METAHEURISTICS GRASP AND ILS APPLIED TO THE AIRCRAFT ROTATION PROBLEM

Alexander de Almeida Pinto

Laboratory for Applications of Digital Video
Federal University of Paraiba
Joao Pessoa, Brazil

Daniel Goncalvez Ramos

Laboratory for Applications of Digital Video
Federal University of Paraiba
Joao Pessoa, Brazil

Lucidio dos Anjos Formiga Cabral

Departament of Informatic
Federal University of Paraiba
Joao Pessoa, Brazil

Abstract

The Aircraft Rotation Problem refers to the designation of each airplane available at airlines companies such that all planned flights are covered. This is a light constrained problem since any aircraft can be assigned to an unlimited quantity of duty work. However, there are many lower cost solutions that can be obtained by simply adjusting the time of departure of a flight, or even positioning some aircraft by executing ferry flights between two near airports. We describe an algorithm based on GRASP metaheuristic using ILS as local search, which is able to calculate good solution to large scaled problems. Some computational results are presented for a real world problem at Rio-Sul Brazilian Airlines.

Keywords: Aircraft Rotation, Optimization, Metaheuristic, GRASP, ILS

1 Introduction

The airline industry has been a rich source of problems related to operational research, mainly due to combinatorial explosive nature of these problems. Typically, planning problems involve the creation of lines of work for both aircraft and crew.

On the literature, there are two well-known problems associated to aircraft planning, the Fleet Assignment and the Aircraft Rotation Problem (ARP). The first deals with the allocation of the fleet, determining the type of aircraft used in each flight [1]. The second is the one covered on this paper, and will be explained on the next section.

Due to the combinatorial explosion of the problems involving airlines, they should be solved separately, though each problem should be solved giving conditions to the next one be solved.

Many techniques have been used to give a solution to the ARP, some of them using exact methods, which always obtain the best result, but uses lots of CPU, and in practice is unfeasible, due to the size of real problems. Other techniques try to find a good solution, instead of the best, and try to do that in a fast way. To do so, they use metaheuristics, exploting some historical knowledge of the solutions combined with some characteristic of the problem to find good solutions for big instances.

1.1 The ARP problem

After solving the Fleet Assignment problem, the one that chooses the kind of airplane that will cover the flights, the problem becomes the organization of the flights. It means, how flights will be connected. The ways to group them can have a very big impact on the cost of the flight network. This cost comes from several factors, like the numbers of aircrafts needed to cover all the flights, the delay between them, the downtime of the flights, and others.

To solve this problem, we must be aware of some constraints involving time and space. For example, an airplane cannot leave before the arrival of the flight that precedes it, neither from a different city that it has arrived on the last flight. Though these are obvious constraints, they make the problem really hard to solve in a large scale, as we could have a very big number of combinations between the connections of the flights. There is also another constraint, related to the time that the airplane has to be on the ground before it leaves. This restriction is usually related to an airport.

Considering these restrictions, we should arrange the flights looking for a cheap network. Its known that finding the cheapest is almost impossible for big instances, but finding a good one can be achieved with some metaheuristics.

Another important aspect is the maintaining restrictions. It is known that a plane must have periodic checks. Opportunities to perform these tasks occur only on some connections that are potentially available. As consequence, a sequence of flights must be constructed so that these restrictions are not violated. In order to easily incorporate these constraints in our framework, we assume that the rotations are not assigned to specific types of aircraft. Thus, if an aircraft needs maintenance, is created a special flight depart and arrive from the maintenance base chosen. The time of flight is exactly the service time [1].

2 Methodology

In order to find good solutions to the ARP problem, two metaheuristics were implemented using java Language. On this section, it will be exposed how to represent a solution, as well as some details about the application of the metaheuristics on the ARP problem.

2.1 Problem Concepts

Each solution has the flight network, with all the connections between the flights. To represent the flights covered by a unique airplane, the concept of a track was applied. So, the goal is to find the smallest number of tracks needed to cover all flights.

