



E.02.18 – ELSA_D2.3

Calibrated agent-based model – final draft

Document information

Project title	Empirically grounded agent based models for the future ATM scenario
Project N°	E.02.18.
Project Manager	Deep Blue
Deliverable Name	Calibrated agent-based model – final draft
Deliverable ID	D2.3
Edition	00.00.04
Template version	02.00.00

Task contributors

G. Affronti, C. Bongiorno, R.N. Mantegna, S. Miccichè [UNIPA], G. Gurtner, F. Lillo, L. Valori [SNS], M. Ducci, B. Monechi, S. Pozzi [Deep Blue]

Abstract

This document is the deliverable D2.3 “Calibrated agent-based model – final draft” describing the Agent Based Model developed within the project. The model is composed of two distinct layers: the strategic layer, simulating the interaction between the Network Manager and the Airline Operators and the tactical layer, focused on aircraft and controllers’ behavior in a single Air Traffic Control (ATC) sector.

The strategic layer simulates the submission of flight plans by the air companies to the network manager, who is in charge of computing the loads of sectors and reject flight plans if capacity of some sector is exceeded. Some sectors can also be randomly shut down after the first round of submissions, which triggers an additional round. The airspace is modeled as the superposition of a network of navpoints and a network of sectors. Finally, an automatic and quantitative procedure of calibration has been implemented in order to assess the values of the implicit parameters of the model and to generate scenario simulation in the presence of sector closure.

The tactical layer works at the level of a single ATC sector. We present here simulation experiments of the current ATM scenario in order to investigate how the model performs in presence of areas of the sector closed to air traffic. We consider different possibilities regarding the concentration and size of these areas showing that for relatively small shocked areas the model generates trajectories that are more optimized than planned ones in terms of the total time needed to cross the sector.

Authoring & Approval

Prepared By		
Name & company	Position / Title	Date
Salvatore Miccichè (UNIPA)	Assistant Professor of Applied Physics	16/02/2014
Giovanni Affronti (UNIPA)	Fellow holder	16/02/2014
Christian Bongiorno (UNIPA)	Fellow holder	16/02/2014
Fabrizio Lillo (SNS)	Junior Research Professor in Mathematical Finance	16/02/2014
Gérald Gurtner (SNS)	Postdoc	16/02/2014
Luca Valori (SNS)	Postdoc	16/02/2014
Marco Ducci (DBL)	Validation Expert	16/02/2014
Bernardo Monechi (DBL and UNIROMA1)	PhD candidate	16/02/2014

Reviewed By		
Name & company	Position / Title	Date
Rosario Nunzio Mantegna (UNIPA)	Full Professor of Applied Physics	24/02/2014

Approved By		
Name & company	Position / Title	Date
Simone Pozzi (DBL)	Project leader	24/02/2014

Document History

Edition	Date	Status	Author	Justification
00.00.01	12/02/2014		F. Lillo	First draft version
00.00.02	16/02/2014		F. Lillo	Second draft version
00.00.03	22/02/2014		F. Lillo	Proposed final version
00.00.04	24/02/2014		F. Lillo	Revised final version
00.00.05	14/07/2014		F. Lillo	Final version

IPR (foreground)

Foreground owned by the ELSA consortium.

foundling members



Avenue de Cortenbergh 100 | B- 1000 Bruxelles | www.sesarju.eu

Contents

INTRODUCTION	13
EXECUTIVE SUMMARY	14
1 Strategic layer	15
1.1 Introduction	15
1.2 Model 2.0 - General features	17
1.2.1 Airspace structure	17
1.2.2 Agents - Air Operators	21
1.2.3 Agents - Network Manager	22
1.2.4 Validation of the Strategic Layer	23
1.3 Model 2.0 - Implementation of the model	24
1.3.1 Networks	24
1.3.2 Crossing times	26
1.3.3 Capacities and Airports	27
1.3.4 Air Operators' behavior	27
1.3.5 Network Manager's behavior	30
1.3.6 Metrics	30
1.4 Model 1.0 - Toy model	32
1.4.1 Airspace structure	32
1.4.2 Airline Operators	32
1.4.3 Network Manager	32
1.5 List of parameters	33
1.5.1 Parameters linked to the network	33
1.5.2 Parameters of the simulation	33
1.6 Results of Model 1.0	34
1.6.1 Basic Mechanisms	34
1.6.2 Mixed populations	41
1.6.3 Departure time pattern	44
1.6.4 Shocks: from M0 to M1	49
1.6.5 Generalization of the results	50
1.6.6 Calibration on real data	52
1.6.7 Conclusions	59
1.7 Results with model 2.0	61
1.7.1 Calibration of the model	61
1.7.2 Testing the model	65
1.7.3 Pure population	67
1.7.4 Mixed population	69
1.7.5 Simulations of scenarios where sectors are closed.	72
1.7.6 Conclusions	74

2 Tactical Layer	75
2.1 Introduction	75
2.2 General features	75
2.2.1 Current scenario Model A	75
2.3 Implementation of the Model	77
2.3.1 Navigation Points module	77
2.3.2 Flight List module	78
2.3.3 Conflict detection module	78
2.3.4 Conflict resolution module	79
2.3.5 Directs module	82
2.3.6 Expected results	82
2.3.7 Input from Validation experts	82
2.3.8 Calibration of the model	84
2.4 Results	93
2.4.1 Poissonian temporal occurrence of shocked area	93
2.4.2 Temporal occurrence localized in specific time intervals	99
2.4.3 Conclusions	100
BIBLIOGRAPHY	103
A Topology properties of the networks	105
B Analytical study of the rejection	107
B.1 Probabilities without overlap	107
B.1.1 First rejection	107
B.1.2 Second rejection and higher	108
B.2 Probability with overlap	109
B.3 Numerical results	109
C Probability to shift/reroute	113
D Details of the Genetic algorithm	115
E Other simulations	118

List of Figures

1.1	Traffic network of navpoints (top panel) and sectors (bottom panel) for France : each navpoint/sector is represented by a node. Two nodes are linked if at least one flight goes from one to the other and the link is weighted proportionally to the number of flights. The colors on the bottom panel represent pieces of ACCs the sector belongs to.	18
1.2	The process of flight planning and monitoring in the Alitalia OCC.	23
1.3	Left: example of a Delaunay triangulation of 30 random points in a square region of the plane. Center: example of a triangular lattice of 30 points in a square region of the plane. Right: example of an Erdos Rényi random graph of order (<i>i.e.</i> number of nodes) 30.	25
1.4	Voronoi tessellation of 8 randomly distributed points on plane. First, we draw randomly some points on the plane (left). Then we build the Voronoi tessellation (center) by drawing the perpendicular bisectors between each pairs of points, stopping the line when it crosses another one. All points in one given tile are now closer to the corresponding vertex than to any other vertex (definition of the Voronoi cell). Finally, the Delaunay triangulation can be easily drawn (right) by putting a link between every pairs of vertices which have one common boundary. Voronoi and Delaunay networks are dual and uniquely defined. Moreover, the Delaunay triangulation is the best triangulation one can get for a set of points.	25
1.5	The figure shows the N_{fp} flight plans selection operated by an AO: given the network and the flight (as a pair of departing/arriving airports), the N_{fp} shortest paths are determined. Different combinations of paths and departing times are tried and their costs are computed. The combinations are sorted following decreasing cost; the N_{fp} with higher rank are selected to represent the array of proposal for the given AO and for the given flight.	28
1.6	Graph used for running the model in the unweighted case. The graph is a Delaunay triangulation of 60 randomly chosen points on a plane. Green nodes represent sectors with airports. In the simulations, node 49 is taken as departing node, while node 46 (at a distance 4) or node 23 (at a distance 6) are the arriving ones.	35
1.7	Left: rejection statistics for a simulation with 100 AOs flying on a 60 node graph. The chosen airport sectors are node 49 and node 46 of Fig. 1.6. Right: rejection statistics for a simulation with 100 AOs flying on a 60 node graph. The chosen airport sectors are node 49 and node 23 of Fig. 1.6.	35
1.8	Graph used for running the model in the weighted case. The graph is a Delaunay triangulation of 90 randomly chosen points on a plane. Green nodes represent sectors with airports. The thicker is the edge, the higher is the associated weight.	37

1.9	Average time of the first rejection in function of the order of rejection for different configurations of the cost function parameters α and β . The graph used for the simulation is a Delaunay triangulation of 90 points and the topological distance between the starting and the arriving airports is 6 edges.	37
1.10	Average number of unregulated flights versus number of flights. Different colors correspond to different configurations of the cost function parameters. We set $N_{fp} = 10$	38
1.11	Satisfaction versus number of flights for different values of β/α	40
1.12	Average number of regulated FPs versus satisfaction of the system. Different colors correspond to different configurations of the cost function parameters. As in all the simulations, the number of flight plans is $N_{fp} = 10$	40
1.13	Global satisfaction versus total number for different values of the fraction f_S of flights S.	42
1.14	Satisfaction of S versus total number of flights for different values of the fraction f_S of flights S.	42
1.15	Satisfaction of R versus total number of flights for different values of the fraction f_S of flights S.	43
1.16	Satisfaction of S and R versus the percentage of flights of R, where 100 flights are simulated.	43
1.17	Global satisfaction versus percentage of flights R, where 100 flights are simulated.	44
1.18	Normalized global satisfaction versus the percentage of flights R, where 100 flights are simulated.	45
1.19	Distribution of departing times. In green, the distribution of desired times. In blue, superimposed, the distribution of real departure. Left: run with companies only of type R. Right: run with companies only of type S.	46
1.20	Satisfaction as a function of the ratio α/β of the cost function for different values of Δt . The simulation is done with a number of flights per peak kept fixed, which the overall satisfaction is decreasing when Δt decreases: there are more and more flights in total, because there are more and more waves.	47
1.21	Global satisfaction versus total number of flights when the two types of companies S and R and mixed for different fraction of companies S. Left: case where $\Delta t = 23$. Right: case where $\Delta t = 1$	47
1.22	Global satisfaction against the fraction of company S for different values of Δt . The number of flights is set to 120.	48
1.23	Satisfaction of company S (left) and R (right) against the fraction f_S of company S for different values of Δt	48
1.24	Satisfaction of the system against the number of shocks for different the ratio β/α in the pure population case (left panel) or different values of the fraction of company S in the mixed populations case (right panel).	49
1.25	Histogram of the quantity $\delta s(0)$ (see Eq. 1.10) giving the difference in satisfaction between company S and R when the departing time pattern is uniform ($\Delta t = 0$).	51
1.26	Relationship between overlap and $\delta s(0)$ (see Eq. 1.10), giving the difference in satisfaction between company S and R when the departing time pattern is uniform (i.e. $\Delta t = 0$). We show the average on each quantile of overlap. Error bars are standard deviations.	51
1.27	Global satisfaction (up), and satisfaction of each company (bottom) as a function of f_S for different values of the number of companies per waves.	53
1.28	Left. Histogram of time of crossing between sectors (average time to reach the center of a sector from another one). Mean/std: 15.0/9.15. Results obtained from the whole ECAC space for one day (6th of May 2010). Right. QQ plot of the sample data versus the best fit with a lognormal distribution.	53

1.29	Histogram of a proxy for the capacity of sectors i.e. the maximum number of flights in the sector at a given time. Left: entire histogram. Right: zoom on 20 first values.	54
1.30	Left panel: relationship between capacity and volume of the sectors. Center panel: relationship between capacity and area. Right panel: relationship between capacity and area averaged on quantiles.	55
1.31	Network of sectors of France used to run the ABM. Each circle represents a sectors and is situated at the geometrical centroid of it. The sectors on top of each other are shifted a bit to better see them.	56
1.32	Trajectories and occupation of airspace resulting from a run of the ABM. The nodes in red are the “airports”, i.e. the departing and arrival sectors. All the other nodes have size corresponding to their averaged occupation. The red lines shows the trajectories actually flown by the flights; their thickness is proportional to the number of flights using them. This map has been obtained with constant capacity (equal to 5) and one type of company with $\beta/\alpha = 0.0001$.	57
1.33	Trajectories and occupation of airspace resulting from a run of the ABM. See figure 1.32 for a complete description. The companies have all $\beta/\alpha = 1$.	57
1.34	Trajectories and occupation of airspace resulting from a run of the ABM. See figure 1.32 for a complete description. The companies have all $\beta/\alpha = 10000$.	58
1.35	Left: satisfaction against number of flights for different parameters. Right: number of regulated flight plans against number of flights for different parameters. Results obtained with the real network, with constant capacities.	58
1.36	Left: satisfaction against number of flights for different parameters. Right: number of regulated flight plans against number of flights for different parameters. Results obtained with the real network, with real capacities.	59
1.37	Satisfaction against the ratio α/β of the cost function for a given number of Air Operator (50).	59
1.38	Histogram of the crossing times between navpoints (left) and the estimated capacities (right). Crossing times are in minutes. The central panel shows QQ plot of the crossing times between navpoints and the best fit with a lognormal.	62
1.39	Scatter plot capacity proxy equal to the maximal number of flights simultaneously present in a sector (in abscissa) and capacity proxy equal to the maximal traffic load in one hour (in ordinate). Each circle represents a French sector. The blue line is a linear regression. The slope is 2.80 ± 0.16 and the ordinate at origin is 6 ± 30 . The Pearson correlation coefficient is 0.77.	63
1.40	Distributions of degree (left panels) and topological length of flights (right panels) for $N_{sp} = 1$ (top panels) and $N_{sp} = 10$ (bottom panels). For each panel, the real and simulated distributions are shown.	64
1.41	Kolmogorov-Smirnov distance between real and simulated distributions of degree, strength, weight and topological length of flights of the navpoint network as a function of N_{sp} .	65
1.42	This map is the result of a run of the ABM on the real network. The blue patches represent the sectors and the round points are navpoints, their colors indicating the sector they belong to. The thickness of the red edges is proportional to the traffic.	66
1.43	Fraction of flight plans, computed as the number of flight plans rejected divided by the total number of flights, against the standard deviation of the noise on desired departure times. The air companies have all a high β/α ratio. The error bars are the standard errors.	67
1.44	Variations of the fraction of rejected flight plans (left), the fraction of rejected flights (center) and the satisfaction (right) with the logarithm of the ratio β/α .	68

1.45 Variations of the fraction of rejected flight plans (left), the fraction of rejected flights (center) and the satisfaction (right) with the logarithm of the ratio β/α	68
1.46 Evolution of the fraction of regulated flight plans (left), the fraction of regulated flights (center) and the global satisfaction (right) versus the fraction of company S in the system. Error bars are standard errors.	69
1.47 Evolution of the fraction regulated flight plans for company S (left panel) and company R (top panel). Error bars are standard errors.	70
1.48 Evolution of the fraction of regulated flight plans (left), the fraction of regulated flights (center) and the satisfaction (right) versus the fraction of company S in the system. Error bars are standard errors.	70
1.49 Evolution of the fraction regulated flight plans (left panels) and the satisfaction (right panels) for company S (top panels) and company R (bottom panel). Error bars are standard errors.	71
1.50 Portions of the French airspace where a sector has been shut down. Left: the initial traffic. Right: the traffic after the shock.	72
1.51 Satisfaction of companies S and companies R as a function of the number of closed sectors.	73
1.52 Drop of satisfaction of the whole system when a sector is closed. Left: satisfaction is computed on all flights. Right: satisfaction is restricted to flights which do not depart or arrive in the given sector.	73
 2.1 Visualization of the navigation points active during the whole AIRAC 334 in the selected sector. The points in blue are the real one derived from the M1 last-filled plan, those in red are randomly generated.	78
2.2 The management of the trajectory is done within a $\Delta t = 240$ secs time horizon. The flights are sampled every $\delta t = 1/3\Delta t$ secs by linearly interpolating if necessary.	79
2.3 The Fig. illustrates the techniques of rerouting after the rescheduling of velocity. Every possible rerouting is tested with the genetic algorithm for the velocity. This would define a point P_{end} . The route selected will be the one such that minimizes the fitness \mathcal{F}_2 , i.e. it minimizes the distance between the point P_{end}^* of the original route and the P_{end} of the new route. The crossed circles identifies either the shocked area or the trajectory conflict. Circles identify true navigation points. Squares identify randomly generated navigation points.	80
2.4 Illustration of the way the fitness function \mathcal{F}_1 is computed. The green points are those whose distance is below threshold.	80
2.5 Illustration of the way the velocity change sub-module. The crossed circles identifies either the shocked area or the trajectory conflict.	82
2.6 Projection of the LIRROV sector on a map including the navigation points. Major routes and their directions are indicated by the blue, red, orange and black arrows while the critical areas emerging from their intersection are marked with red circles.	84
2.7 Main metrics used to calibrate the model: the frac (top-left panel), fork (top-right panel), average delay of the considered trajectories on 6 May 2010 (bottom-left panel) and average delay of the considered trajectories on the whole AIRAC (bottom-right panel).	85

