle, leading to myopic solutions. Additionally, the increasing workload on the ATC leads to suboptimal utilization of resources especially in critical situations such as low visibility and climatic variation. Significant research and development both at commercial and academic level is being undertaken to deal with the issue. The introduction of tools such as Airport Collaborative Decision Making (ACDM) [8] for efficient

*Department of Aerospace Engineering, IIT Bombay, Email: pushkar.godbole@iitb.ac.in, AIAA non-member
†Department of Computer science and Engineering, IIT Bombay, Email: ranade@cse.iitb.ac.in, AIAA non-member
‡Department of Aerospace Engineering, IIT Bombay, Email: rkpant@aero.iitb.ac.in, AIAA member

assimilation and dissemination of data has provided a platform for improved decision making across all partners. Tools such as Advanced Surface Movement Guidance and Control System (ASMGCS) [4] by EUROCONTROL are being developed for better surveillance and automated decision making and scheduling.

Aircraft scheduling poses itself as an NP-Hard problem with an enormous solution space. This makes deterministically finding a globally optimal solution within an acceptable run-time difficult. Hence various stochastic and semi-deterministic algorithms have been developed and investigated to tackle the problem in a realistic run-time. Smeltink et al. [10] use a Mixed Integer Linear Programming (MILP) model to generate a locally optimal schedule on a predetermined path allocation for aircraft. They use the concept of *Rolling Horizons* to subdivide the problem into briefly solvable sub-problems. The disadvantage of this algorithm is that it uses fixed paths and re-routing in case of conflicts is not possible. Baik et al. [5] use Time Dependent Shortest Path (TDSP) to schedule each aircraft individually. The aircraft are ordered based on a predetermined priority and scheduled one by one. The one by one scheduling leads to myopic schedules just as in case of manual scheduling. Gupta et al. [9] use a combination of the two methods by Smeltink and Baik. In this approach, TDSP is used for routing of the aircraft which is then fed to the MILP model similar to that by Smeltink. Roling et al. [3] also use MILP similar to Smeltink for scheduling although rerouting of aircraft is possible in this model.

Amongst the stochastic approaches, Gottland et al. [2] use a combination of Genetic algorithm and A* search to generate the schedule. The Genetic algorithm is used for aircraft prioritization and path allocation. A* is used to compute the fitness function of the Genetic algorithm while resolving conflicts. Baijal et al. [1] use Bacterial Foraging on a fixed set of path allocations generated using Dijkstra's shortest path algorithm to resolve conflicts and generate the schedule.

All the algorithms use the total taxi time including the waiting time as the cost function to be minimized. As can be observed, different strategies have been used to determine an optimal schedule although none ensures global optimality due to the NP-Hard nature of the problem. A metric however is necessary to evaluate various approaches quantitatively. A global optimum is thus needed to benchmark a particular solution generated by an algorithm to establish its *nearness to optimality*.

II. Present work

In spite of being NP-Hard, the limited size of the airport map (interpreted as a graph) ensures that the aircraft scheduling problem can be solved deterministically in a (possibly long but) finite run time. An exhaustive search algorithm that ensures global optimality has hence been developed using a combination of path search and Mixed Integer Linear Programming (MILP). The starting node, destination node, starting time and operational parameters namely speed, trailing separation, priority for each aircraft are given as input to the algorithm. The path search algorithm finds all simple paths for each aircraft. All flight-path combinations in order of increasing path lengths are generated and fed to the scheduler. The scheduler uses an MILP model to generate a conflict free optimal schedule for every flight-path combination and finally determines the global optimum. The total taxi time including the waiting time over all aircraft is used as a metric to be minimized.

The global search algorithm has been developed as a *post-facto analysis tool* to evaluate the performance of other algorithms and deduce the quality of their solutions. The algorithm has been implemented in C++ and CPLEX [7] has been used as the MILP solver. The visualizations are implemented in Python. An illustration of two problems has been shown here. The results have been generated on an Intel Xeon Quad core processor.