Each track is composed by any number of flights, considering the restrictions described on last section. The connection between flights is called arc. An arc has to be one of the followings:

- Simple Connection Arc (type 1): This arc connects flights that dont need to be changed. There is no need to create another flight to connect them too. The cost of this arc is zero.
- Connection With Delay Arc (type 2): With this arc, the second flight needs to be delayed, in order to conform the time restriction. The cost of this arc

is the delay (in minutes) of the second flight.

- Connection With Repositioning Arc (type 3): This arc creates a third flight between the two previously selected, in order to conform the space condition. The third flight goes from the destination of the first flight to the origin of the second flight. The cost of this arc is the duration of the created flight.
- Connection With Repositioning and Delay Arc (type 4): Besides creating a new flight to connect the two ones, this type of arc also delays the second flight, so it obeys both space and time restrictions. The cost of this arc is the addition of the delayed time plus the duration of the created flight.

2.2 Problem Solution and Representation

The generation of a solution was divided into two steps using GRASP and ILS. The first one constructs a good solution and the second tries to improve the given solution with some local search algorithms avoiding local minimums.

3 The constructive algorithm

In this phase, the computer generates a good solution, following the steps:

3.1 Track formation

3.1.1 Initializing a track

To initialize a track, a flight is randomically chosen between the n flights that have the lower departure time and that have not been chosen by another track.

3.1.2 Composing and finalizing a track

After choosing the first flight, or with the previous flight, it is made the choice of the next arc k, $k \in \{K1, K2, K3, K4\}$, where Ki is the set of i-type arcs. This choice is made with a probability associated with each arch type P1, P2, P3, P4. The next flight should obey the arch rule and should not have a departure time very far from the arrival of the last. With all the candidates, only the α fraction the best are chosen to be a part of the possible set. So it is randomically picked one and added to the track.

The value of α was defined as 0.15, to type 1 arcs, and variable, according to the number of candidates, to the other arches.

If the generated set dont have any flight, then the arch type will be changed. If no arch have any possible flight, then the last flight ends the track.

3.2 Composing a network

It is generated as much tracks as all the flights are covered. The constructive algorithm stops when all the flights are already corered.

4 The local search Algorithm

After selecting the best solution generated in construction phase, it is used some algorithms to try to find a way to improve the solution. It was used the swap and compaction methods and the ILS (Iterated Local Search) metaheuristic.

Three kinds of swaps were implemented, the Swap-1, Swap-2 and the Swap 3. The swap-n procedure is done by trying to change n connected flights with 1 to $(n + 1)$ flights. The change is accepted if there is a cost reduction. The compaction method tries to merge two different tracks into one.

The ILS algorithm was implemented in a way of perturbing the current solution, and then applying the methods described before. Forcing four Swap-2 movements, not considering if they improve the solution, makes this perturbation.

4.1 Computational results

To run the tests, it was used a small daily flight network from Rio-Sul airlines [1]. All data is described on Attachment A to D. With this information, any program can be tested with the same entrance, verifying its efficiency.

All tests run on a Intel Core 2 T5300 1.73GHz, with 1GB DDR-2 RAM. The algorithm was implemented using Java, on NetBeans IDE 6.0.1.

The Table 1 shows details of the achievement of the best solution:

Number of tracks	Cost	Time(seconds)
21	21335	0.039
17	17100	0.113
17	17078	0.315
17	17074	0.346
17	17066	6.012
17	17054	9.612
17	17048	14.179
17	17040	15.0
17	17033	19.484

Table 1 - Cost and time of solutions

On the attachment A we have a visualization of the best result obtained and in attachment D this result can be viewed in more detail.

In the test described was set a ground time standard for all airports, which is 20 minutes this single parameter was fixed at work. The time between two airports can be found in Attachment C.

An account of less than 15 aircraft was determined using a degree of freedom of 17 minutes, as can be seen in the attachment E.

5 Conclusion and Future Researches

The aircrafts are one of the most expensive resources on aeronautics, so every flight planning should be done for a cost reduction. The ARP is one of the problems that can lead to a very high operational cost to a company, if erroneously treated or ignored.