2.8	Correlation between fracs profile observed in the different days of AIRAC 334. The left panel shows the distribution of all correlation coefficients. The central panel shows a contour plot of the correlation between any pair of days in AIRAC 334. The white color is assigned when the absolute value of the correlation coefficient is smaller than $1/\sqrt{14}$. The right panel shows a contour plot of the correlation coefficients after we have ordered days according to the results of a hierarchical clustering procedure applied to the different days of the correlation matrix.	86
2.9	Distribution of percentage difference of velocity (M3,M1). The two vertical red lines indicate the best x_{min} for the two tails, selected by using the procedure sketched above.	87
2.10	Cumulative distribution of delay in the simulated and real data when the distribution of velocities in each trajectory segment is calibrated with velocities maintained by aircraft obtained from real M3 data.	88
2.11	Box-plot of delay in real data for different aircraft type. The shown error bars are the standard deviations. The boxes correspond to the first and third quantile.	89
2.12	Simulations with different maximum angle compared with real data relative to the whole AIRAC 334. In the bottom-left panel we show fracs and in the bottom-right panel we show the pdf of delays for different values of α . In the top panels we show the Pearson correlation coefficient (left panel) between the frac measured in the simulations and the weighted correlation coefficient (right panel), were the weight of each navigation point was given by the the number of aircraft that cross the selected navigation point.	90
2.13	Simulations with different maximum angle compared with real data relative to the same day of the simulation. In the bottom-left panel we show fracs and in the bottom-right panel we show the pdf of delays for different values of α . In the top panels we show the Pearson correlation coefficient (left panel) between the frac measured in the simulations and the weighted correlation coefficient (right panel), were the weight of each navigation point was given by the the number of aircraft that cross the selected navigation point.	91
2.14	Simulations with different values of the probability p by which directs are issued. In the left panel we show fracs and in the right panel we show the pdf of delays for different values of p	92
2.15	Shocked areas of radius 5 NM. In the top panels we show the average and the standard deviation of delay for the three considered distributions. In the bottom panels we show the average number of reroutings and flight level changes for the three distributions considered. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ . The directs module is switched off.	95
2.16	Shocked areas of radius 5 NM. The Fig. shows the radial porosity [15] for the three different considered distributions: uniform in the left panel, power-law in the central panel and gaussian in the right panel. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ	96

2.17 Shocked areas of radius 5 NM. The Fig. shows the frac metric for the three different considered distributions. In the top panel simulated data are compared with fracs (black curve) relative to 06 May 2010. In the bottom panels data are compared with fracs measured over all the AIRAC 334. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. The two left panels refer to the uniform distribution. The two central panels refer to the power-law distribution and the two right panels refer to the gaussian distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ . The directs module is switched off.	98
2.18 Average delay against porosity, for all simulations we have performed so far, i.e. with different values of n , σ or γ and different values of the extension of the shocked area, see appendix E.	99
2.19 The Fig. shows the average (top panel) and standard deviation (bottom panel) delay for the uniform distribution when shocks are temporally concentrated in a time widows of 3 hours.	100
A.1 Average number of shortest paths as a function of the path length for the Delaunay triangulation, triangular lattice and Erdos Rényi random graph. The order (number of nodes) of the graph is 90.	105
A.2 Average number of overlapping nodes found among the shortest paths (starting and ending nodes of a path are not taken into account) connecting two given nodes, as a function of the path length.	106
B.1 Left: probability of first rejection as a function of the number of flight plans submitted for $N_{sp} = 6$, $c = 4$. The blue line is the analytical prediction, the red one comes from simple simulations. Right: average number of flight plans submitted before first rejection as a function of the number of shortest paths, for the analytical expression and the simulations. The green line is the maximum one could obtain (planifications).	109
B.2 Same figure than figure B.1 with an overlap $Q = 1/N_{sp}$ ($k = 1$).	110
B.3 Evolution of the probability of first rejection against overlap and number of flight plans submitted.	111
B.4 Evolution of the probability of first rejection against overlap and number of flight plans submitted in the simulations. The noise becomes very big for high Q and n (overlap and number of flight plans), because the occurrences of a high number of flights plans rejected are rare and thus the statistic poor.	112
B.5 Probability of second rejection against number of flights plans submitted.	112
D.1 On the left the figure illustrates the first mutation operator. It modifies a velocity on an element of population v . As a result all points in P after v_2 will be modified. On the right the figure illustrates the second mutation operator. It modifies two contiguous velocity on v according to Eq. (D.1). One velocity is increased and the other is correspondingly decreased. As a result all points before v_2 and after v_3 do not change. Further details are given in the text.	116
D.2 The figure illustrates the Crossover operator. It acts on two father and mother elements and generate P , the child element. As much as in the biological/genetic crossover P inherits some genetic information from both parents.	117

INTRODUCTION

This document presents the work done in Work Package 2 (WP2) of the project ELSA. The deliverable describes the two layers of the ATM model that ELSA has developed:

- the strategic layer, simulating the interaction between the Network Manager and the Airline Operators.
- the tactical layer, focused on aircraft and controllers' behavior in a single Air Traffic Control (ATC) sector,

Both layers are fully finished and some results are presented, although the code could be easily used to obtain some additional results.

The strategic layer of the model uses two types of agents, air companies and network manager, and uses two levels of spatial description with a network of navigation points (navpoints) and a network of sectors. Air companies arrives in a random order and submit to the network manager a sequence of decreasingly optimal flights plans (succession of navpoints + time of departure) for a given flight. The network manager examines them one by one and, by forecasting the loads of the crossed sectors, rejects those who are overloading the sector (capacity reached) and accepts the best one which does not. The airspace can also undergo some shocks, with the shutdown of some sectors. The air companies have to resubmit some flight plans if they are concerned. Both artificially generated airspaces and real ones can be used with the strategic layer. Networks are generated with the help of Delaunay triangulation on a set of random points, whereas real airspaces are extracted thanks to the ELSA database (see deliverable D1.3). An automatic and quantitative procedure of calibration can be used to assess the values of implicit parameters and to generate empirically validated scenarios.

The tactical layer of the model have experienced some improvements in order to take into account some aspects related to the issuing of directs, the introduction of many more flight levels and the fact that now the re-routed trajectory minimizes the path inside the sector. The main ingredients are clearly stated and we (i) have calibrated the model against real data and (ii) performed a set of simulation experiments relative to the current ATM scenario in presence of areas of the sector closed to air traffic that therefore reduce the sector volume available to aircraft. We consider different possibilities regarding the concentration and size of these areas. The main result of our model is that the larger the porosity the smaller the delay (the greater the anticipation) we observe in the simulated trajectories, where porosity is measured as the available volume divided by the total volume of the sector. This result seems to revert when we start considering shocked areas with larger radius. This might indicate that the automated air traffic management strategies proposed in the model might even outperform the human based strategies, at least as far as delays are concerned.

The document is organized in chapters and appendices. Seven chapters are relative to the strategic layer and four chapters are relative to the Tactical layer. For each layer we have chapters about the general features of the model, its implementation and the results.

EXECUTIVE SUMMARY

This document is the deliverable D2.3 *Calibrated agent-based model – final draft*. The ELSA Agent Based Model was developed to reproduce some of the stylized facts observed in the Air Traffic Management of the European airspace. The model itself had two main parts: (i) The strategic layer, focused on the interaction between the Network Manager and the Airline Operators and (ii) the tactical layer, focused on aircraft and controllers behavior in Air Traffic Control (ATC) sectors. The deliverable describes the final version of the two layers of the model: the strategic and the tactical one.

The ABM development was driven by a criterion of simplicity, in order to model only the aspects that are strictly necessary to replicate the selected real features. Validation activities were carried out as much as possible to calibrate both layers on ATM real features.

The strategic layer of the Agent Based Model simulates the submission of flight plans by the air companies to the Network Manager, who is in charge of computing the loads of sectors and reject flight plans if capacity of some sector is exceeded. Some sectors can also be randomly shut down after the first round of submissions, which triggers an additional round. There are two versions of the model. The simplest version considers only the sector network and allows to perform extensive exploration of the subtle interplay between the strategies of the air companies, the structure of departing patterns, and the geometry of the airspace. The second version of the model is more realistic and considers also the navpoint network.

The major improvement of the strategic layer, as compared to its previous version, is the use of a navpoint network on top of the existing sector network. Flight plans are then described as a succession of navpoints, which is much more closer to reality. The sector structure is kept to allow the computation of the load and the possibility of overloading when the capacity is reached. The latter is now computed in a more realistic way. Combined to an automatic quantitative procedure of calibration, this allows the simulation of the strategic layer on a real airspace and the generation of empirically validated scenarios. The tactical layer of the Agent Based Model aims at describing the interactions between flights and controllers in a single Air Traffic Control (ATC) sector. This model was calibrated against real data and used to perform current ATM scenario simulations. For the Tactical layer, the main advances regard the implementation of a new rerouting strategy that minimizes the path inside the sector and the issuing of directs.

The model is used to run simulation experiments of the current ATM scenario in order to investigate how the model performs in presence of areas of the sector closed to air traffic. We simulated the case when airspace areas are closed to air traffic, for example for adverse weather conditions, or military maneuvers. We consider different possibilities regarding the concentration and size of these areas.

The main result of our model is that the larger the porosity, the smaller the delay. The model even generates trajectories that are more optimized than planned ones in terms of the total time needed to cross the sector. This result seems to revert when we start considering shocked areas with larger radius.

1 Strategic layer

1.1 Introduction

This part of the document describes the strategic phase of the ABM of the ATM. In the strategic phase, we model the interaction process between air companies and the network manager (CFMU) in order to choose the allocation of flights in the airspace, given the constraints due to capacity and/or safety restrictions. More specifically, in this phase of the model, we assume that a specific airspace is given, together with its navpoint and sector structure, airport location, and sector capacities. Then air companies submit to the network manager a number of possible flight plans for each desired flight. The way in which air companies select the flight plans is driven by an optimization process, where a company-specific cost function is used. The network manager then selects which flight plans are viable, given the current constraints and allocations of the airspace. Finally, the air company selects the best among the accepted flight plans.

The strategic ABM model has three main components, namely the airspace structure, and the two types of agents, i.e. the air companies and the network manager. In the following we shall describe in some detail these three components. For each of them, in section 1.2, we first give an overview of their real main characteristics. Then we give for each characteristic a brief description of the modelization we have chosen for it. In section 1.3 we describe in detail the implementation of the model, with the choice of parameters, function, etc. In section 1.4 we highlight the differences between the current model, which we call 2.0, and the previous one, that we call 1.0. We explain also why we keep it as a separate branch. In section 1.5 we make a list of the important parameters in the model(s). In section 1.6 we give the results obtained with the model 1.0. It gathers the results we obtained by using artificially generated (i.e. model) networks and also the calibration of the model by using real data. To this end we consider the case study of the airspace of France. We make use of the stylized facts detected by the empirical analysis performed in WP1 of ELSA and present in deliverable D1.3. We also needed to extract new statistical regularities on the crossing time of sectors and on the sector capacity. With these new empirical results we present the outcome of the model simulation calibrated on real data. Finally, we present in section 1.7 the results obtained with model 2.0. They are concerning exclusively a setup on the real airspace. After the extraction of the same type of input from WP1 that we used for model 1.0, we use a quantitative calibration to assess one of the implicit parameter of the network. Then we present the results obtained with this setting.

Finally we point out that in this deliverable we preserved most of the material present in deliverables D2.1 and D2.2, even if we reorganized the material. In particular, some old material has been moved to dedicated appendices. The current version of the model is improved with respect to the original version in several respect. The major shift has been to include the structure of the navpoint network on top of the sector network. This allows a more precise definition of the flight plan, as it is in reality. The times of travels are also much more realistic, given that they are computed on the navpoint network. We also included a procedure of “shocks”, where we study the resilience of different types of companies when we shutdown some sectors. Added to this, we changed the definition of

the capacity, so as to fit to the real definition. Finally, we implemented a procedure of quantitative calibration which can find the best agreement of a metric with a given target, the latter being computed from data for instance.

1.2 Model 2.0 - General features

1.2.1 Airspace structure

1.2.1.1 Generality

The airspace structure in its generality is pretty complex. The airspace is **structured hierarchically**, with several scales: transnational, national, ACC, pieces of ACCs, sectors, navigation points, etc.. However, we believe that, as far as the trajectories are concerned, once the airspace is fixed, the primary components of the strategic phase are the sectors and the navigation points. The latter determine the fine structure of the flight plans, while the former represent the basic unit of air traffic control.

For this reason we will model the airspace at two levels: the sector level and the navigation point level. We represent the airspace as the superposition of a network of navpoints and a network of sectors (see Section 7.4 of the Deliverable 1.3 of ELSA). In the former each node is a navpoint and links represent the possibility of going from one navpoint to another without passing through an intermediate one. In the network of sectors, each node is a sector and two nodes are linked if it is possible to go directly from one sector to the other one. This essentially means that two sectors are linked if they have a common boundary. The two networks are related one to each other because each navpoint belongs to exactly one sector, while clearly more than one navpoint is present in a sector.

As discussed in the validation analysis (see Section 1.2.4) when a Air Operator prepares a flight plan, it consists essentially in a path on the network of navpoints. This means that the flight plan contains the time (and the altitude) at which the aircraft is supposed to pass through each navpoint. This information¹ is then used by the Network Manager to infer the entry and exit time of an aircraft in and out of each interested sector. The Network Manager uses this information to check whether the maximum sector capacity is reached and then approves or rejects the flight plan. Thus from the perspective of the Network Manager a flight plan can be seen as a path on the network of sectors, i.e. a succession of sectors, with timestamps of entry and exit. As a consequence, several trajectories in terms of navpoints can have similar flight plans (same sector path but slightly different times of entrance for instance). As we will see below when we will explain the details of the model, it is important to model the choice of the best flight plans in such a way to avoid that they are very similar in terms of sector path because this increases significantly the risk that a large fraction of these flight plans are rejected by the Network Manager.

Clearly the structure of the navpoint and sector network and their interdependence is critical for the modeling of the strategic phase. In Deliverable 1.3 we investigated in details the properties of the networks of navpoints and sectors in the European airspace and in the following we will use this information as an input of our model, together with artificially generated networks with statistical properties similar to the real ones. In Figure 1.1 we show the network of navpoints in France (top panel) and the network of sectors in France (bottom panel, also present in Deliverable D1.3).

It is important to notice that the airspace is by nature **three dimensional** for nearly all the components². This fact could have many important consequences, for instance that the flights can take “short-cuts” in the network of sectors, which is impossible in a purely planar graph³. However, due to the strong anisotropy (the vertical dimension is very different from the other ones), we believe

¹In reality (see Section 1.2.4) the flight plan that the Air Operator gives to the Network Manager does not contain all the details of the flight plan, but only the planned departing and landing time, together with the cruise altitude. The Network Manager reconstructs the detailed flight plan from this information. For the purpose of our model this difference is not important and will not be considered.

²The only exception are probably the navigation points which are defined only with a latitude and a longitude.

³A planar graph is a graph which can be drawn on the plane in such a way that no edges cross each other.

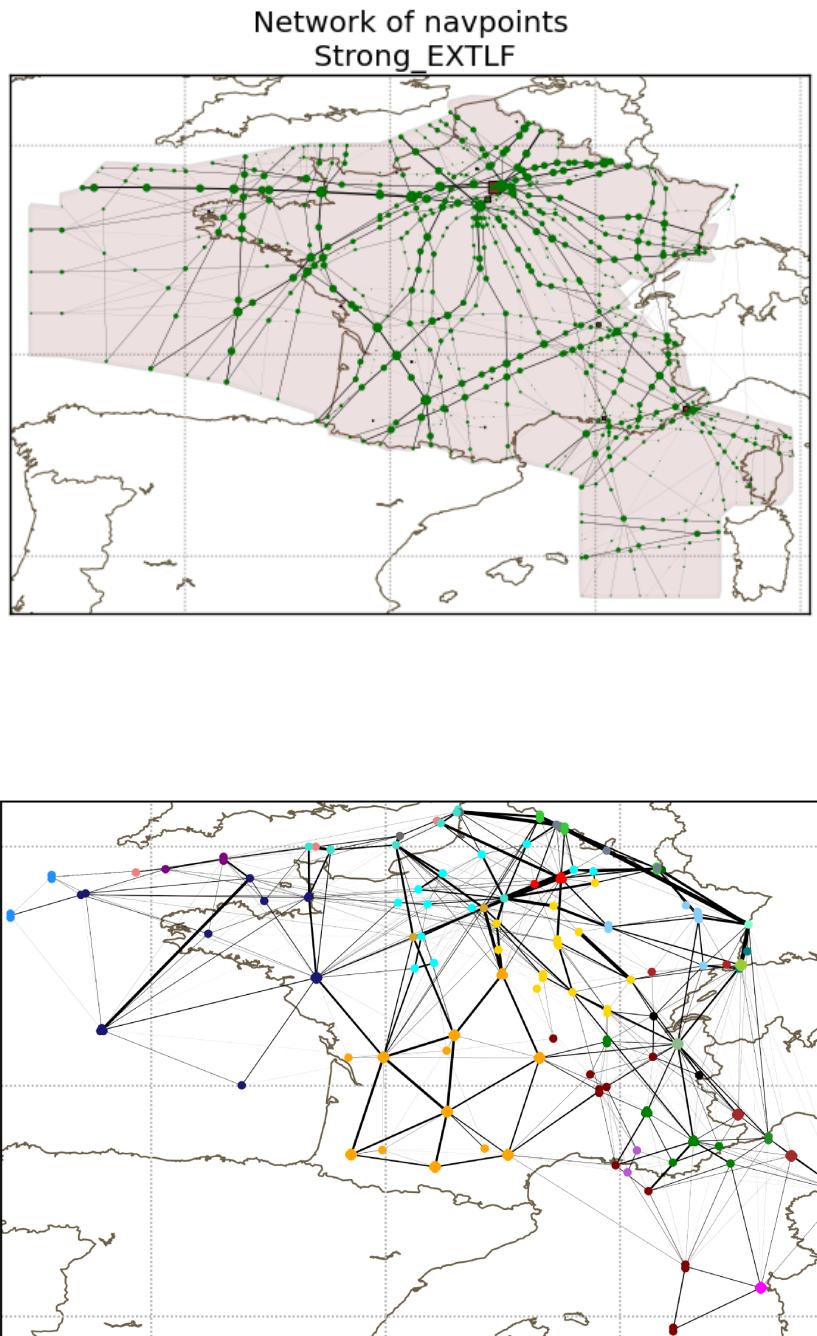


Figure 1.1: Traffic network of navpoints (top panel) and sectors (bottom panel) for France : each navpoint/sector is represented by a node. Two nodes are linked if at least one flight goes from one to the other and the link is weighted proportionally to the number of flights. The colors on the bottom panel represent pieces of ACCs the sector belongs to.

that we can consider as a first approximation the airspace as a tiling of the plane, i.e. we will use mainly planar graphs.