Test case problem by Roling et al [3]:

The problem, map and initial conditions have been taken from the test case problem by Roling et al. [3].

Flight	Origin	Destination	Start time	Speed	Trailing sep	Priority
1	26	15	7	1	4	1
2	24	15	6	2	4	1
3	25	6	10	2	4	1
4	25	6	8	1	4	1
5	25	6	16	2	4	1
6	24	6	14	1	4	1
7	28	26	0	1	4	1
8	28	26	3	1	4	1

Table 1: Test case problem by Roling etal.: Initial conditions

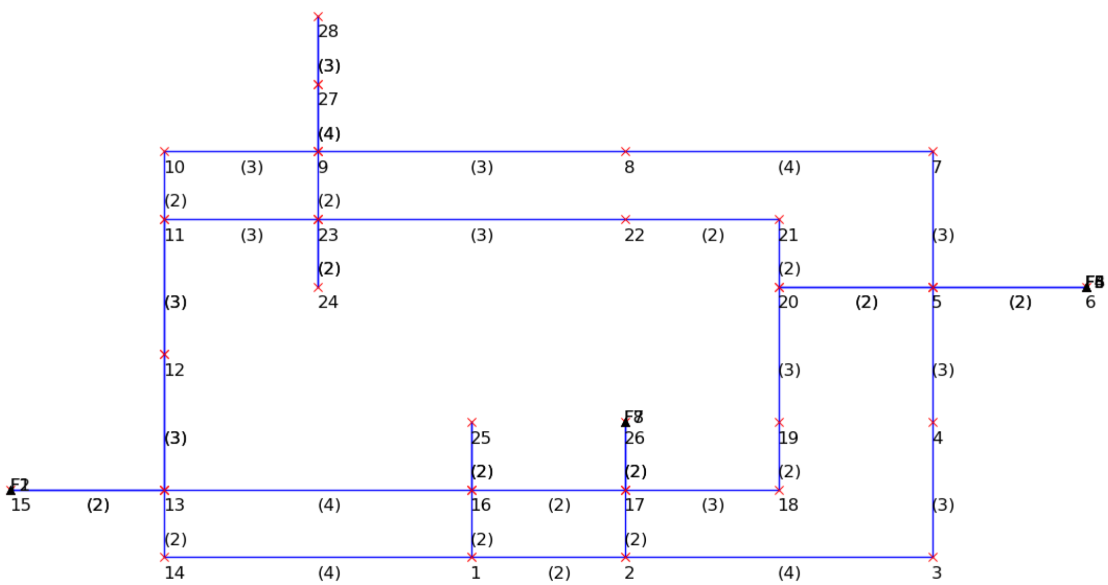

Figure 1: Test case problem by Roling etal.: Map

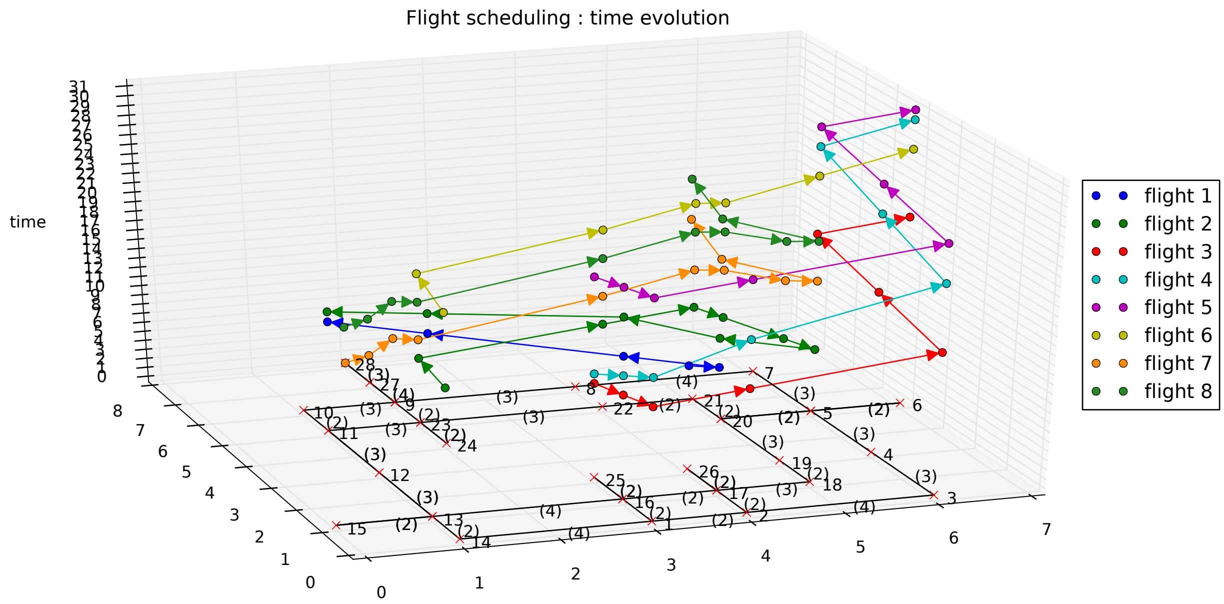