With simple time changing or creating a repositioning flight, the flight network could have a substantial improvement, decreasing aircrafts downtime and reducing the number of planes needed to cover all the defined flights.

For next researches, some efforts are being made to idealize a complete mathematical model of the ARP for use in a hybrid method, leading with weekly tasks.

References

[1] Pimentel, A. G., *Uma abordagem heurística para a solução de problemas de recobrimento de conjuntos de grande porte, com aplicação à alocação de tripulações para companhias aéreas*, Tese de Doutorado, Universidade Federal do Rio de Janeiro (2002).

[2] Pontes, R., Lucena, A. e Cabral, L., *Exact and Heuristic Techniques for Solving the Aircraft Rotation Problem*, Technical report Department of Statistics UFPB (2002).

[3] Rezende, M, *Greedy Randomized Adaptative Search Procedures (GRASP)* , Technical Report, ATT Labs Research (1998).

[4] Rubin, J. , "A Technique for the Solution of Massive Set Covering Problems with Application to Airline Crew Scheduling," Transportation Science (1973).

[5] Clarke L, Johnson E, Nemhauser G, Zhu ZX, "The Aircraft Rotation Problem," Annals of Operations Research (1997) 69: 33-46

[6] Klabjan, D. Johnson, E. Nemhauser, G., "Solving Large Airline Crew Scheduling Problems: Random Pairing Generation and Strong Branching," Technical Report, Georgia Institute of Technology (1999).