The **capacity** of each airspace is a **very complex quantity** in general. For a simple sector, a direct metric for the capacity is the maximum number of flights that can be simultaneously present in it. This is what we used in Deliverable 2.2 and we will use it for the Toy Model 1.0. In reality the most used measure of capacity is the maximum number of aircraft which are present in the sector at any moment during an hour. In the advanced model 2.0 we will use this definition of capacity. Other more complex measures of capacity also consider how the aircraft are distributed in the space inside the sector, their directions, etc., at a given time. Thus, although capacity can be approximated by considering flights present in the sector regardless of their precise trajectory, the real definition may depend on the spatial distribution of aircraft, resulting in heterogeneous values of the capacity.

The air traffic has two sources of dynamics: the traffic itself, with its fluctuations, and the structure of the airspace. In fact the airspace structure changes dynamically as a response to changing traffic or weather conditions. Thus **the airspaces are by nature dynamical**, with collapses and splits of sectors depending on the expected traffic and the real one.

Finally, the **airports** can be considered as part of the airspace, not only because they have some dedicated airspaces around them, but also because they dictate the rhythm of the traffic, based on their own capacity and optimization strategies. These include for example the intraday pattern of departing and arrivals. Moreover the locations of airports are obviously far from random, being dictated by demographic and transport conditions.

1.2.1.2 Present Model

Main characteristics

We now list the characteristics of the model (see section 1.3 for more details). In order to help the reader, the presentation order is the same as the one used in the previous section. Specifically,

- We identified the navpoints and the sectors (elementary sectors and collapsed sectors) as the main components of the airspace. Thus, we built two networks. In the first one, each node represents a navpoint and an edge between two nodes indicates the possibility for a flight to go from one to the other. This is done either by using as input the real traffic flow when we work with the real network, or by performing a Delaunay triangulation (see 1.3.1.1) in the case of an artificially generated airspace. In the second network, each node is a sector, and an edge is present if the two areas of the sectors represented by the nodes have a common boundary. Note that each navpoint is uniquely associated with a sector, so that each trajectory of navpoints can be converted to a trajectory on the network of sectors (but not the other way around).
- The crossing times⁴, i.e. the time needed to go from one node to the other are now fixed at the navpoint level. In the case of a real network we take as crossing time between two navpoints the average crossing time of all the real flights traveling between the two considered nodes. Note also that our navpoint network is directed and therefore for each pair of connected navpoints we have two crossing times depending on the travel direction. In the case of artificially generated networks the crossing time between two connected navpoints is taken proportional to the distance between the two navpoints. The constant of proportionality is then chosen so that the average crossing time matches the one extracted from the data. The average crossing

⁴Note here that the edges of the network are weighted with the crossing times. This is different from network generated from traffic data, like in figure 1.1, where the edges are weighed with the number of flights going from one node to the other in the given times window.

time between sectors can then be computed based on the underlying navpoint network, but they are not explicitly used in the model.

- As stated above, we choose to work with a bidimensional network, because we believe this is enough to grasp the main characteristics of the network. In this way companies cannot use the altitude in their optimization process (see section 1.2.2). In fact, in the simulations the sectors are equivalent to a vertical stack of real sectors.
- We considered the simplest case where the capacity can be described as a simple scalar depending only on the node (sector). Moreover, in the artificially generated networks we use most of the time the same capacity for each node. For the real network, the capacities are extracted from the traffic data, by considering the maximum number of aircraft in an hour *observed* in a given time period. Clearly this is an underestimation of the real capacity (if sectors always operates below the maximum capacity). It seems however that value is not far from the real one, see section 1.7.1.2.
- We chose not to consider the dynamics for the airspace. Hence, the network is static, each node (sector) being defined for the whole duration of the simulation. However we investigate simulations in which one or more sectors can be shut down and companies have to resubmit new flight plans to the Network Manager. This is done to investigate the resilience of the system by simulating weather hazards or military requirements.
- In the artificially generated networks, we set airports randomly in the network, allowing flights only between airports which are more distant than a given threshold (in number of nodes). In the real networks, apart from the real airports, we include also entry and exit points of the investigated airspace. This is because each airspace is an open system with “sources” and “sinks” being the navpoints at the boundary of the investigated airspace. Finally, note that airports are defined at the navpoint level, so that we can have several “airports”, e.g. entry points, in the same “sector”. The sectors containing airports have also a dedicated capacity, distinct from their capacity as sector.

1.2.2 Agents - Air Operators

Here we describe one of the two types of agents we use in the ABM, namely the Air Operators (AOs) or air companies.

1.2.2.1 Generality

The air companies are big entities which have very complex strategies of optimization. We focus here on how they are choosing the flight plans submitted to the network manager.

First, the **optimization** process is of course hidden by the companies to the other companies and to the Network Manager. However, it is clear that it takes place at a **global level**, and not for each flight separately. Among the visible consequences of this are the waves (clustering of flights departing time at the same airport), the structure of the airline network (e.g. hub and spoke) and the triangular flights (when an aircraft makes a loop between three cities, instead of two).

Moreover, the optimization **strategies** takes into account the **behavior of the other air companies as well as the network manager**. This process is of course also hidden in practice and might have subtle consequences.

The companies have also **several types of aircraft** with different properties, such as speed, cruise altitude, etc.. They use them depending, in particular, on the number of passengers, thus the optimization might depend on the types and number of aircraft available.

Also, different companies have clearly different optimization strategies, resulting in **various behaviors for different companies**. It is also possible that one company has different optimization schemes for **different flights within a company**.

Finally, a major issue is the presence of a “**time memory**” in the system. Indeed, air companies have a “legacy” in the sense that time slots are allocated not only on immediate considerations but also by taking into account the fact that they were previously allocated to a given company.

1.2.2.2 Present Model

Main characteristics

- We perform a simple, local optimization of the flight plans. This means that a company optimizes the trajectory only for one flight at a time. Optimization takes into account the length of the flight and the departure (or arrival) time, but does not consider the other flight plans’ characteristics. In particular, we do not take into account the fact that the flight plan needs an available aircraft. Indeed, we are interested in the last part of the optimization of the air companies. Once they have computed their need in aircraft, crew and potential passengers, they choose the actual path of the flights, which is what we are modeling.
- There is no interactions between the strategies, a consequence of the fact that the optimization is local, i.e. done separately for each flight.
- We consider all the aircraft as equivalent in terms of speed and flight characteristics.
- We can have an arbitrary number of AOs, possibly having different utility/cost functions used in the optimization.
- There is no time memory, i.e. past allocations of airspace do not have any role in the decision of the Network Manager or in the choice of the AOs.

1.2.3 Agents - Network Manager

We now describe the other agent of the model, namely the Network Manager (NM). This agent is much less complicated than the AOs. Moreover, we distinguish here the real actual characteristics and characteristics one could want to implement in the future in ATM (and for which the model could give us some insight).

1.2.3.1 Generality

Real behavior

The Network Manager is a pretty **passive** agent. It only receives the flight plans given by the companies and accept or reject them. filling the airspace as best as it can. Moreover, the “best” is achieved through **local optimization**, because the flight plans are not submitted at the same time, but sequentially. In particular the NM uses a First Come, First Served strategy.

It is unclear to us whether the network manager is sensitive to a **risk aversion** or not. Implementing a strategy where some flight plans are penalized (or rejected), because they could lead the system too close to its limits in the case some deviation is needed during the flight, would be a interesting simulation exercise, but will not be considered in this deliverable.

Finally, it is important to notice that the NM has actually two roles. It accepts or rejects the initial flight plans, building the so-called **“M0” files**. This is the ordinary administration activity of the NM. However in case of abrupt and unexpected changes of the airspace availability, it also accepts and rejects new flight plans building the so-called **“M1” files** (last filled flight plans). The strategies followed by the NM for the updating could be different from the ones for the initial filling.

Possible improved “real” behavior

The NM could become more **proactive** in its management practice. Instead of simply filling the flight plans one after the other (first come, first served), it could make counter propositions balanced between the submitted flight plans and a **global optimization** of the system. For example, it could try to optimize the allocation globally or in a “batch” system, i.e. by optimizing simultaneously a certain number of flights.

On the same line, it could be risk averse toward situations too close to the limit, at the cost of decreasing of the efficiency, but with the advantage of increasing robustness.

1.2.3.2 Present Model

Main characteristics

- The NM is passive and performs a local optimization. Essentially it answers the AOs by communicating if the flight plan of a given flight is allowed. In case of rejection, the AO submits a new flight plan which is considered by the NM.
- There is no risk aversion in the decision process of the NM.
- The M0 phase is modeled through a first round of submission and allocation of the airspace. Then the network can be disturbed (i.e. some sectors can be shut down) and when it happens, the AOs whose flights are affected by the sector closure resubmit new flight plans. This process leads to the M1 trajectories, which are the last filed flights plans.

1.2.4 Validation of the Strategic Layer

The main operational inputs for the tactical layer have been collected during interviews with Alitalia Flight Dispatchers that work at the Alitalia Operation Center (OCC)⁵. They are the professional figures in charge of defining the flight plans and monitoring the flight execution phase. The Alitalia Operation Center is responsible of coordinating and managing almost 700 flight per day, of which around 70 are long-haul flights. For each of these flights a flight plan has to be produced by the OCC and then submitted and approved by the CFMU. Long-haul flights' planning is handled manually and starts 6 hours before the scheduled departure time while short and medium-haul flights are handled using an automatic procedure. Dispatchers have to intervene only if the system flags an exception. This process starts 2 hours before the scheduled departure time. In both cases flight dispatcher make use of a dedicated software tool called LIDO Flight.

The planning phase starts by collecting information about the flight such as weather at destination and on the route or the aircraft performances and possible limitations and failures on board. On the basis of the information collected a flight plan is prepared by optimizing the overall cost of the flight and by ensuring at the same time the safe execution of the flight. For example, the occurrence of a weather perturbation is considered to be an unsafe event and it will always be avoided even at the cost of travelling a longer route. The costs taken into account always include fuel and ATC fees. Costs related to delays are not taken into account by the software tool but can be evaluated on a case-by-case basis by the flight dispatcher. At this stage no information about other flight trajectories is taken into account. As a result, flight trajectories might be not conflict free. After the flight plans are prepared, manually or automatically, they are submitted according to the ICAO format to the CFMU through a dedicated system (SITA). The ICAO format contains the take-off and landing times and a list of navigation points with the related flight level. The CFMU recalculates the flight plan using their own models. These models differ from those used by Airline Operators. In fact they do not consider the differences in performance that aircraft of the same type may have and they manage the vertical profile of the trajectory in a different way. If the flight plan is rejected the dispatcher is noticed; CFMU gives the reason of the rejection but they do not suggest an alternate solution. Moreover the flight dispatcher is unaware of what other companies are doing. When a flight plan is rejected there is no bargaining between CFMU and the dispatcher. The dispatcher simply submits an updated flight plan and waits for its approval. This process is iterated until the dispatcher has a flight plan approved. Communication between CFMU and the dispatcher takes place almost exclusively through the SITA system. In some case the final flight plan can be discussed with CFMU through a phone call. A schematic representation of the process is shown in figure 1.2. The

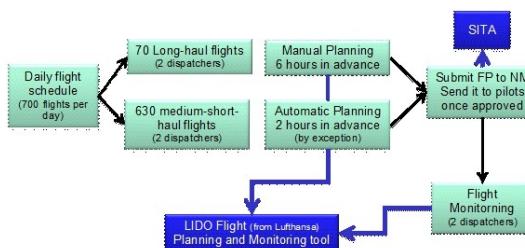


Figure 1.2: The process of flight planning and monitoring in the Alitalia OCC.

information collected that are more relevant for the development of the strategic ABM are mainly related to: (a) the timeframe of the flight plan definition process (6 hours in advance for long-haul flights and 2-hours in advance for medium and short-haul flights); (b) the costs taken into account

⁵This small section is taken directly from the SID paper 2013 [16].

for the flight plan optimization; (c) the interactions between the Flight Dispatcher and the Network Manager and the fact that the flight dispatcher is unaware of other companies strategies; (d) the flight plan submission process, how flight plans are rejected and submitted again for the final approval; (e) the criticalities related to the planning phase such as the exceeding of capacity of one or more sector, bad weather avoidance, partial or total closure of destination airport or unpredictable events like strikes, big events, wars.

1.3 Model 2.0 - Implementation of the model

1.3.1 Networks

As stated in the previous section, our ABM requires a schematic representation of the airspace, which in our case is a superposition of the a network of sectors and a network of navpoints. We will consider both real networks and artificially generated ones. The latter are introduced in order to have some control on the simulation of the model. In fact, with the simple model networks we could achieve some analytical intuition of the results and moreover we can see the effect of changing the network structure on the output of the model.

We considered different types of model networks. Since we work in 2D, the network of sectors should be planar, that is drawable on the plane in such a way that its edges do not cross one with each other. A planar network is also sparse because the number of edges has the same order of magnitude of the number of nodes. The network of navpoints can be in general non-planar. However the two dimensional constraints limits significantly the structure of the network and, given also the relative small degree of its nodes, it can be modeled as a first approximation as planar. However in order to investigate the role of planarity of the networks on the output of the model, we considered also a non planar network.

Specifically, we generated independently two networks on a plane, one for sectors and one for navpoints. An important parameter of this part of the model is the average number of navpoints per sector. We consider different choices, focusing mostly on values close to what observed in real airspace. As far as the structure of each network is concerned, we decided to run our model on three different graph structures to reproduce the network of sectors and navpoints:

- Delaunay triangulation of a set of randomly chosen points (Fig. 1.3, left),
- Triangular lattice, which is a regular graph (Fig. 1.3, center),
- Erdos-Rényi random graph, which is a non planar graph (Fig. 1.3, right).

1.3.1.1 Delaunay triangulation, Voronoi tessellation

Given N points on a plane, a Delaunay triangulation for them is a collection of edges satisfying an “empty circle property”, that is, a subdivision of the plane in triangles such that no element of the point set is inside the circle passing through the three vertices of any triangle (*i.e.* the circumcircle). The minimum of the angles of each triangle, used for the triangulation, results maximized in order to provide the best possible covering of the plane with the minimum number of triangles. This particular triangulation is named after the Soviet/Russian mathematician Boris Delaunay⁶.

An alternative way of defining the Delaunay triangulation of a discrete set of points is through its connection with the Voronoi tessellation. This tessellation, named after the Ukrainian/Russian

⁶B. Delaunay: Sur la sphère vide, Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk, 7:793–800, 1934.

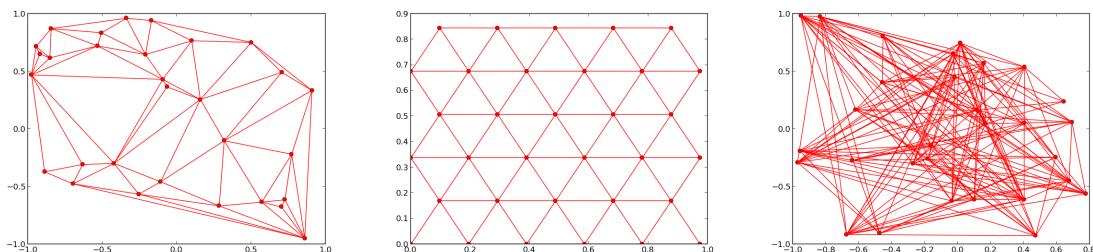


Figure 1.3: Left: example of a Delaunay triangulation of 30 random points in a square region of the plane. Center: example of a triangular lattice of 30 points in a square region of the plane. Right: example of an Erdos Rényi random graph of order (*i.e.* number of nodes) 30.

mathematician Georgy Voronoi, is a way of dividing a plane into N regions. In order to build a Voronoi tessellation, one first chooses a set of points (termed “generators”) in the plane. Then for each one of them one constructs a surrounding tile as the region consisting of all the points closer to that generator than to any other element in the set.

The Delaunay triangulation and the Voronoi tessellation are in a dual relationship. In fact, starting from a Voronoi tessellation one can build another graph that has a vertex corresponding to each tile. For each tile boundary in the Voronoi diagram one then draws an edge joining two neighboring tiles. The obtained graph is nothing but a Delaunay triangulation. The term dual is used because this property is symmetric and works also the other way around. As figure 1.4 shows, the Delaunay triangulation can be easily interpreted in terms of sectors. In fact, each generator point is the centroid of a sector, here defined as one of the tile in the Voronoi tessellation. The triangulation is thus the natural way of building a network of sectors by joining two neighboring sectors.