Figure 2: Time evolution of global optimum for test case problem by Roling et al. (X-Y plane represents airport map, Z axis represents time slots)

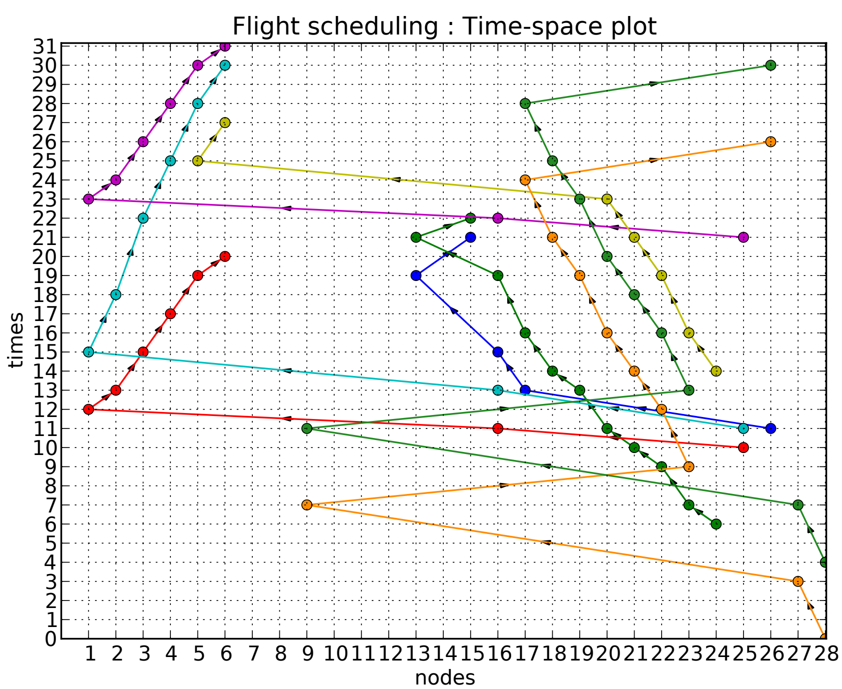

Figure 3: Time-space plot of global optimum for test case problem by Roling et al. (X axis represents nodes and Y axis represents time slots)

Roling test case solution cost	Global optimum	No. of MILPs solved	Optimum reached	Runtime	Roling solution excess cost
362	207	7776	4 mins (16%)	39 mins	75%

Test case problem on CSIA, Mumbai Airport map by Baijal et al.[1]:

The problem, map and initial conditions have been taken from the Mumbai Airport test case problem by Baijal et al. [1].