A Graphical Representation of the best solution

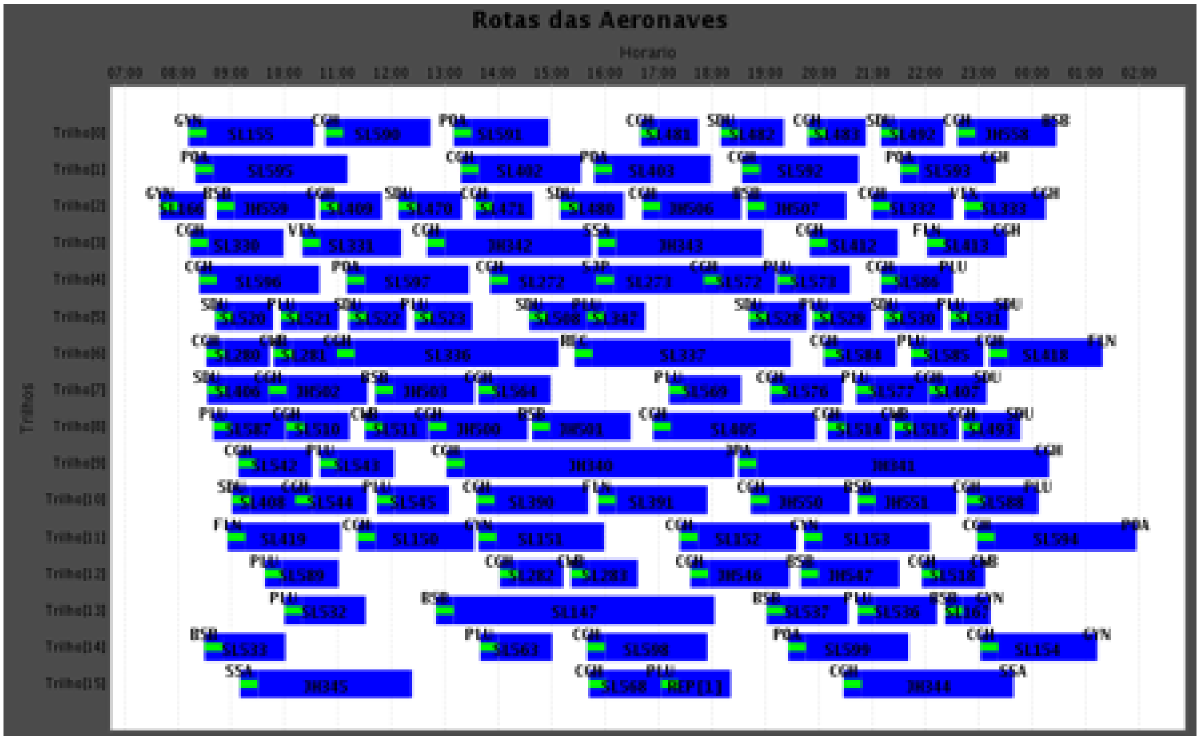

Figure 1 - Graphical Representation of the best solution

B Flights Network

001	SL166	08:00	08:31	GYN	BSB
002	SL155	08:32	10:33	GYN	CGH
003	SL330	08:34	09:59	CGH	VIX
004	SL595	08:40	11:11	POA	CGH
005	SL596	08:44	10:39	CGH	POA
006	SL533	08:50	10:01	BSB	PLU
007	SL280	08:52	09:43	CGH	CWB
008	SL587	09:00	10:01	PLU	CGH
009	SL520	09:02	09:47	SDU	PLU
010	JH559	09:04	10:35	BSB	CGH
011	SL406	09:06	09:55	SDU	CGH
012	SL419	09:16	11:03	FLN	CGH
013	SL542	09:28	10:31	CGH	PLU
014	SL408	09:30	10:21	SDU	CGH
015	JH345	09:30	12:23	SSA	CGH
016	SL589	09:58	11:01	PLU	CGH
017	JH502	10:02	11:33	CGH	BSB
018	SL281	10:08	10:59	CWB	CGH
019	SL521	10:16	11:01	PLU	SDU
020	SL532	10:20	11:31	PLU	BSB
021	SL510	10:22	11:13	CGH	CWB
022	SL544	10:32	11:33	CGH	PLU
023	SL331	10:40	12:11	VIX	CGH
024	SL409	11:00	11:49	CGH	SDU
025	SL543	11:00	12:03	PLU	CGH
026	SL590	11:06	12:44	CGH	POA
027	SL336	11:16	15:05	CGH	REC
028	SL597	11:30	13:27	POA	CGH
029	SL522	11:32	12:17	SDU	PLU
030	SL150	11:42	13:33	CGH	GYN
031	SL511	11:50	12:41	CWB	CGH
032	JH503	12:02	13:33	BSB	CGH
033	SL545	12:04	13:05	PLU	CGH
034	SL470	12:28	13:19	SDU	CGH
035	SL523	12:46	13:31	PLU	SDU
036	JH342	13:00	15:45	CGH	SSA
037	JH500	13:02	14:33	CGH	BSB
038	SL147	13:10	18:03	BSB	BSB
039	JH340	13:22	18:25	CGH	JPA
040	SL591	13:30	14:57	POA	CGH
041	SL402	13:38	15:33	CGH	POA
042	SL471	13:54	14:39	CGH	SDU
043	SL390	13:56	15:41	CGH	FLN
044	SL564	13:58	14:59	CGH	PLU
045	SL151	13:58	15:59	GYN	CGH
046	SL563	14:00	15:01	PLU	CGH
047	SL272	14:12	15:51	CGH	SJP
048	SL282	14:22	15:13	CGH	CWB
049	SL508	14:56	15:41	SDU	PLU
050	JH501	14:58	16:29	BSB	CGH
051	SL480	15:30	16:21	SDU	CGH
052	SL283	15:42	16:37	CWB	CGH
053	SL337	15:46	19:29	REC	CGH
054	SL347	16:00	16:45	PLU	SDU
055	SL598	16:00	17:55	CGH	POA
056	SL568	16:02	17:03	CGH	PLU
057	SL403	16:08	17:59	POA	CGH
058	SL273	16:10	17:51	SJP	CGH
059	SL391	16:12	17:55	FLN	CGH
060	JH343	16:12	18:57	SSA	CGH
061	SL481	17:00	17:45	CGH	SDU
062	JH506	17:02	18:33	CGH	BSB
063	SL405	17:14	19:57	CGH	CGH
064	SL569	17:32	18:33	PLU	CGH
065	SL152	17:44	19:35	CGH	GYN
066	JH546	17:56	19:27	CGH	BSB
067	SL572	17:58	18:59	CGH	PLU
068	SL482	18:30	19:21	SDU	CGH
069	JH341	18:50	00:19	JPA	CGH
070	SL592	18:54	20:45	CGH	POA
071	JH507	19:00	20:31	BSB	CGH
072	SL528	19:02	19:47	SDU	PLU
073	JH550	19:04	20:35	CGH	BSB
074	SL537	19:22	20:33	BSB	PLU
075	SL576	19:26	20:27	CGH	PLU
076	SL573	19:34	20:35	PLU	CGH
077	SL599	19:46	21:41	POA	CGH
078	JH547	20:00	21:31	BSB	CGH
079	SL153	20:04	22:05	GYN	CGH
080	SL483	20:08	20:53	CGH	SDU
081	SL412	20:10	21:29	CGH	FLN
082	SL529	20:14	20:59	PLU	SDU
083	SL584	20:26	21:27	CGH	PLU
084	SL514	20:30	21:21	CGH	CWB
085	JH344	20:48	23:39	CGH	SSA
086	SL577	21:02	22:03	PLU	CGH
087	SL536	21:04	22:13	PLU	BSB
088	JH551	21:04	22:35	BSB	CGH
089	SL332	21:20	22:31	CGH	VIX
090	SL492	21:30	22:21	SDU	CGH
091	SL586	21:30	22:31	CGH	PLU
092	SL530	21:34	22:19	SDU	PLU
093	SL515	21:45	22:37	CWB	CGH
094	SL593	21:52	23:19	POA	CGH
095	SL585	22:04	23:05	PLU	CGH
096	SL518	22:16	23:07	CGH	CWB
097	SL413	22:22	23:31	FLN	CGH
098	SL407	22:24	23:09	CGH	SDU
099	SL167	22:42	23:13	BSB	GYN
100	SL531	22:48	23:33	PLU	SDU
101	JH558	22:56	00:27	CGH	BSB
102	SL493	23:02	23:47	CGH	SDU
103	SL333	23:04	00:15	VIX	CGH
104	SL588	23:06	00:07	CGH	PLU
105	SL594	23:18	01:57	CGH	POA
106	SL154	23:22	01:13	CGH	GYN
107	SL418	23:32	01:19	CGH	FLN