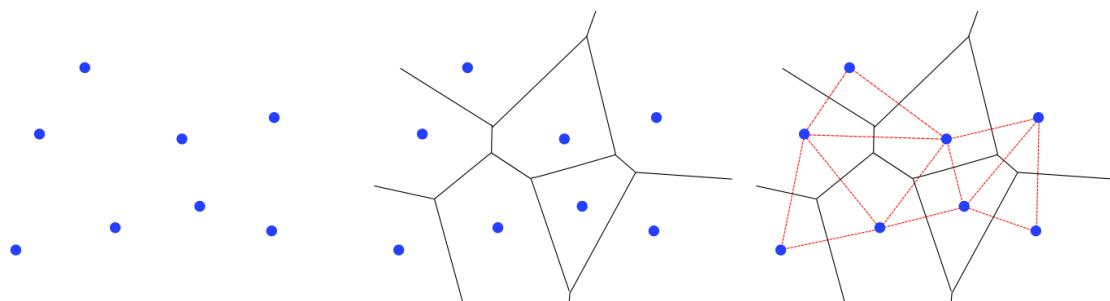


Figure 1.4: Voronoi tessellation of 8 randomly distributed points on plane. First, we draw randomly some points on the plane (left). Then we build the Voronoi tessellation (center) by drawing the perpendicular bisectors between each pairs of points, stopping the line when it crosses another one. All points in one given tile are now closer to the corresponding vertex than to any other vertex (definition of the Voronoi cell). Finally, the Delaunay triangulation can be easily drawn (right) by putting a link between every pairs of vertices which have one common boundary. Voronoi and Delaunay networks are dual and uniquely defined. Moreover, the Delaunay triangulation is the best triangulation one can get for a set of points.

In order to build the Delaunay triangulation we generated a set of uniformly distributed points in a square region of the plane (*i.e.* between -1 and $+1$ both in x and in y). Then we performed the Voronoi tessellation. As stated above, in such a tessellation the points inside the boundaries of each tile are nearer to its centroid point than to each other point in the starting set. The tiles represent

airspace sectors. The Delaunay triangulation is obtained by plotting the dual graph associated with the Voronoi tessellation. Finally this Voronoi tessellation is unique, as well as the Delaunay triangulation. The Delaunay triangulation is clearly a planar graph and each vertex has on average six surrounding triangles (*i.e.* the mean degree is equal to 6, which is also the maximum average degree for planar graph).

To have a planar network to compare with the Delaunay triangulation, we considered also a graph in which the degree of each node is exactly 6. Such kind of regular graph is called a triangular lattice. Note that the triangular lattice is also the Delaunay triangulation of the corresponding set of points – its dual being the hexagonal tiling.

1.3.1.2 Erdos-Rényi graph

Starting from the same set of points used to construct the Delaunay triangulation, we also defined a random graph. Such kind of graph is obtained starting from a set of nodes and adding edges between them at random. Different random graph models produce different graphs. We decided to use the model developed by Erdos and Rényi⁷. In this model the graph is built by drawing an edge between two nodes with a probability p independent from every other edge.

We decided to impose a value of $\langle k \rangle = 6$ as the mean degree of the nodes, in order to have the same degree obtained by the Delaunay triangulation and the triangular lattice. This corresponds to fixing the probability of connection, which is $p = \langle k \rangle / (N - 1)$. In fact, each vertex can have a maximum of $N - 1$ connections. As stated at the beginning of this section, using a random graph as a model for the sector network means to relax the planarity condition. In general, the random graph cannot be drawn in such a way that no edges cross each other.

1.3.2 Crossing times

In our model time is a continuous variable, therefore it is not a model in discrete time. Hence, the crossing times between navpoints (and a fortiori between sectors) are real numbers. This implies that the cost function of a flight plan, which in part depends on the time duration of the flight, is a real number. Note that on the contrary of the previous deliverable, here the crossing times are chosen first in the navpoints network and then computed, if needed, for the sector network.

There are several possible choices for the crossing times. They can be generated:

- By drawing them independently from a given probability distribution centered on a given value. The average crossing time is chosen so that it matches the average value in the real data.
- By choosing them proportionally to the Euclidean distance between nodes (navpoints), the constant of proportionality being fixed by the inverse average speed observed in the real data.
- By drawing them from real data in the case of real networks. In this case we consider the average crossing time between two navpoints by considering all the flights crossing them consecutively.

A flight plan is a vector containing the navpoints connecting the departing and arriving airport, plus a vector of times at which the aircraft passes through each navpoint. From the point of view of the Network Manager a flight plan is a vector containing the sectors traversed by the flight, together with a series of real numbers indicating the times of entrance in and exit from each sector.

We introduce also a parameter, called τ , which is the minimal time step used by Air Operators for shifting the departure time of the flight when submitting the different flight plans. This parameter is usually set to 20 minutes, which is close to the typical duration of the airport slot.

⁷P. Erdős and A. Rényi (1960). "On the evolution of random graphs". *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* 5, 17–61.

1.3.3 Capacities and Airports

In this version of the model, we can choose an arbitrary distribution for the capacities of sectors. Specifically, we can choose:

- The same value for each sector.
- A different value for each sector drawn from a given probability distribution (e.g. Gaussian, uniform, bootstrap from real data etc.).
- A different value based on a specific characteristic of the sector (e.g. area, volume, etc.).
- The real (estimated) capacity of the sector, when investigating the real airspace.

Note that the definition of the load – thus of the capacity too – has changed since last Deliverable 2.2. Previously, the load of a sector was defined as the maximum number of flights simultaneously present in the sector. The capacity was just the maximum load allowed. Now, the load is defined as the number of flights which, in a given hour, has traveled through the sector (including those who are entering and those which were already there at the beginning of the hour). The capacity is still the maximum allowed load.

Finally, we chose a set of airports. In the generated networks, we chose the airports randomly, without constraint. We then compute the possible pairs of trajectories, with the only constraint that the source and the destination cannot be closer than a minimal topological distance, i.e. a number of sectors between the airports on the shortest paths. Note that the airports themselves are navpoints, which allows for several airports in the same sector, since “airport” here has to be understood in a broad sense, including entry and exit points of the airspace. In the artificially generated networks we restrict the number of airports per sector to a maximum of one.

For the real network, we set the airports and the possible pairs of airports (including the boundary points) based on the real traffic.

1.3.4 Air Operators' behavior

The AOs are supposed to perform several tasks in a proactive way. At each time step a company is chosen randomly and it selects randomly a pair of airports and a desired departing time t_0 for the flight (see Section 1.3.4.1 for details on the possible distributions). The AO considers the N_{fp} best flight plans (FPs) according to its cost function. Then it submits them in sequential order from the best to the last to the NM until one is accepted. The accepted FP modifies the availability of the airspace for the next AOs contacting the NM.

The process of choosing the N_{fp} best flights plans is now more complicated than in the previous version of the model (see Deliverable D2.2). Clearly, now the network of navpoints comes into play into the optimization procedure. However the selection of the best flight plans presents some subtleties that requires some preliminary discussion.

Given a source and a destination, the algorithm of the previous version of the model (Deliverable 2.2 and Model 1.0 of the current deliverable) computes the N_{fp} shortest paths on the network of sectors. One might naively think to replace the same procedure in the Model 2.0, by identifying the N_{fp} shortest paths in the network of navpoints. These paths are then converted by the NM in paths in the network of sectors to check their viability, given the current traffic. However, if this is done by using realistic values of the number of navpoints per sectors, one obtains only very few different sector paths, namely one or two for $N_{fp} = 10$. Indeed, since there are typically more than 20 navpoints per sector, it is quite easy to build different navpoints paths (which can differ only by one nodes and have similar length) within the same path of sectors. In terms of traffic regulation this is a problem because if one sector path is not allowed because of a capacity limit, it is likely

that another navpoint path with the same network path (even if with slightly different entry and exit times) will be forbidden. Therefore we need to design a AO's strategy of selection of the N_{fp} best flight plans, which allows to explore a larger variety of sector paths.

In order to do so, we introduced another parameter, N_{sp} , which is the maximum number of navpoint shortest paths for each sector path. For instance, if $N_{sp} = 1$, we compute first the N_{fp} shortest paths of sectors, then for each of them, we restrict the network of navpoints to the nodes contained in the sectors⁸ of that path, and finally we compute the shortest path of navpoints. With this process, we end up in general with $N_{fp} \times N_{sp}$ paths, among which we choose the N_{fp} best ones. Note that even if we take $N_{sp} = N_{fp}$, which in theory gives only one sector path, in reality we might end up with more, because there are not enough navpoint paths taking this sector path. In this case, the algorithm looks for navpoint paths in another sector path.

In order to compute the shortest paths themselves, we used the so called Dijkstra's and Yen's algorithms⁹.

Once we have described how the AO selects the best paths, let us consider now how it chooses the flight plans. First, the AO chooses at random a pair of airports among the available ones and a desired departing time t_0 . Then it computes N_{fp} (spatial) paths to the arriving airport, using the process described above. After that, the AO tries a series of combinations of the previously found paths by shifting them in time. For instance, given one of the shortest paths, it is shifted by one or two time steps of length τ . Each path associated with a departing time becomes a "flight plan". After that, the AO sorts the flight plans by computing the values of a predefined cost function $c(p_i, t_0^i)$. Here t_0^i is the departing time of the i^{th} flight plan and p_i is the spatial path, i.e. the array of navpoints. Note that in general t_0^i is different from the desired departing time t_0 . Finally, it submits to the NM the N_{fp} flight plans one by one from the one with the smallest to the one with the largest cost, until one is accepted, if any (see Fig. 1.5).

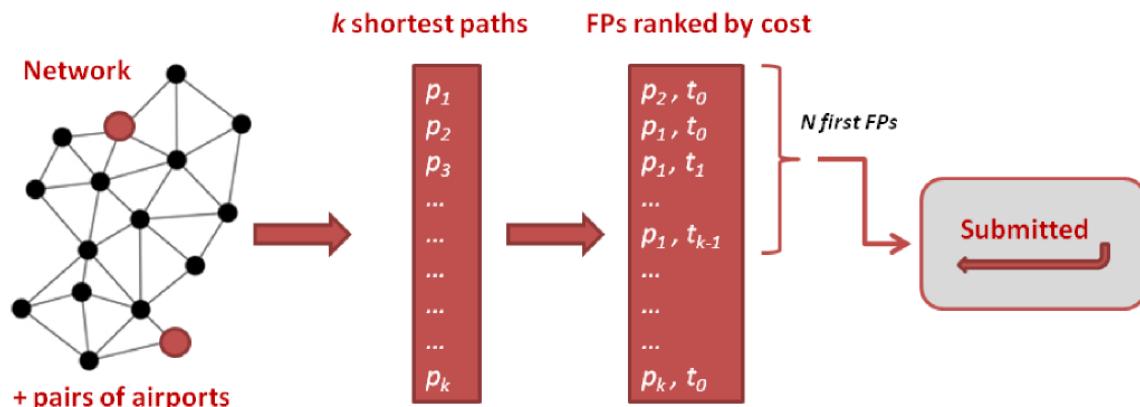


Figure 1.5: The figure shows the N_{fp} flight plans selection operated by an AO: given the network and the flight (as a pair of departing/arriving airports), the N_{fp} shortest paths are determined. Different combinations of paths and departing times are tried and their costs are computed. The combinations are sorted following decreasing cost; the N_{fp} with higher rank are selected to represent the array of proposal for the given AO and for the given flight.

⁸Note that this is actually more complicated than that. In reality, since the restriction can include shorter sector paths, we have to cut the edges between two navpoints which are not in two consecutive sectors.

⁹E.W. Dijkstra (1959). "A note on two problems in connexion with graphs". *Numerische Mathematik* **1**, 269–271. J.Y. Yen (1971). "Finding the k Shortest Loopless Paths in a Network". *Management Science* **17** (11): 712–716.

The AO's cost function depends on two variables:

- The **weighted** length of the path l i.e. the sum of the weights of the edges (i.e. crossing times, see 1.3.2) of the path. This is also exactly the spatial length of the flight, since we assume constant speed.
- The difference between the time of departure t_0^k and the originally desired time of departure t_0 .

The selected AO cost function reads:

$$c(p_k, t_0^k) = \alpha l + \beta |t_0 - t_0^k|. \quad (1.1)$$

The ratio between α and β characterizes the type of AO. In fact, a company that values most the length of the flight (for example, because of fuel consumption) will have a cost with an high value of α compared to β . For example a low cost carrier without a hub and spoke structure would probably be described by such a cost function. On the contrary, a company that values most the punctuality will have an high value of β . This is the case of a major carrier with a networked hub and spoke structure of flights. Since the cost function is a linear combination of two terms and due to the fact that what matters for the process is the ranking of the costs of the flight plans, what really matters is only the ratio between α and β . For this reason in most of the following analyses we will set arbitrarily $\alpha = 1$ and we vary β .

Finally, note that for any company the best flight plan is always the one with the shortest path and the no time shifting ($t_0^i = t_0$).

1.3.4.1 Times of departure

The setup of the desired times is an important component of the model. In fact, we can set an arbitrary distribution for the desired departing times t_0 during the day. Four cases are implemented in the model:

- All flights have the same desired departing time.
- Desired departing times are drawn from an uniform distribution between two given times (e.g. the beginning and the end of the day).
- Desired departing times are drawn from a distribution with a certain number of peaks.
- Desired departing times are taken from the real data.

For the last case, notice that we do not have direct information on the desired departing time, but only the actual last filed flight plan (M1 file). The departure times in the data *are not* the desired times, but merely the already regulated times. We use the departing time as a proxy of the desired departing time. However since the M1 file are regulated to avoid capacity overload we also implemented a module in which the desired departure times are equal to the actual departing times plus a noise term. This is taken from a normal distribution centered in zero and with standard deviation δt_0 .

1.3.4.2 Diversity of companies

We have designed the ABM so that we can put an arbitrary number of types of Air Operators, each of them having a different cost function, i.e. different parameters α and β . These parameters are permanently associated with the type of company. One can see (and numerical simulations confirms

it) that the ratio β/α of the cost function is in fact the only key parameter of the cost function. We can therefore compute the statistics when only one type of company is present. More interesting is the case when two or more different types of companies are simultaneously present in the airspace, because, as we will show in the results section, the degree of satisfaction of a type of company depends on the fraction of the other types of AOs present in the airspace. Moreover by mixing different types of air companies it is then possible to see if it is possible to increase the global satisfaction.

We mainly focused on two extreme types of AO strategies:

- the first one has a high ratio β/α , meaning for these companies it is very important to be on the desired time. Hence, they prefer to choose an alternative route rather than shifting in time. This type of companies will be called “company R” – for “rerouting” – in the following.
- the second one has a small ratio β/α , resulting in a preference for a shifting in time if the previous flight plan is rejected. These companies will be called “company S” – for “shifting”.

Note also that, for any ratio, the best first flight plan is the one with the shortest path and with a time of departure equal to the desired time.

1.3.5 Network Manager's behavior

In our model the NM acts by defining a random queue among the AOs. Their proposed flight plans are considered following the order settled by the queue. The NM reacts to each AO proposal by checking which FPs do not cross full sectors. In practice, this is done by checking one FP after the other beginning with the best one (according to the AO). The first allowed FP is allocated to the airspace, but, of course, it can happen that there is no accepted flight plans. It is important to note that one does not need to assume that the NM knows the ranking of the company, but simply tests all the submitted FPs, communicating to the AO which are allowed, and the company will then choose among them the one with the lowest cost.

In the case where a sector is shut down, the NM finds out which companies are impacted and asks them to resubmit a new set of flight plans on the new network (i.e. the old one without the sectors which are shut down). It then recreates a queue, keeping the priority defined in the original queue, and checks for each flight plan, as before.

1.3.6 Metrics

We describe here the metrics we used to measure the performance of the model. We discriminate between metrics that describe the level of regulation of the system and metrics that measure the degree of satisfaction of the different air companies.

1.3.6.1 Regulated flight plans

The first metric we considered is the number of flight plans which are rejected by the network manager. More precisely, we measure the average number of flight plans rejected per flight. In a well performing system this number should be as small as possible. Clearly this quantity gives also a measure of the degree of agreement between AO desires and what the NM is able to give them. In fact the more flight plans are rejected, the less likely it is that the chosen flight plan is good for the AO. In fact flight plans are sorted by increasing cost.

1.3.6.2 Regulated flights

This second metric is the average of a binary variable. We simply count the number of flights for which all flight plans have been rejected. In other words, we consider a flight as non-regulated if at least one of its N_{fp} flight plans has been accepted. This measure is complementary with the previous.

1.3.6.3 Satisfaction

In addition to the two previous metrics, we also consider a quantity related to the cost. Even if the absolute value of the cost has a relationship with a real cost, the increase or decrease of this function could be of interest to better understand several issues.

For each flight f , we define the satisfaction of the AO as

$$S_f = c_f^{best} / c_f^{accepted}, \quad (1.2)$$

where c_f^{best} is the cost of the optimal flight plan for the flight f according to the AO cost function (this flight is also the first flight plan to be submitted for the flight), and $c_f^{accepted}$ is the cost of the flight plan accepted for this flight. If no flight plans has been accepted, we set S_f to 0. Note that S_f is then always between 0 and 1. The value 1 is obtained when the best FP is accepted.

The satisfaction is a metric defined for a single flight. We also define the average satisfaction across all the flights of a given AO, i.e.

$$Satisfaction[AO] = \frac{1}{M} \sum_{i=1}^M S_f^{(i)}, \quad (1.3)$$

where M is the number of flights of the air company AO and $S_f^{(i)}$ is the satisfaction of the i -th flight of the company.

When more than one AO is present in the system, the definition of global satisfaction is more subtle. We define two new measures of satisfaction: a *Global satisfaction* (GS) and a *Normalized global satisfaction* (NGS). Let us consider the simple case of two air companies, AO_1 and AO_2 , and assume that there is a fraction f_1 of flights belonging to company AO_1 and a fraction f_2 of flights belonging to company AO_2 . Finally, let us call $Satisfaction[AO_1]$ and $Satisfaction[AO_2]$ the average satisfaction of the two AOs.