Flight	Origin	Destination	Start time	Speed	Trailing sep	Priority
1	1	8	13	1	3	1
2	1	13	12	1	3	1
3	5	12	6	1	3	1
4	9	2	6	1	3	1
5	1	15	0	1	3	1
6	15	1	0	1	3	1
7	1	8	0	1	3	1

Table 2: Mumbai Airport test case problem by Baijal etal.: Initial conditions

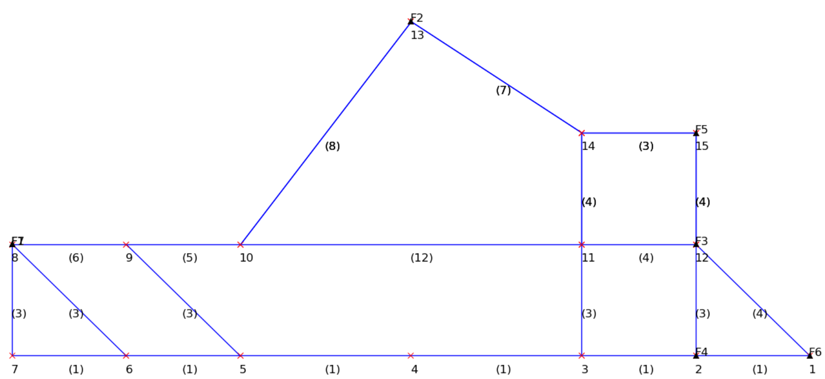

Figure 4: Mumbai Airport test case problem by Baijal etal.: Map

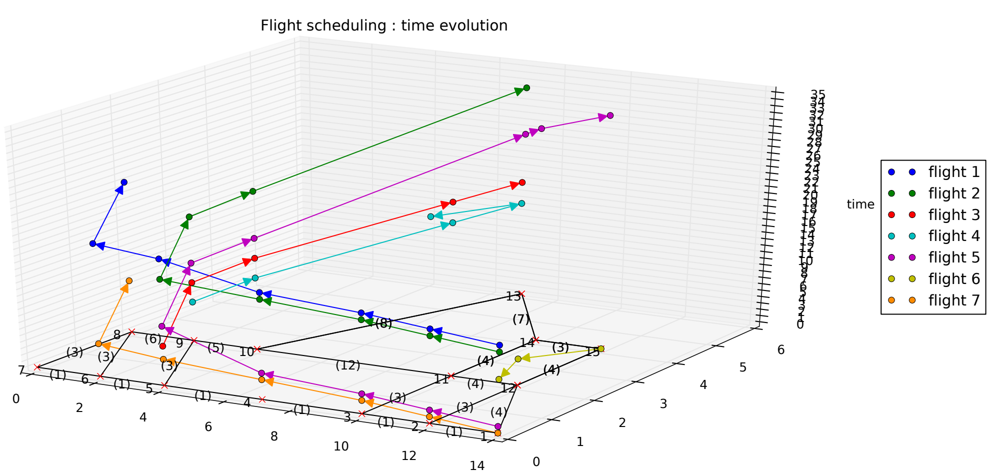

Figure 5: Time evolution of global optimum for the Mumbai Airport test case problem by Baijal etal. (X-Y plane represents airport map, Z axis represents time slots)

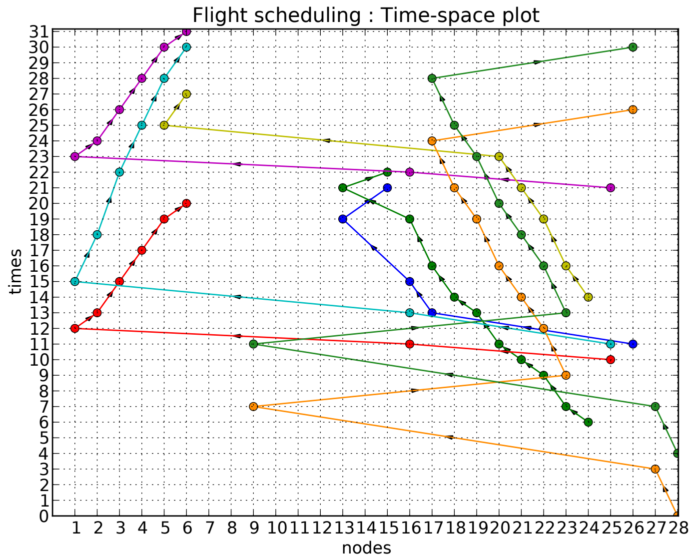

Figure 6: Time-space plot of global optimum for Mumbai Airport test case problem by Baijal etal. (X axis represents nodes and Y axis represents time slots)