C Flight Time Between Airports

BSB	CGH	091	BSB	CWB	125	BSB	FLN	152	BSB	GYN	031
BSB	JPA	197	BSB	PLU	071	BSB	POA	185	BSB	REC	191
BSB	SDU	107	BSB	SJP	066	BSB	SSA	125	BSB	VIX	109
CGH	CWB	051	CGH	FLN	105	CGH	GYN	095	CGH	JPA	255
CGH	PLU	058	CGH	POA	097	CGH	REC	246	CGH	SDU	049
CGH	SJP	049	CGH	SSA	171	CGH	VIX	087	CWB	FLN	028
CWB	GYN	114	CWB	JPA	293	CWB	PLU	095	CWB	POA	062
CWB	REC	283	CWB	SDU	078	CWB	SJP	060	CWB	SSA	208
CWB	VIX	125	FLN	GYN	142	FLN	JPA	311	FLN	PLU	114
FLN	POA	042	FLN	REC	300	FLN	SDU	087	FLN	SJP	088
FLN	SSA	225	FLN	VIX	136	GYN	JPA	216	GYN	PLU	076
GYN	POA	173	GYN	REC	210	GYN	SDU	109	GYN	SJP	054
GYN	SSA	143	GYN	VIX	118	JPA	PLU	198	JPA	POA	352
JPA	REC	013	JPA	SDU	226	JPA	SJP	251	JPA	SSA	085
JPA	VIX	181	PLU	POA	155	PLU	REC	187	PLU	SDU	045
PLU	SJP	067	PLU	SSA	113	PLU	VIX	044	POA	REC	342
POA	SDU	129	POA	SJP	119	POA	SSA	267	POA	VIX	177
REC	SDU	215	REC	SJP	242	REC	SSA	075	REC	VIX	169
SDU	SJP	079	SDU	SSA	130	SDU	VIX	048	SJP	SSA	169
SJP	VIX	110	SSA	VIX	097						