The global satisfaction is the sum of the weighted satisfactions of the different air companies, where the weights are given by the corresponding portions of flights, i.e.

$$GS = f_1 \times Satisfaction[AO_1] + f_2 \times Satisfaction[AO_2]. \quad (1.4)$$

In the normalized global satisfaction, instead, we consider the satisfaction of an AO by normalizing it over the one computed when the “rival” company is not present.

$$NGS = f_1 \times \frac{Satisfaction[AO_1]}{Satisfaction[AO_1|f_2=0]} + f_2 \times \frac{Satisfaction[AO_2]}{Satisfaction[AO_2|f_1=0]}. \quad (1.5)$$

As it will be clear in Section 1.6.2, NGS is a better measure of global satisfaction than GS because it is insensitive to the different level of satisfaction observed when each AO is alone.

1.4 Model 1.0 - Toy model

The model we call 1.0 is an updated version of the one described in the previous deliverable (D2.2). It is a simpler version of model 2.0 with less features. The reason why we keep it as a separate model is precisely to be able to understand some characteristics of model 2.0 by seeing for instance if model 1.0 displays them. This model is useful as toy model, as it reduces the complexity of the code, including the number of parameters. In this section, we briefly describe the main differences with model 2.0.

1.4.1 Airspace structure

The main difference between both models lies in the structure of the airspace. Model 1.0 use a network of sectors, without any underlying navpoint network. Hence, crossing times have to be drawn either from a given normal distribution or computed with their distance from each other (we use the sector centroids for this). Concerning airports, model 1.0 does not have dedicated capacity for the airports. Finally, the capacity is defined, as in the previous deliverable, as the maximum number of flights present in the sector at any time.

1.4.2 Airline Operators

Airline Operators are almost the same as in Model 2.0. Only the computation of their shortest paths is different, because it is based only on the sectors. Thus, they take the N_{fp} shortest paths and try to shift them in time.

1.4.3 Network Manager

The network manager acts exactly the same as in Model 2.0, only its definition of the load and the capacity, as explained before, has changed. Note that in addition to the model presented in deliverable D2.2, it can manage some shut down of sectors like in model 2.0.

Note also that model 2.0 has an explicit time scale: every time crossing, for instance, are expressed in minutes. Model 1.0, on the other hand, is calibrated on the typical crossing time between sectors, which is set to 1.

1.5 List of parameters

Here we list all the parameters used in the models. There is an indication when the parameter is not present in model 1.0

1.5.1 Parameters linked to the network

For a generated network:

- Number of sectors,
- number of navpoints per sector (not in model 1.0).

For a real network:

- Area (country for instance),
- date,
- filter.

In both cases:

- Number of airports,
- crossing times of edges (time between two consecutive navpoints in model 2.0, time between two adjacent sectors in model 1.0)
- capacity of airports (not in model 1.0),
- capacity of sectors,
- minimum distance (in sector) between two airports potentially linked by a flight.

1.5.2 Parameters of the simulation

- Number of flight plans submitted for each flight, N_{fp} ,
- number of paths of navpoints per path of sector, N_{sp} (not in model 1.0),
- time shifting step for the flight plan, τ ,
- type of departure times pattern, and parameters associated (width and time Δt between peaks etc.),
- standard deviation δt_0 of the Gaussian noise added to the real departure times (not in model 1.0),
- number and types of companies, i.e. set of parameters (α, β) .
- number of sectors which are shut down.

1.6 Results of Model 1.0

In this section we present the results of the numerical simulations of the strategic ABM obtained with model 1.0. Since our objective here is to have an understanding of the model, also by taking into account that the model 1.0 has a similar structure of the more complex and complete model 2.0, presented in the next section, we will present the results by considering progressively more complex versions of the model 1.0. Specifically, we will start from the simplest case with only two airports, neglecting the weights of links (e.g. all the sector are crossed in exactly the same time), all the aircraft with the same desired time and only one type of air company. We will then start to make the model more complex, by adding weights to the links of the sector network and then by considering mixed populations of two types of air companies. These second additions create a much richer phenomenology, because the satisfaction (or fitness) of a company now depends on the composition of the population of the other companies, in a game theoretical framework. We then show that the desired departure time pattern plays a very important role in deciding which company is fittest. This is a typical case where changing the environmental conditions (the time departing pattern) can change significantly the pecking order of the different strategies. We will also consider the resilience properties of this system and of the different strategies to shocks, here modeled as unexpected closure of a set of sectors. Finally we shall consider two further generalizations of the model: in the former we show how the results obtained depends on the geometrical configuration of sectors and location of the airports, showing again an interesting interplay between structure and strategy. The second generalization is a complete simulation of a setting where more airports are simultaneously present and we will consider the general case of mixed population and realistic departure time patterns.

The second part of this section is devoted to the issue of calibration of the model 1.0 with real data. We shall consider the case study of France and we discuss how to calibrate the different parameters of the model (network structure, capacities, crossing times between sectors, etc.) from the DDR data. We then run a full fledged model on the network of sectors of France and we compare the results with those obtained with artificially generated networks.

1.6.1 Basic Mechanisms

One preliminary question we want to address is how many flights can be allocated to the airspace before one of the flight plans is rejected by the NM because the flight plan would violate the maximum capacity of some sector. Since an AO submits up to $N_{fp} > 1$ flight plans, we can elaborate our question by asking how many flights can be allocated to the airspace before the first, second,... N_{fp} -th flight plan is rejected by the NM. We also consider the number of regulated flights and flight plans, as described in the previous sections. Finally, we consider the different satisfaction metrics, namely the single company satisfaction and the global satisfaction, in the case of mixed populations of more than one type of company.

Clearly all these metrics are stochastic and depend on the specific realization of the noise. This is due to the fact that the order of arrival of companies is a random process. We therefore perform a large number of simulations and we average the results. We consider different types of cost functions of the AOs in order to see their role in the number of flights that can be allocated in an airspace.

1.6.1.1 Unweighted case

We perform the first simulations of the model by considering the case of unweighted links, *i.e.* the crossing time of all the sectors is the same. In order to contain the simulation complexity, we decided to choose a network topology and to test the ABM on it. To this end we used a Delaunay triangulation with 60 nodes (Fig. 1.6) and we fixed on it only two airport sectors at the same time.

We simulated 30 identical AOs, each of them submitting up to $N_{fp} = 10$ FPs for each flight. We performed two runs of the model (of 100 iterations each) by using airport distances of 4 and 6 edges, respectively.

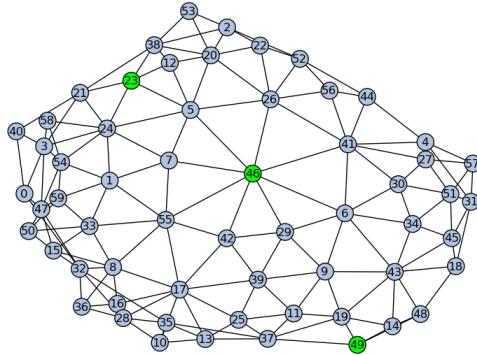


Figure 1.6: Graph used for running the model in the unweighted case. The graph is a Delaunay triangulation of 60 randomly chosen points on a plane. Green nodes represent sectors with airports. In the simulations, node 49 is taken as departing node, while node 46 (at a distance 4) or node 23 (at a distance 6) are the arriving ones.

After running the model, we obtain as output a vector in which the i^{th} element stored the last rejected FP of the array of proposals of the i^{th} company. In Fig. 1.7 we plot the average position in the array (average rejection time step) as a function of the position in the array of the FP which was rejected. We considered three types of cost function, namely one AO considering only departing time ($\alpha = 1$ and $\beta = 0$, red lines), one AO considering only path length ($\alpha = 0$ and $\beta = 1$, green lines), and a last one with a mixed behavior ($\alpha = 1$ and $\beta = 1$, blue lines).

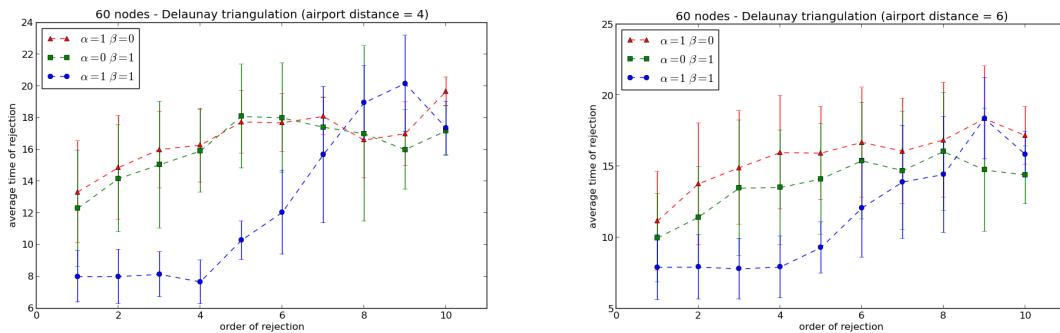


Figure 1.7: Left: rejection statistics for a simulation with 100 AOs flying on a 60 node graph. The chosen airport sectors are node 49 and node 46 of Fig. 1.6. Right: rejection statistics for a simulation with 100 AOs flying on a 60 node graph. The chosen airport sectors are node 49 and node 23 of Fig. 1.6.

We note that the red and green dashed lines follow a similar behavior and are consistent one with each other. The blue line, instead, exhibits an almost equal time of rejection for the first 4 flight

plans. Moreover it is lower than the ones found by using different parameter configurations of the cost function. The blue line then starts to increase, reaching the other curves around the 7th flight plan. This can be explained by considering that the algorithm, after the (*path, departing time*) pair choice, produces a shuffling among the FPs which have the same value of the cost function. This is done to prevent generating different AOs having too similar arrays of FPs. This could, in principle, lead to have FPs of given orders never being the last ones rejected.

For the configuration ($\alpha = 1, \beta = 0$), different FPs on the same path, although with different departing times, share the same value of the cost. This occurs because in computing the cost function the AO only takes into account the length of the flight. So, having different paths with the same length and shifting their time of departure, only lead to a multiplicity of FPs with same costs. A similar effect occurs for the parameter configuration ($\alpha = 0, \beta = 1$) when one has different paths with different lengths but with the same departing time. In this case it is the temporal part which does not enter the computation of the cost function. In the mentioned situations, the shuffling is made over a large number of flight plans, so it is not immediate to find FPs of different AOs sharing a same node at the same time. This effect prevents early time rejections. The case in which the cost function parameters are ($\alpha = 1, \beta = 1$) takes into account both path length and departing time and so only small sets of FPs share a same value of the cost function; accordingly, the shuffling is performed on these small sets and finally it is easier to find space-time overlapping among FPs of different AOs. For this reason the rejection occurs earlier in respect to the other cases.

In Appendix A we present some analytical calculations that helps understanding the observed simulation results. This is done by performing a combinatorial calculation on a simplified version of the air traffic allocation problem, which has the same characteristics of the simulated model.

1.6.1.2 Weighted case, rejection and regulation

After working with the simplest case of an unweighted network, and in order to account for different sector crossing lengths, we attribute weights to the edges of a 90 nodes Delaunay graph. The weights are drawn from a normal distribution with mean equal to 1 and standard deviation equal to 0.1. Performing such a choice allows us to have different sector crossing times without changing too much the contribution that an edge gives to the total length of the path between two airports in comparison to the unweighted case. Figure 1.8 shows the network we used in our simulations. In the figure we highlight the two nodes chosen as airports and the thickness of the lines gives information about the weights.

Rejection and regulation statistics

We run the ABM on this network and we plot the average time at which, for the first time, a given order of FP is rejected. We do this by using different configurations of α and β at a fixed topological distance (figure 1.9).

The ratio α/β leads to quite different behaviors of the curves, influencing the ranking of the FPs. If α is higher than β an AO will prefer to always fly along the shortest path and will shift forward in time the FP. This produces the curves in the upper part of the plot. On the contrary, when β is higher than α the behavior is the one which can be seen in the lower portion of the plot. Here the overlap among the paths plays a crucial role. In fact FPs with overlapping flight paths are more likely to be early rejected. After filling the first 5 FPs (5 being the capacity of the sectors), one sector in the path reaches the capacity and no more FPs can be filled on the same path, so one has to switch to the higher order FPs in the bench.

It is possible to perform a calculation for the probability of inclusion of a FP in the FP bench. Consider a number of flight plans $N_{fp} = n$ and two airports at topological distance ℓ , an air company with cost function $\alpha L + \beta \Delta t$, where L is the weighted length of the path and Δt is the shifting time.

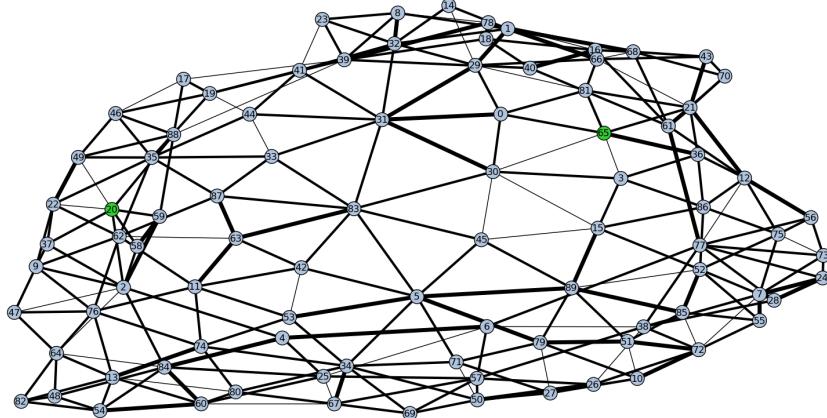


Figure 1.8: Graph used for running the model in the weighted case. The graph is a Delaunay triangulation of 90 randomly chosen points on a plane. Green nodes represent sectors with airports. The thicker is the edge, the higher is the associated weight.

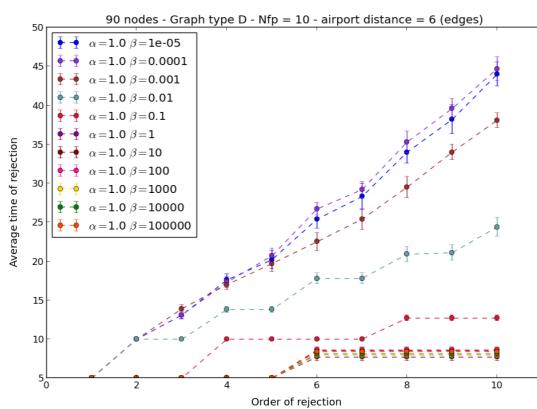


Figure 1.9: Average time of the first rejection in function of the order of rejection for different configurations of the cost function parameters α and β . The graph used for the simulation is a Delaunay triangulation of 90 points and the topological distance between the starting and the arriving airports is 6 edges.

For simplicity let us assume here that $\Delta t = \tau = 1$. The flight plans considered in the optimization part include the n non time-shifted shortest paths and the time shifted version. Thus the costs are

$$\begin{array}{llll} \alpha L_1 & \alpha L_1 + \beta & \alpha L_1 + 2\beta & \dots \\ \alpha L_2 & \alpha L_2 + \beta & \dots & \\ \vdots & & & \\ \alpha L_n & \alpha L_n + \beta & & \end{array}$$

Let us consider the case in which the number of topological shortest paths between the two airports is larger than n and assume that the noise is small enough so that all n paths have topological length ℓ . What is the probability that the flight plan with cost $\alpha L_1 + \beta$ is included into the list of flight plans? In order to happen, it must be

$$\alpha L_n > \alpha L_1 + \beta \rightarrow L_n - L_1 > \frac{\beta}{\alpha} \quad (1.6)$$

By assuming that weights are independent and identically distributed Gaussian variables with standard deviation σ_0 and by using extreme value theory, we show (see Appendix B) that we have a time shifted flight plan if

$$\sigma_0 > \frac{\beta}{\alpha} \frac{1}{g_n \sqrt{\ell}} \quad (1.7)$$

where g_n is a function of the number of flight plans (for example $g_5 = 2.26$, $g_{10} = 2.997$, and $g_{20} = 3.648$). Hence, the actual behavior of the companies depends on their environment, i.e. the precise weights on the possible path. Note that it is more likely to find rerouting strategies for long paths rather than short ones. Hence, in a way all strategies collapse to the same one when the distance is short: the time shifting one.

We now consider the degree of regulation of the system as a function of the number of flights and for different cost function parameters of the unique type of AO present in the system. The simulation results are summarized in Figure 1.10 where we show the fraction of unregulated flights as a function of the total number of flights in the airspace. For every choice of parameters we observe the same

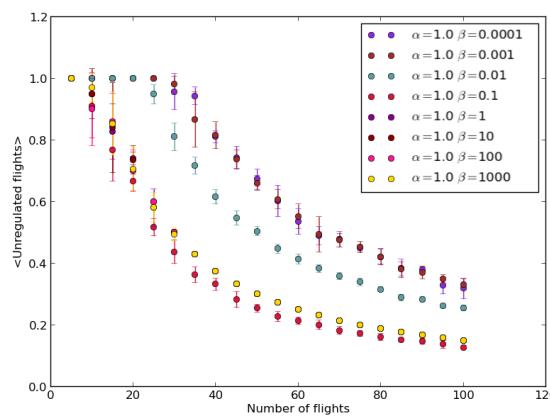


Figure 1.10: Average number of unregulated flights versus number of flights. Different colors correspond to different configurations of the cost function parameters. We set $N_{fp} = 10$.

intuitive behavior: the fraction of regulated flights is zero when the traffic load is small and there is a threshold value when this number declines abruptly toward zero. The value of the threshold is strongly dependent on the cost function parameters. Specifically, when the air companies have a large value of α compared to β , i.e. they care mostly about length or duration of the flight and less about punctuality, the airspace can be filled by more aircraft before starting to regulate them. On the contrary, when AO cares mostly about punctuality and less about the length of the flight, i.e. α is smaller than β , the transition to a regulated phase of the airspace occurs for a relatively small traffic load.