Baijal test case solution cost	Global optimum	No. of MILPs solved	Optimum reached	Runtime	Baijal solution excess cost
186	166	124416	2 mins (2%)	151 mins	12%

As can be seen from the results, the solutions generated by the global search algorithm are quite naturally better than those generated by the benchmarked algorithms. But its exhaustive nature leads to long runtimes rendering it unfeasible for real-time usage. It can however be seen that the global optimum is reached at a significantly early stage of the run; rest of the run only ensures its global optimality.

III. Work in Progress

The global search algorithm treats every flight-path combination equally and runs an MILP on each. But merely arranging the paths in order of increasing lengths results in *global optima* being reached in 2 and 4 minutes for the Baijal [1] and Roling [3] problems respectively. Thus the algorithm if appropriately modified definitely appears to have a potential to give at least near optimal solutions in real-time.

In order to modify the global search algorithm to work in real-time, a method to appraise a flight-path combination without running the MILP is being explored. The use of A* algorithm on lines of Gottland et al. [2] to estimate the lower bound on the schedule for a particular flight-path combination is being examined. Another approach to reduce the number of MILP runs is using a stochastic method such as Genetic algorithm or BAFO [2, 1] for combination selection.

To deal with large problem sizes in terms of the number of aircraft within the scheduling horizon, the concept of *Rolling Horizons* [10, 9] is being explored. Rolling horizons essentially extrapolates the first come first serve method of priority allocation [5] to a group of aircraft. The aircraft are divided into batches based on time and aircraft within a batch are allowed to interact and affect each other's schedules. The aircraft in the next batch are then scheduled around these aircraft with some buffer aircraft between the two sets. These buffer aircraft's schedules can be affected by either of the batch and by effect, can affect either of the batches.

To incorporate variability of aircraft arrival times, arrival nodes and operational conditions, means to introduce empirical information into the algorithm would be investigated. Thus an algorithm that *learns* from the past data and quality of decisions made on lines of the work by Khadilkar [6], could be explored.

References

[1] Baijal A. and Pant R. S. Airport ground movement optimization using bacterial foraging algorithm. *Proceedings of International Symposium on Combinatorial Optimization, University of Oxford*.

[2] Gotteland J. B., Durand N., Alliot J. M., and Page E. Aircraft ground traffic optimization. *Proceedings of the 4th International Air Traffic Management R & D Seminar, Santa Fe, NM, USA*.

[3] Roling P. C. and Visser H. G. Optimal airport surface traffic planning using mixed-integer linear programming. *International Journal of Aerospace Engineering*, Volume 2008(Article ID 732828).

[4] Advanced Surface Movement Guidance and Control Systems Manual. 2004.

[5] Baik H, Sherali H. D., and Trani A. A. Time dependent network assignment strategy for taxiway routing at airports. *Transport Research Record*.

[6] Khadilkar H. *Networked Control of Aircraft Operations at Airports and in Terminal Areas*. PhD thesis.

[7] CPLEX release 12.5 ILOG. 2012.

[8] Airport CDM Implementation Manual. 2012.

[9] Gupta P., Subramanian H., and Pant R. S. Generation of optimized routes and schedules for surface movement of aircraft on taxiways. *AIAA ATIO/ISSMO Conference, Fort Worth, TX, USA*.

[10] Smeltink J. W., Soomer M. J., Waal P. R., and Van der Mei R. D. An optimization model for airport taxi scheduling. *Proceedings of the INFORMS Annual Meeting, Denver, Colo, USA*.