D Result Obtained By The GRASP

```
Trilho[1]
SL155 08:32 10:33 GYN CGH
SL590 11:06 12:44 CGH POA
SL591 13:30 14:57 POA CGH
SL481 17:00 17:45 CGH SDU
SL482 18:30 19:21 SDU CGH
SL483 20:08 20:53 CGH SDU
SL492 21:30 22:21 SDU CGH
JH558 22:56 00:27 CGH BSB
SL332 21:20 22:31 CGH VIX
SL333 23:04 00:15 VIX CGH

Trilho[2]
SL595 08:40 11:11 POA CGH
SL402 13:38 15:33 CGH POA
SL403 16:08 17:59 POA CGH
SL592 18:54 20:45 CGH POA
SL593 21:52 23:19 POA CGH
SL480 15:30 16:21 SDU CGH
JH506 17:02 18:33 CGH BSB
JH507 19:00 20:31 BSB CGH

Trilho[3]
SL166 08:00 08:31 GYN BSB
JH559 09:04 10:35 BSB CGH
SL409 11:00 11:49 CGH SDU
SL470 12:28 13:19 SDU CGH
SL471 13:54 14:39 CGH SDU

Trilho[4]
SL330 08:34 09:59 CGH VIX
SL331 10:40 12:11 VIX CGH
JH342 13:00 15:45 CGH SSA
JH343 16:12 18:57 SSA CGH
SL412 20:10 21:29 CGH FLN
SL413 22:22 23:31 FLN CGH
SL586 21:30 22:31 CGH PLU
SL529 20:14 20:59 PLU SDU
SL530 21:34 22:19 SDU PLU
SL531 22:48 23:33 PLU SDU

Trilho[5]
SL596 08:44 10:39 CGH POA
SL597 11:30 13:27 POA CGH
SL272 14:12 15:51 CGH SJP (-1)
SL273 16:10 17:51 SJP CGH
SL572 17:58 18:59 CGH PLU (+13)
SL573 19:34 20:35 PLU CGH
SL528 19:02 19:47 SDU PLU

Trilho[6]
SL520 09:02 09:47 SDU PLU
SL521 10:16 11:01 PLU SDU
SL522 11:32 12:17 SDU PLU
SL523 12:46 13:31 PLU SDU
SL508 14:56 15:41 SDU PLU (-1)
SL347 16:00 16:45 PLU SDU

Trilho[7]
SL280 08:52 09:43 CGH CWB
SL281 10:08 10:59 CWB CGH
SL336 11:16 15:05 CGH REC (+3)
SL337 15:46 19:29 REC CGH
SL584 20:26 21:27 CGH PLU
SL585 22:04 23:05 PLU CGH
SL418 23:32 01:19 CGH FLN
SL407 22:24 23:09 CGH SDU
SL493 23:02 23:47 CGH SDU

Trilho[8]
SL406 09:06 09:55 SDU CGH (-13)
JH502 10:02 11:33 CGH BSB
JH503 12:02 13:33 BSB CGH
SL564 13:58 14:59 CGH PLU
SL569 17:32 18:33 PLU CGH
SL576 19:26 20:27 CGH PLU
SL577 21:02 22:03 PLU CGH
SL515 21:45 22:37 CWB CGH

Trilho[9]
SL587 09:00 10:01 PLU CGH
SL510 10:22 11:13 CGH CWB
SL511 11:50 12:41 CWB CGH
JH500 13:02 14:33 CGH BSB
JH501 14:58 16:29 BSB CGH
SL405 17:14 19:57 CGH CGH
SL514 20:30 21:21 CGH CWB

Trilho[10]
SL542 09:28 10:31 CGH PLU
SL543 11:00 12:03 PLU CGH
JH340 13:22 18:25 CGH JPA
JH341 18:50 00:19 JPA CGH
SL391 16:12 17:55 FLN CGH
JH550 19:04 20:35 CGH BSB
JH551 21:04 22:35 BSB CGH
SL588 23:06 00:07 CGH PLU

Trilho[11]
SL408 09:30 10:21 SDU CGH (-9)
SL544 10:32 11:33 CGH PLU
SL545 12:04 13:05 PLU CGH
SL390 13:56 15:41 CGH FLN
SL153 20:04 22:05 GYN CGH
SL594 23:18 01:57 CGH POA

Trilho[12]
SL419 09:16 11:03 FLN CGH
SL150 11:42 13:33 CGH GYN
SL151 13:58 15:59 GYN CGH
SL152 17:44 19:35 CGH GYN

Trilho[13]
SL589 09:58 11:01 PLU CGH
SL282 14:22 15:13 CGH CWB
SL283 15:42 16:37 CWB CGH
JH546 17:56 19:27 CGH BSB
JH547 20:00 21:31 BSB CGH
SL518 22:16 23:07 CGH CWB

Trilho[14]
SL532 10:20 11:31 PLU BSB
SL147 13:10 18:03 BSB BSB
SL537 19:22 20:33 BSB PLU
SL536 21:04 22:13 PLU BSB
SL167 22:42 23:13 BSB GYN

Trilho[15]
SL533 08:50 10:01 BSB PLU
SL563 14:00 15:01 PLU CGH
SL598 16:00 17:55 CGH POA
SL599 19:46 21:41 POA CGH
SL154 23:22 01:13 CGH GYN

Trilho[16]
JH345 09:30 12:23 SSA CGH
SL568 16:02 17:03 CGH PLU
NOVO 17:23 18:21 PLU CGH
JH344 20:48 23:39 CGH SSA
```