Note that Figure 1.10 and also some figures in the following (like Figure 1.11) are dependent on the parameter N_{fp} , the number of flight plans submitted by one flight. In fact, the threshold is obviously going to be bigger with one flight submits more flight plans, because then it takes a higher total number of flights to reject every flight plan. Similarly, we have the same tendency when increasing the capacity of sectors.

1.6.1.3 Satisfaction metric and its relation with regulation

In order to have a metric to measure the performance of the system under different input conditions (*i.e.* number of AOs, number of flights, number of FP proposals per flight, different configurations of the cost function parameters, etc.) we defined a *satisfaction* measure.

As explained in 1.3.6, each FP that an AO proposes for a given flight is associated with a cost. As shown in the previous sections, different AOs could attribute different values to different terms of the cost function: for example a big company could care much more about arriving on time, than to save fuel, while for a small, “low cost”, one to contain the fuel consumption could be crucial. In our model this is accounted by using different configurations of the cost function parameters for companies that behave in different ways.

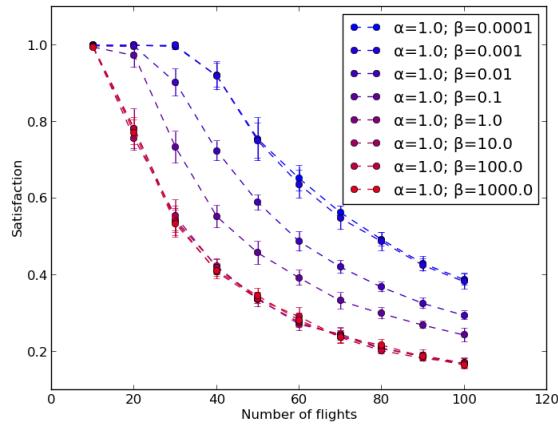
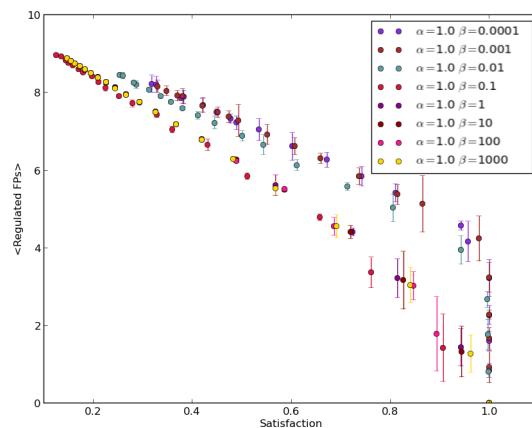
We remind that a proxy for the *satisfaction* of an AO, for a given flight, can be defined as the ratio between the cost of the best FP and the cost of the one which is allocated. When for a flights there are not allocated FPs, the satisfaction is equal to 0, while if the least costly one is allocated the satisfaction is equal to 1. The *satisfaction* of the system is defined as the average of the individual satisfactions of the AOs, performed over all the flights of a simulation.

Here we simulate only one AO type, as before, and we calculate the satisfaction as a function of the number of flights. Different simulations were done by changing the cost function parameters. We show first the relationship that exists between the regulation and the satisfaction, before moving to the analysis of the satisfaction plots.

Figure 1.11 shows the AO satisfaction as a function of the total number of flights for different choices of the parameters α and β characterizing the cost function. This plot is very similar to the previous one of figure 1.10. We find again that for low traffic load the satisfaction is maximal and that there is threshold that is associated to a transition to the regulated phase where satisfaction rapidly decreases to zero.

In order to investigate the relationship between regulation (fig. 1.10) and satisfaction (fig. 1.11), we plot one against the other in figure 1.12. It shows clearly that for each choice of the cost function parameters, there is a negative linear relation between the two quantities, *i.e.* when a large fraction of flight plans is rejected the satisfaction of the company is low. This very sharp relation implies that analyzing the regulated flight plans or the satisfaction gives the same information. In the following we will focus mainly on air companies satisfaction, but, by using the result of Figure 1.12 one can easily translate the conclusion on satisfaction into the corresponding conclusion of regulation of flight plans.

The reason for this relationship is intuitive. For example, consider the case where $\beta/\alpha = 0.0001$. Then if all flight plans are shifted in time, the increase of cost between the first plan and the last one

Figure 1.11: Satisfaction versus number of flights for different values of β/α .Figure 1.12: Average number of regulated FPs versus satisfaction of the system. Different colors correspond to different configurations of the cost function parameters. As in all the simulations, the number of flight plans is $N_{fp} = 10$.

is one order $N_{fp} \times \beta/\alpha \simeq 0.001$, which is very small compared to 1. Hence, the satisfaction value of the last flight plan is very close to 1 (the value of the first one), but then the satisfaction drops to 0 when it is rejected. Hence, it gives almost the same result than by using the rejected flight metric: 1 when at least one is not rejected, 0 after that. All this suggests that in our picture of the air traffic dynamics the satisfaction could be a good metrics to measure the performance of the system.

1.6.1.4 Pseudo-jamming transition

From figure 1.10 and 1.11 one can understand that the satisfaction decreases quickly as the airspace is more and more filled with flights. As the flight “density” increases, the behavior of the system becomes similar to the one found in the dynamics of the *Nagel-Schreckenberg model*¹⁰ for the simulation of freeway traffic; the latter is a simple cellular automaton model, for road traffic flow, that can reproduce what is called a *jamming transition*. In condensed matter physics “jamming” is a process by which some materials, such as granular materials, glasses, foams, and other complex fluids, become rigid with increasing density. The jamming transition has been proposed as a new type of phase transition, with similarities to a glass transition but very different from the formation of crystalline solids. The crowding of the particles constituting the complex fluid prevents them from exploring phase space, making the aggregate material behave as a solid.

In our representation the satisfaction is maximal up to a given number of flights, at which point there is a discontinuity in the slope due to the sudden appearance of “jams” in the air space sectors¹¹; then as the “density” increases further, the satisfaction decreases quickly .

Finally, we see that companies having small ratio β/α are doing better than the other ones, i.e. they have a higher satisfaction. This might be understood by considering that, since this type of companies tend to shift their flight plans in time, they avoid a jammed network finding a relatively more free network one or more time steps ahead. On the other hand, when β/α is high, companies try to find alternative routes. However, since they are geographically constrained at the departure and the arrival, they hit inevitably a full sector.

1.6.2 Mixed populations

After simulating only one type of AO, we extended our simulations by differentiating the flights between two AOs with opposite flight strategies as described in 1.3.4.2. We recall that the first one is a company preferring a short flying route (termed *S*, corresponding to $\beta/\alpha = 10^{-3}$), and describing for example low cost carriers, while the second company attributes a high value to departure punctuality and wants to avoid shifting the time of departure of its flights (termed *R*, corresponding to $\beta/\alpha = 10^3$), describing for example flag carriers or major hub and spoke airlines. We again assume that all the flights connect two airports (in both directions) and we assume that the capacity is the same for all sectors.

It is very important to stress that all the results shown in this section refer to the case when all the aircraft have the same desired departing time. As we will see in the next section, there is a subtle interplay between departing time pattern and satisfaction of air companies. The results presented here correspond almost to the case $\Delta t = 23$ of the next section.

In figure 1.13 we show the global satisfaction GS of the system as a function of the total number of flights. We consider different proportions of flights of the two types S and R of companies. In this case the threshold behavior is barely seen, probably only when the majority of flights is of type S. Clearly GS decreases when the traffic load increase. For a given total number of flights, the global

¹⁰Nagel, K.; Schreckenberg, M. (1992). “A cellular automaton model for freeway traffic”. Journal de Physique I **2** (12): 2221

¹¹Here the term jams is improperly used to indicate the “overloading” of a sector and for this reason we call it a pseudo-jamming transition.

satisfaction is lower when there are more flights of type R. This is expected since we know from Figure 1.11 that, when alone, companies with high β are less satisfied than companies with low β .

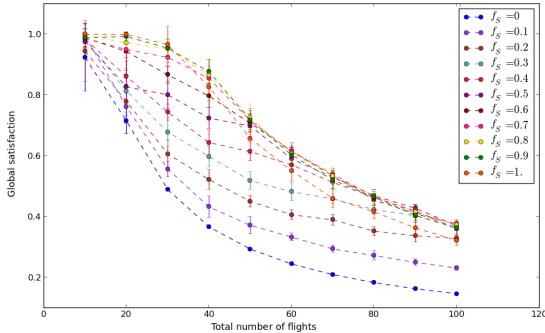


Figure 1.13: Global satisfaction versus total number for different values of the fraction f_S of flights S.

We now consider the satisfaction of the individual companies. Figure 1.14 shows the satisfaction of S as a function of the total traffic load. Each curve refers to a different fraction of flights belonging to company S. One can see that, when the percentage of its flights is low, the satisfaction of S remains stable as the total number of flights grows. After a given threshold value, which is inversely related to the fraction of flights of type S, the satisfaction of S companies starts to decline. Similarly, figure 1.15 shows the satisfaction of R as a function of the total traffic load for different values of the fraction of flights belonging to company R. In this case the thresholding behavior is not observed.

Comparing the two figures, a clear pattern emerges. While obviously for both companies the satisfaction declines with the total traffic load, if we fix the total number of flights, each company (type) will be more satisfied when they contribute for a small fraction of flights. In other words, for a given type of air company it is “better to be alone”, i.e. to be in a minority group gives an higher satisfaction.

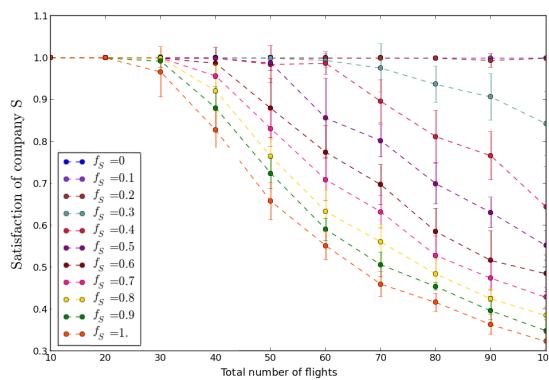


Figure 1.14: Satisfaction of S versus total number of flights for different values of the fraction f_S of flights S.

This effect is even more evident if we plot the satisfaction of the S and R as a function of the

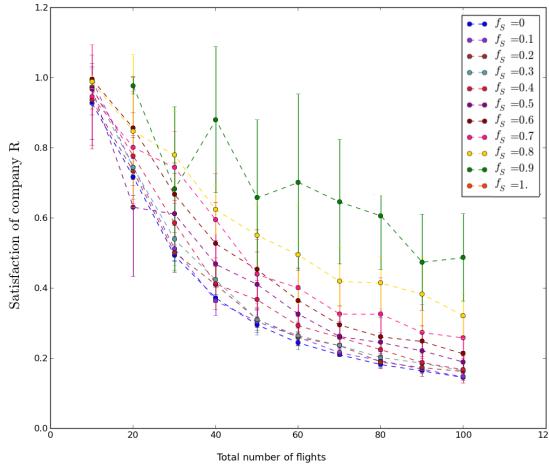


Figure 1.15: Satisfaction of R versus total number of flights for different values of the fraction f_S of flights S.

percentage of flights of R for a fixed total number of flights equal to 100. As figure 1.16 shows, the satisfaction of a company is a monotonically decreasing function of the fraction of flights the company is responsible for. Finally, the two air companies reach a same value of satisfaction for a proportion of R between 10% and 20% (figure 1.16).

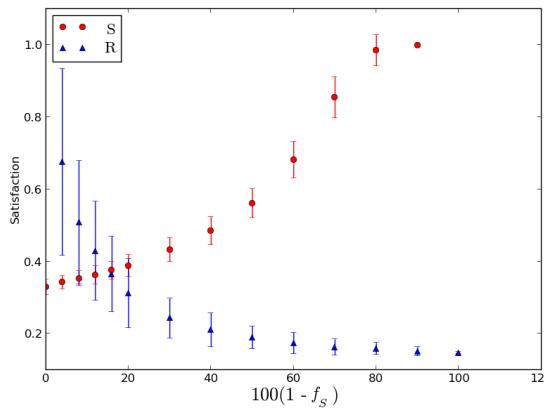


Figure 1.16: Satisfaction of S and R versus the percentage of flights of R, where 100 flights are simulated.

Plotting GS versus the percentage of R flights (figure 1.17) one can see that it monotonically decreases. This is related to the fact that companies with high β (like R) have a satisfaction which is lower than companies with low β (as S). Therefore the global satisfaction depends inversely on the fraction of R flights, as seen in the figure. From this observation one might erroneously conclude that the global optimum is when the airspace is composed only by flights of S.

However this is trivially due to the different levels of satisfaction observed when each company

is alone. We need thus a measure of global satisfaction that does not depend on the different level of satisfaction observed when AOs are alone. In order to obtain a measure of the gain, in Section 1.3.6.3 we introduced the normalized global satisfaction (NGS) of Equation 1.5 where the satisfaction of each company is normalized by the value of satisfaction that one would observe if the company would be alone.

The plot for NGS (figure 1.18) shows that the best way to obtain an high level of normalized global satisfaction is to have an airspace where flights of different AOs are mixed (here in the proportion $f_S = 0.3$). Any other case is sub-optimal.

Notice that the existence of the maximum is not *a priori* obvious, but is related to the dependence of the satisfaction from the fraction of flights. To be specific, consider a toy example in which

$$s_i = \frac{A_i}{f_i^\eta}, \quad (1.8)$$

where $i = R$ or S , s_i and f_i are the satisfaction and fraction of flights of company i , A_i are parameters, and $\eta > 0$ describes the inverse relation between fraction and satisfaction. The NGS is

$$NGS = f_S \frac{A_S f_S^\eta}{A_S} + f_R \frac{A_R f_R^\eta}{A_R} = f_S^{1-\eta} + (1 - f_S)^{1-\eta} \quad (1.9)$$

because $f_R = 1 - f_S$. If $\eta = 1$, this function is a constant, if $\eta > 1$ the function NGS has a minimum in $f_S = 0.5$, while if $\eta < 1$ NGS has a maximum in $f_S = 0.5$. Thus the simulation observation of figure 1.18 is not trivial and depends on the functional dependence of the satisfaction of a company from the its fraction of flights.

In conclusion, when one takes into account the specificity of the cost function parameters of each company, a mixed solution is optimal with respect to a pure case.

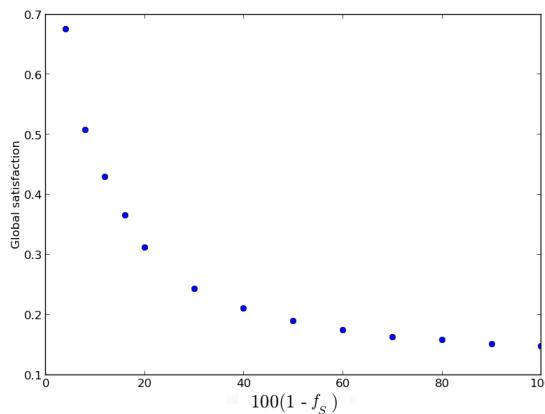


Figure 1.17: Global satisfaction versus percentage of flights R, where 100 flights are simulated.

1.6.3 Departure time pattern

In this section, we investigate how well the airspace is filled, depending on different setups: the departure time pattern and the heterogeneity of companies.

In model 1.0, we can set arbitrary conditions on the departure times. We first explain what is our choice and the associated parameters.

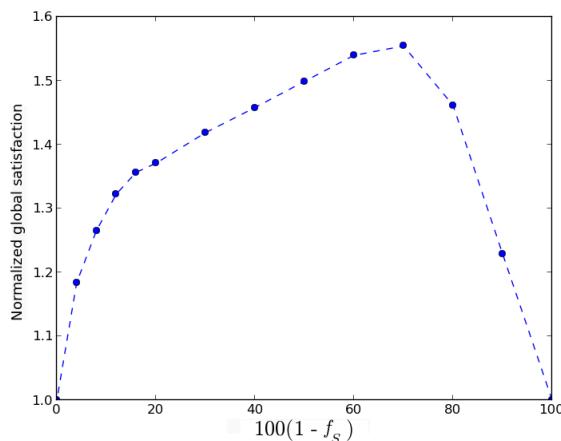


Figure 1.18: Normalized global satisfaction versus the percentage of flights R, where 100 flights are simulated.

1.6.3.1 Modeling departure time pattern

In reality, departure times tend to be gathered in the so-called “waves”. This is due to two reasons:

- First, passengers tend to have a preference for certain hours of the day (like 8-9 am).
- Second, traditional companies have a hub-and-spoke network of airport. They first take all the passengers from their origin, bring them to the hub (Paris for Air France, Rome for Alitalia for instance), and then take them to their final destination. Allowing all the passengers to be gathered at the same time in the same place, and then letting go all the flights at once, allows the company to reduce the connection times for passengers.

We model this in the following way: we define first d , the length of the “day”, i.e. the time window of departure for all flights. In this time window, we define N_p peaks of length 1 (typical crossing time of a sector), by setting a time Δt between the end of the peak and the beginning of the next one (thus, $N_p = d/(\Delta t + 1)$). This is the parameter we are going to use in the following, d being fixed to 24 (notice that this number not necessarily must be interpreted as one day, since the typical crossing time is less than one hour, see also later). Then, we define either a total number of flights and divide them equally between peaks (within which they depart at random), or we set a given number of flights per peak. Figure 1.19 shows the result of such runs, where we set 20 flights per wave, with only companies R (left) or only companies S (right). As we can see, the distribution of real departing times are much wider than the initial distribution of desired flights for this company.