E Colision with window

All ground time is fixed in 20 minutes.

16 Flights with up to 1 minute window	15 Flights with up to 17 minute window
001 SL155 0 08:32 0 10:33 GYN CGH	068 JH341 0 18:50 1 00:19 JPA CGH
002 SL330 0 08:34 0 09:59 CGH VIX	069 SL592 0 18:54 0 20:45 CGH POA
003 SL595 0 08:40 0 11:11 POA CGH	070 JH507 0 19:00 0 20:31 BSB CGH
004 SL596 0 08:44 0 10:39 CGH POA	072 JH550 0 19:04 0 20:35 CGH BSB
005 SL533 0 08:50 0 10:01 BSB PLU	073 SL537 0 19:22 0 20:33 BSB PLU
006 SL280 0 08:52 0 09:43 CGH CWB	074 SL576 0 19:26 0 20:27 CGH PLU
007 SL587 0 09:00 0 10:01 PLU CGH	075 SL573 0 19:34 0 20:35 PLU CGH
008 SL520 0 09:02 0 09:47 SDU PLU	076 SL599 0 19:46 0 21:41 POA CGH
009 JH559 0 09:04 0 10:35 BSB CGH	077 JH547 0 20:00 0 21:31 BSB CGH
010 SL406 0 09:06 0 09:55 SDU CGH	078 SL153 0 20:04 0 22:05 GYN CGH
011 SL419 0 09:16 0 11:03 FLN CGH	079 SL483 0 20:08 0 20:53 CGH SDU
012 SL542 0 09:28 0 10:31 CGH PLU	080 SL412 0 20:10 0 21:29 CGH FLN
013 SL408 0 09:30 0 10:21 SDU CGH	081 SL529 0 20:14 0 20:59 PLU SDU
014 JH345 0 09:30 0 12:23 SSA CGH	082 SL584 0 20:26 0 21:27 CGH PLU
015 SL589 0 09:58 0 11:01 PLU CGH	083 SL514 0 20:30 0 21:21 CGH CWB
016 JH502 0 10:02 0 11:33 CGH BSB	