Finally, with this picture, we can have an intuition of what is going to happen for different companies. At the beginning, the first flights are accepted without problem. When the network becomes crowded at the times of the waves, company R will be able to find other paths. However, alternative paths are limited, because anyway the flights have to go through the sectors surrounding the airports, which are quickly crowded. On the other hand, company S can delay the flight plan and find a network less crowded at a later time. When the peaks are well separated, this is what is happening, as it was happening in the previous section 1.6.1. But when Δt becomes small enough, the waves are close to each other. Hence, the delayed flights of company S will “hit” the next wave and get rejected. What was an advantage turns into a disadvantage: delayed flights are not accepted anymore, whereas rerouted flights can find an accepted flight plan.

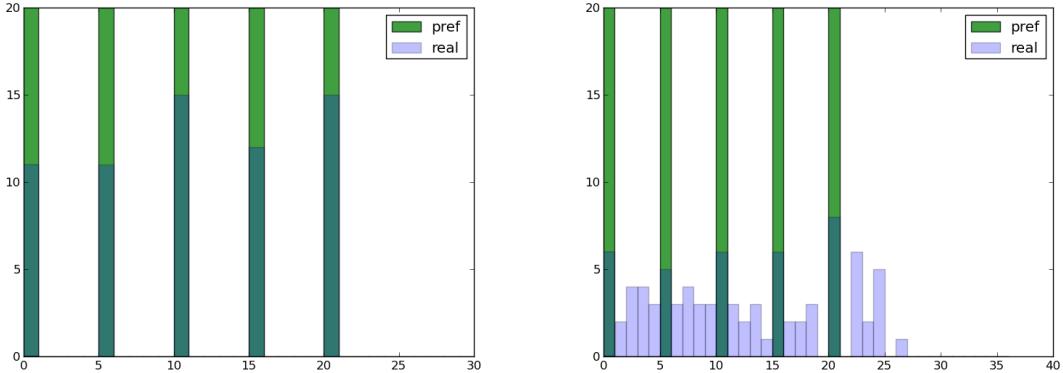


Figure 1.19: Distribution of departing times. In green, the distribution of desired times. In blue, superimposed, the distribution of real departure. Left: run with companies only of type R. Right: run with companies only of type S.

1.6.3.2 The interplay between departure time pattern and fittest company, pure populations

Now that we have sketched out what is likely to happen, we show some simulations displaying the interplay between the ratio β/α and Δt , i.e. the behavior of companies and the departing time pattern.

Figure 1.20 shows the evolution of the satisfaction when we change β/α and Δt . Starting with $\Delta t = 23$ (one peak), we see that the companies begin to lose their advantage when β/α is increasing, i.e. when they shift less flights. However, as Δt decreases, companies of type R regain their advantage and eventually get a higher satisfaction.

Added to this, and quite surprisingly this relation is not always monotonic. For instance for $\Delta t = 1$, it is always better to have $\beta/\alpha = 0.1$ than the values at the extremes. It means that mixed strategies – a bit of rerouting, a bit of shifting – are doing better for some values of the parameters. In a way, this is also intuitive, because mixed strategies are able to get the best of the extremes when these ones do not have a clear advantage one over the other.

Note that the ranking of the companies for different values types of departing pattern depends on the particular pairs of airports one considers and the typical overlap between the sector paths connecting the two airports. We study more in details this point in section 1.6.5.

1.6.3.3 The interplay between departure time pattern and fittest company, mixed populations

After simulating only one type of AO, we extended our simulations by differentiating the flights between two AOs with opposite flight strategies, companies R and companies S (see 1.3.4.2 for details).

In figure 1.21 we show the global satisfaction GS of the system as a function of the total number of flights for $\Delta t = 23$ (one peak, left panel) and for $\Delta t = 1$ (12 peaks, right panel). We consider different proportions of flights of the two types S and R of companies. Clearly GS decreases when the traffic load increases. In the case where $\Delta t = 23$, for a given total number of flights, the global satisfaction decreases with the fraction of companies R. This is expected since we know from Figure 1.11 that, when alone, companies with high β/α are less satisfied than the others.

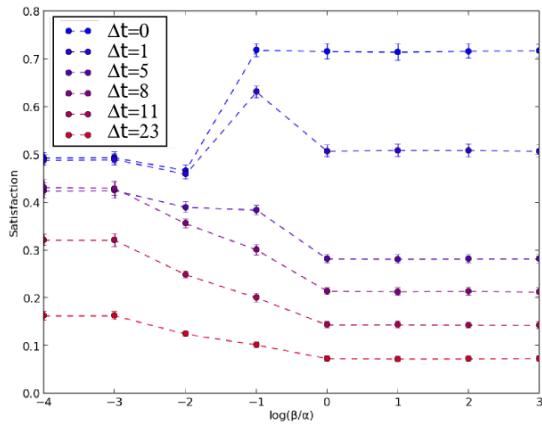


Figure 1.20: Satisfaction as a function of the ratio α/β of the cost function for different values of Δt . The simulation is done with a number of flights per peak kept fixed, which the overall satisfaction is decreasing when Δt decreases: there are more and more flights in total, because there are more and more waves.

On the other hand, the right panel of figure 1.21 shows that the picture is more complex when Δt is decreased. We clearly see that the different level of mixing of companies do not result in the same behavior with respect to the number of flights. In particular, for a given number of flights, the satisfaction does not seem monotonic anymore with the fraction of f_S of flights S.

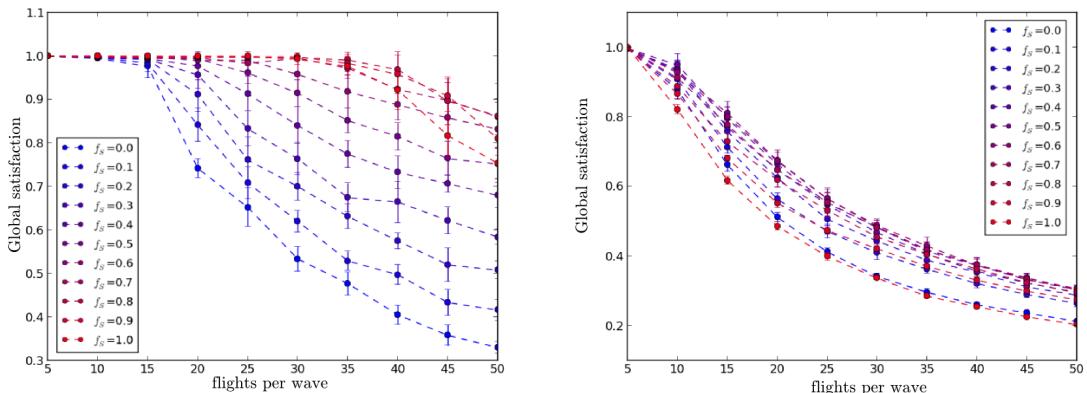


Figure 1.21: Global satisfaction versus total number of flights when the two types of companies S and R are mixed for different fraction of companies S. Left: case where $\Delta t = 23$. Right: case where $\Delta t = 1$.

In order to see this effect, we plot in figure 1.22 the satisfaction against f_S for a given number of flights (120). First, we see that the extreme values have the expected relative behavior: for big Δt , a pure S population ($f_S = 1$) is more satisfied than a pure R population ($f_S = 0$). But in between, the behavior is not trivial and in general there exists a maximum of satisfaction for an intermediate level of mixing. Hence, it is better for the system as a whole to have different types of companies in

the same airspace.

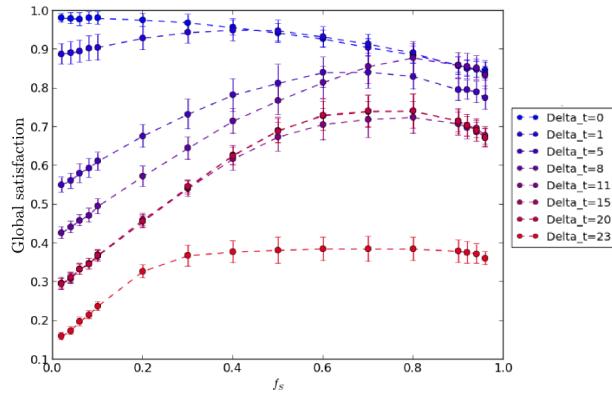


Figure 1.22: Global satisfaction against the fraction of company S for different values of Δt .
The number of flights is set to 120.

This results calls for a close inspection of the individual satisfactions, i.e. for each type of companies. Figure 1.23 shows their evolution against the level of mixing. On the right panel, we see that the satisfaction of company R decreases monotonically with $1 - f_S$, the fraction of R companies. This means that R companies tend to have a big advantage when they are surrounded by the other type of companies, thus displaying a “the loner, the better” effect.

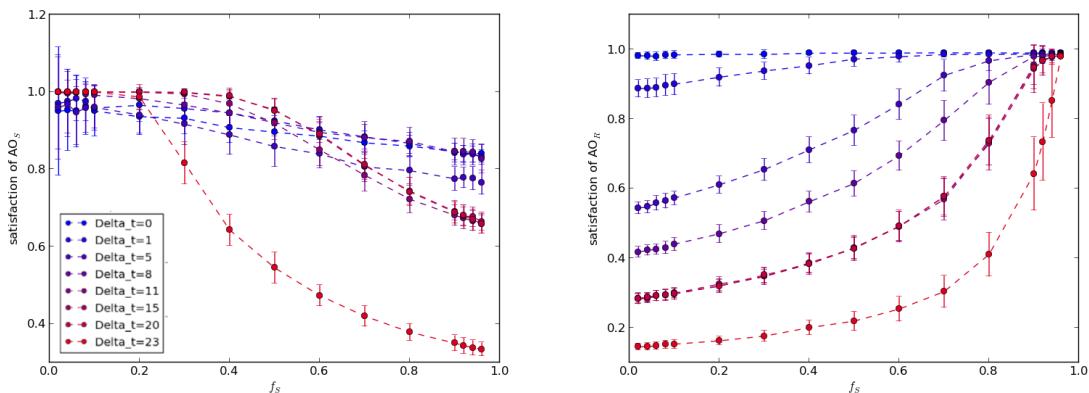


Figure 1.23: Satisfaction of company S (left) and R (right) against the fraction f_S of company S for different values of Δt .

This is also true for company S, as the left panel shows. However, the variations are more complex here. As one can see, the different curves corresponding to different values of Δt are crossing each other. This means that for different values of f_S , it is better sometimes to have a small Δt , and sometimes a big Δt . For instance, for $f_S = 0.4$, it is better to have $\Delta t = 20$ rather than to have $\Delta t = 1$, which is perfectly normal for S companies. On the other hand, for $f_S = 0.8$, the contrary happens, and suddenly it is better for S companies to have a more uniform distribution of departing times. The whole picture is even more complicated by the fact that this behavior is not monotonic

with Δt . With the same example, with $f_S = 0.4$, it is better for S to have $\Delta t = 20$ rather than 1, but it is much worse to have $\Delta t = 23$. This non trivial effect of mixing different companies gives a rich behavior in terms of optimization of the total satisfaction, as noted above.

1.6.4 Shocks: from M0 to M1

In this section, we briefly show the results of the presence of shocks in the system, after the flights have already been assigned to the airspace. This is a model for the unexpected events happening only a few hours before the flight departure like bad weather, strikes, or military requirement. The way we are modeling these shocks is the following: after the first round of allocation, we shut down a sector randomly, for all times. The NM detects which flights are affected by the shut down, and asks the companies to resubmit a bunch of flights. All these companies recompute N_{fp} new flight plans on the modified network, and the NM tries to allocate them again. Then, we shut down a second sector, and so on, until we reach the desired number of shutdown sectors.

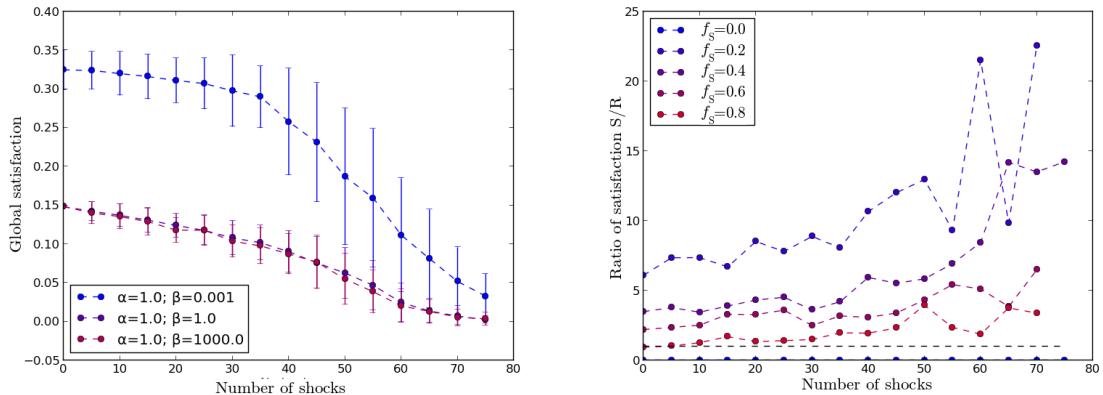


Figure 1.24: Satisfaction of the system against the number of shocks for different the ratio β/α in the pure population case (left panel) or different values of the fraction of company S in the mixed populations case (right panel).

Figure 1.24 shows the results of the procedure in two cases: pure population (left panel) and mixed populations (right panel). All flights here have the same desired time of departure. We see that the satisfaction decreases monotonically with the number of shocks, with a small kink for the S company. This kink happens at a value slightly smaller than the percolation threshold (which is 47 ± 8) for this network. As we already know, S companies have the advantage over the other types of companies when the number of shocks is small. However, the relative levels of satisfaction are changing when the number of shocks increases. In fact, it is possible to show that the ratio of the satisfaction S over the satisfaction R is increasing. This means that company S tends to increase its advantage over company R. This can be understood the following way: when there are few paths remaining on the network, company R is initially the best one. However, the other paths are much longer, thus much more costly, due to the “holes” in the network. On the contrary, company S finds the best path and stick to it, only shifting in time the other flight plans, which is not more costly than usual.

This is also what we see in the mixed populations case (right panel of fig. 1.24). The advantage of company S tends to increase with the number of shocks, whatever the level of mixing. This behavior is much stronger when f_S is small.

1.6.5 Generalization of the results

In this section we consider two further generalizations of the previous results. First, we will show that the ranking of companies in terms of satisfaction not only depends on the departure time pattern, as we have seen above, but also on the geometry of the local network of sectors between the two airports. In the second generalization we will consider a full simulation of the model with multiple airports.

1.6.5.1 Ranking of companies and topology of the sector network

All the above results have been obtained with the same pair of airports on the same network, only changing the parameters of the simulations themselves. In this section, we briefly investigate what happens in the general case. First, it is clear that the results of section 1.6.1 still holds for other (Delaunay) networks/airports. Thus we focus mostly on the main mechanism of change of advantage between companies S and R, i.e. the interplay between β/α and Δt .

In order to do this, we have extracted all the pairs of airports of our network which are at the same topological distance (6 edges) one from each other. Then we perform simulations on them independently.

One of the main points of section 1.6.3 is that a well spaced wave structure gives an advantage to company S over company R, whereas a more uniform departing time pattern gives an advantage to R. This advantage can be quantified by the difference

$$\delta s(\Delta t) = s_S(\Delta t) - s_R(\Delta t) \quad (1.10)$$

where $s_S(\Delta t)$ ($s_R(\Delta t)$) is the satisfaction of company S (R) when the departing time pattern has waves separated by Δt . Performing simulations on the thousand pairs independently, we first checked the sign of $\delta s(\Delta t)$ in the case where $\Delta t \rightarrow \infty$, i.e. where there is only one peak. We found that only very few cases were returning a negative $\delta s(\infty)$. Hence, company S has always the advantage when there is only one peak. Given that this value has always the same sign, we focus on the other extreme, $\delta s(0)$, corresponding to a uniform departing time pattern, to see if the company R has always a higher satisfaction in this case.

Figure 1.25 shows the histogram of the quantity $\delta s(0)$. As we can see, there is a significant fraction of cases (but smaller than 50%) where $\delta s(0) > 0$ (right side of the plot). This means that in these cases, company S is still getting a better satisfaction than company R. What are these cases? Intuitively, the R company does not get any advantage when the rerouted flights keeps hitting at least one overloaded sector. This means that, no matter its rerouted path, if the flight is crossing a sector which is already in a previous flight plan, it will be rejected by the NM. Hence, an important measure here is the overlap between the paths of the submitted flights. Our hypothesis is that if the typical overlap between the paths connecting two airports is large, there is no advantage to reroute the flight, even when an uniform departing time pattern would give an advantage to company R with respect to S.

To test this hypothesis, given the set P of the N_{fp} shortest paths between a couple of airports – which are entirely used by company R – we define the overlap $O(P)$ as following:

$$O(P) = \frac{1}{N_{fp}(N_{fp} - 1)} \sum_{p_1 \in P} \sum_{p_2 \in P} \frac{|N(p_1) \cap N(p_2)|}{|N(p_1) \cup N(p_2)|},$$

where $N(p)$ is the set of sectors of path p . This is the ratio of the number of sectors common to two paths over the total number of (unique) sectors in these paths, averaged for every pairs of paths

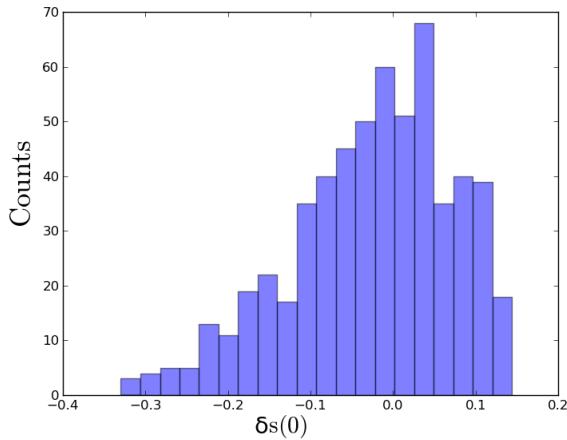


Figure 1.25: Histogram of the quantity $\delta s(0)$ (see Eq. 1.10) giving the difference in satisfaction between company S and R when the departing time pattern is uniform ($\Delta t = 0$).

among the N_{fp} . Note that we have included both airports in the metric, so it cannot be equal to 0. It cannot reach 1 either, since it would mean that every path is the same¹².

In figure 1.26 we show the relationship between this overlap, computed for each pair of airport, and $\delta s(0)$. It displays a clear positive correlation between the two quantities. In particular it shows that the bigger is the overlap between the paths connecting the two airports, the bigger the advantage of S over R.

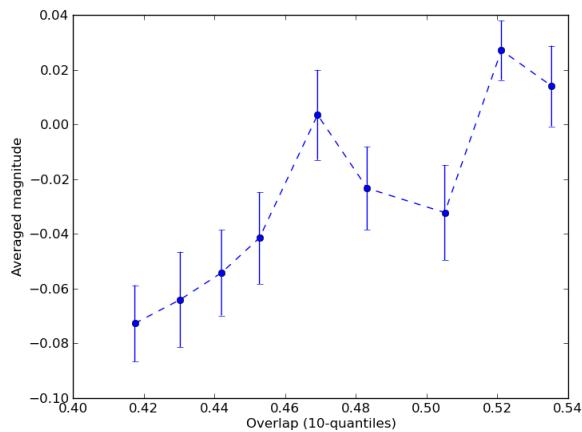


Figure 1.26: Relationship between overlap and $\delta s(0)$ (see Eq. 1.10), giving the difference in satisfaction between company S and R when the departing time pattern is uniform (i.e. $\Delta t = 0$). We show the average on each quantile of overlap. Error bars are standard deviations.

This sheds light on our previous results. The mechanisms we have found are present in the

¹²More precisely, that they are composed of the same sectors, which could in principle be in different orders. This is very unlikely to happen for $N_{fp} = 10$.

system, but are potentially hidden by some special configuration of the network for certain airports. For the two airports considered in the previous sections, the overlap is 0.48, giving a negative value of $\delta s(0)$, consistently with the fact that uniform departing time gives an advantage to R companies over S.

In conclusion, the precise interplay between the environment (the network), the common trends (the waves), and the individual behaviors (the parameters of the cost function) gives rise to different advantage for each agent, and thus to different ranking orders of different companies.

1.6.5.2 Simulations with more airports

Finally, we describe some simulations which get closer to reality, even though they are still run on the generated network. More specifically, we now take six airports, which results in 13 pairs of airports open to flights. We also setup the waves in a more realistic way. Since the typical crossing time in the model is the unit of time and that it represents roughly 10 minutes, we considered three waves of departures of 6 units, hence of one hour approximately. The end of a wave is separated by 24 units from the beginning of the next, representing 4 hours. the total length of the “day” is 90 units of time.

We look only at the mixed population case to see the differences with the previous setup. Figure 1.27 shows the global satisfaction as well as the satisfaction of each type of company. Quite surprisingly, the company R is quite insensitive to the level of mixing, whereas company S is still sensitive to it. Hence the overall satisfaction depends only on the pattern of the latter. In fact, we see that the satisfaction shows a clear plateau for high values of the number of flights. This means that in this region, the variation of the satisfaction of company S exactly cancels the effect due to f_S . Thus, the satisfaction of S has to decrease as $1/f_S$.

1.6.6 Calibration on real data

1.6.6.1 Input from WP1 and new empirical facts

In order to calibrate the ABM based on real data, we needed new empirical facts, which were not considered in deliverable D1.3.

Time of crossing

We extracted the duration of flights within a sector to calibrate the weight of the edges (see section 1.3.2) with real data. The procedure is the following: using the database (see deliverable D1.1), for each flight we collect the trajectory in terms of navigation points and in terms of sectors. Knowing also to which sector each navigation point belongs, we extract the time spent within each sector by the flight. Then, we set the weight of an edge equal to the sum of half the crossing time of sectors of the two nodes. Hence, we consider that, on average, a flight spends half of its time in one sector and the other half in the other sector (which is actually exactly the case for Voronoi/Delaunay networks).

Figure 1.28 gives the histogram of these weights, extracted from the database for the whole ECAC space. Note that the ratio between the standard deviation and the average is 0.61, which is larger than the value we used in the simulations. We have performed an analysis of the theoretical distribution that best fits the empirical data. We have considered several different functional forms and we have found that data are best fit by a log-normal distribution with parameters $\mu = 2.542 \pm 0.008$ and $\sigma = 0.582 \pm 0.006$. The right panel of Figure 1.28 shows the QQ plot of the empirical data versus the theoretical best fit lognormal distribution. The fact that the points lie on a straight line, also for small and large values of the crossing time, indicates that a remarkable good fit.

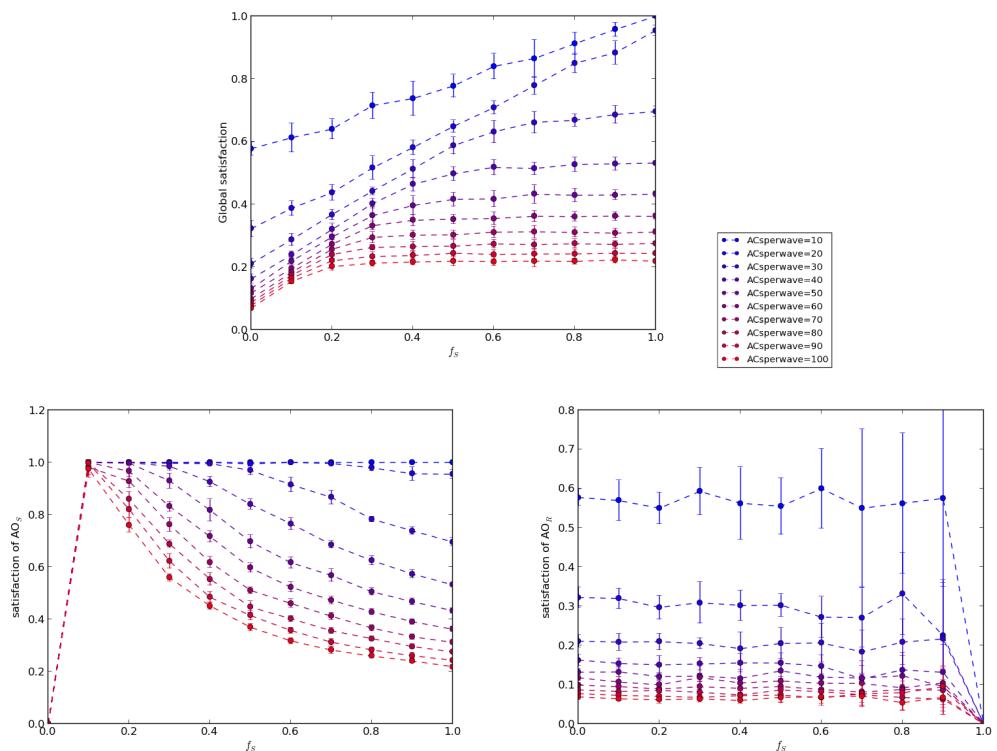


Figure 1.27: Global satisfaction (up), and satisfaction of each company (bottom) as a function of f_S for different values of the number of companies per waves.

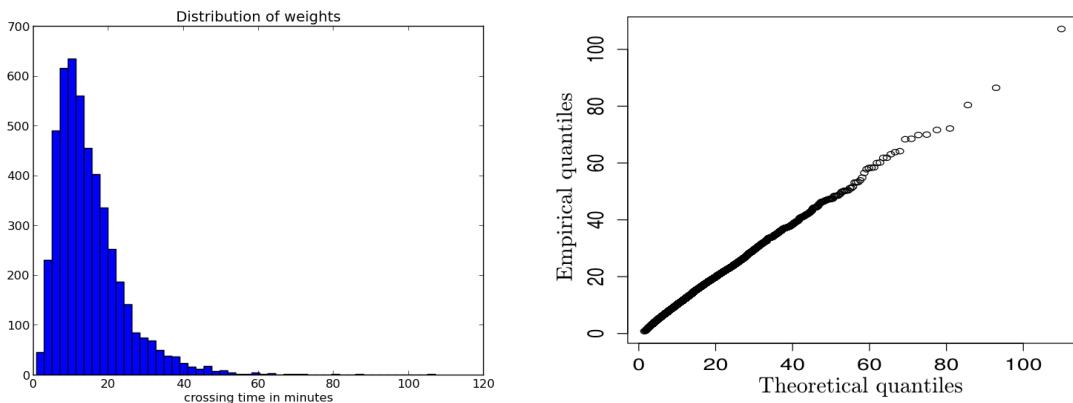


Figure 1.28: Left. Histogram of time of crossing between sectors (average time to reach the center of a sector from another one). Mean/std: 15.0/9.15. Results obtained from the whole ECAC space for one day (6th of May 2010). Right. QQ plot of the sample data versus the best fit with a lognormal distribution.

Capacities, area and volume

Another quantity of importance for the model is the capacity of sectors, which, in our modelization, is a constant number. The real capacity of sectors is out of our reach since we have no data about it. Instead, as mentioned in the description of the model, for the model 1.0 we used a measure related to real capacity, namely the maximum number of aircraft present at the same time in a given sector in the M1 file (planned trajectory). As we will show in the next Section on model 2.0, where we infer the sector capacity by measuring the maximum number of flights present in a sector in one hour, the two proxies are strongly related. In fact their correlation coefficient is 0.8 and the true capacity (as estimated from traffic data) is on average 2.8 times the proxy used here. Therefore, apart from a scaling factor, we expect that the results shown in this calibration exercise are a pretty faithful representation.

Figure 1.29 shows the histogram of capacities for one entire day. The capacities are centered around 6, except for very few, large outliers which correspond to big sectors in Germany.

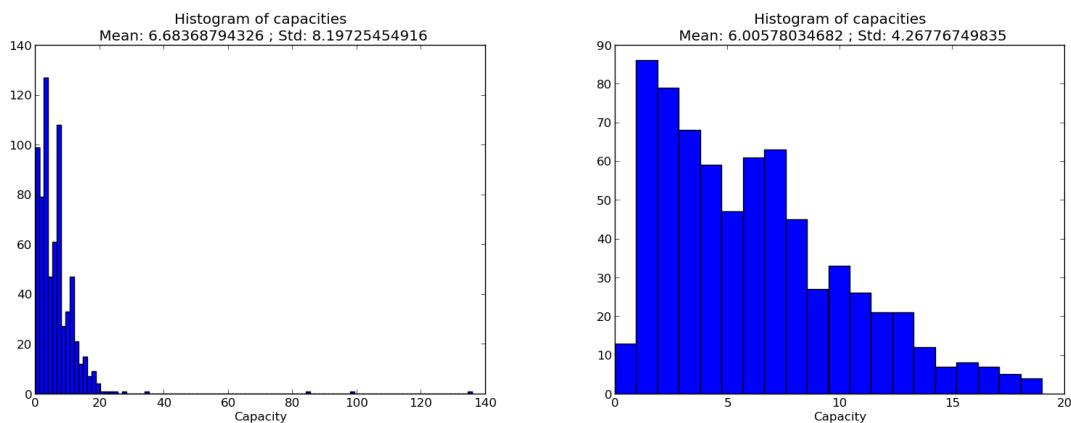


Figure 1.29: Histogram of a proxy for the capacity of sectors i.e. the maximum number of flights in the sector at a given time. Left: entire histogram. Right: zoom on 20 first values.

Moreover, we are interested in how the capacities are related to other properties of the sectors. Specifically, we test if the capacity is related to the size of the sector. To this end, we have extracted from the database information about the spatial extension of the sectors, namely their area and their approximate volume. In figure 1.30, we present the relationship between these quantities. It is interesting to notice that the right panel indicates that the capacity used here scales quite nicely as the square root of the sector's area (with the only exception of few very large sectors). This fact is actually related to how controllers perceive the complexity of traffic as a function of the number of flights. We can explain it by the following reasoning.

Given a sector of area A with f flights traveling through it, the number of potential collisions have an upper bound, which is the number of pairs of flights: $f(f - 1)/2$. Hence, if the capacity of the sector implies that C flights maximum are allowed to fly through this sector, the maximum number of potential safety issues is $C(C - 1)/2 \sim C^2$. Now note that these conflicts are harder to manage if flights are close one to each other, and easier in the other case. If fact, the density of flights plays a crucial role here. Hence, we assume that the controller can handle a given maximum number of conflicts for a given area. The complexity of a sector is thus proportional to C^2/A . If we consider that sectors are created so that the controllers are close to their limits, the complexity should be constant throughout the network. Hence, $C \sim \sqrt{A}$, which is the relationship we observe

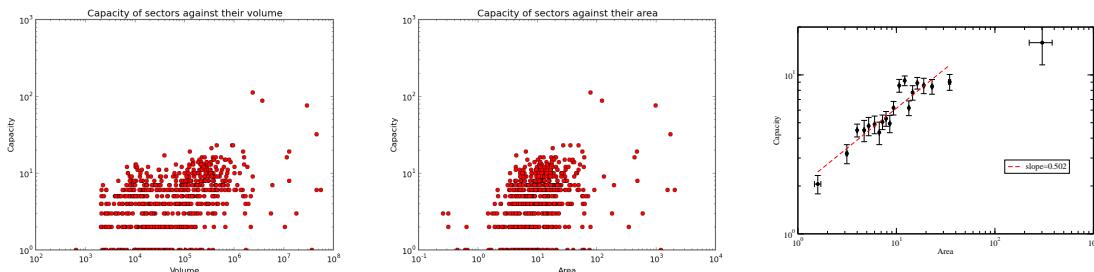


Figure 1.30: Left panel: relationship between capacity and volume of the sectors. Center panel: relationship between capacity and area. Right panel: relationship between capacity and area averaged on quantiles.

empirically.

The real network

In order to test the model on a real system, we selected a piece of the real European network of sectors. We chose France for its high traffic and its regular shape.

First, we extract the network of sectors from the data, in the same way we have done in the Deliverable D1.3, selecting only sectors in the French airspace. The links between the nodes are then given by the traffic: two nodes (sectors) are linked if at least one flight went from one to the other in the given time interval. A major issue is to properly extract the right nodes. Indeed, since the nodes can appear or disappear (split/union of sectors), we needed to find a time interval in which we observe a static network. Fortunately, one can find such a time interval, of reasonable length, for France: from 3 pm to 5 pm on the 6th May 2010, all the sectors remain static. Hence, we have built the network based only on the flights travelling during this time period. Using a weak filter (which includes almost all the flights, see Deliverable D1.3 for more details), we found 1316 flights, creating a network of 162 nodes and 528 links.

We decided to reduce this network for several reasons. The first one is that some flights cross only one or two sectors and then exit the airspace. This gives a lot of isolated very small sets of nodes (1 or 2 nodes) not connected to the rest of the network. Hence, we considered only the biggest connected component of this network. The second issue is the following: some edges are only used by one flight in the time interval. These edges, although relevant generally speaking, are clearly of smaller interest for us than the “backbone” of the network, gathering up to 40 flights. Moreover, once we have created the edge, it will have the same importance during the ABM process than other links, which we know are of major interest. Thus, we decided to discard all edges having only one flight (and the nodes not connected anymore).

We end up with a network of 106 nodes and 316 edges, comparable in size with the ones generated in the previous sections. The mean degree is 6.0, exactly like the Delaunay and triangular networks, but is not planar. It is displayed in figure 1.31.

We chose two airports for the test. Since airports in France are of course close one to each other, we decided to take instead two sectors as “airports”. This does not change anything to the procedure in the ABM, the only potential drawback being that they have an infinite capacity. It is not a problem here, since we have only a pair of airports, and thus no other flight is passing through them. The two specific sectors are shown in the next section 1.6.6.2.

The crossing times between sectors are computed by using their geometrical (2D) distance between their centroids. We will use the real time-crossing in the future, even though we do not expect a real

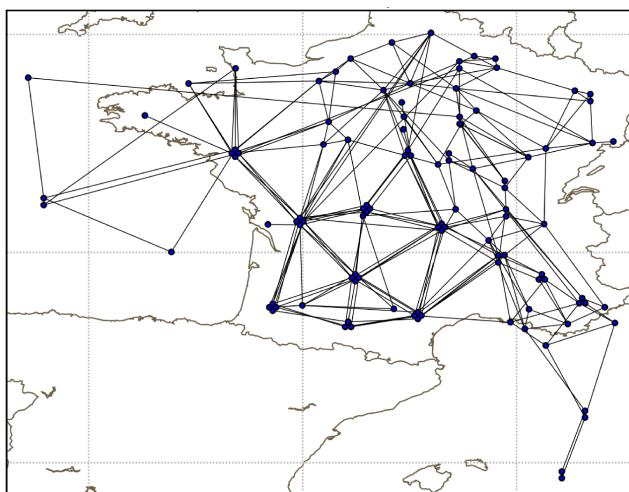


Figure 1.31: Network of sectors of France used to run the ABM. Each circle represents a sectors and is situated at the geometrical centroid of it. The sectors on top of each other are shifted a bit to better see them.

difference.

Finally, we can choose the capacity based on several distributions – constant, uniform, gaussian – but we can also take the real capacities extracted from the data, as explained previously.

1.6.6.2 Results calibrated on the airspace of France

Maps

In this section we present the results obtained by running the ABM calibrated on the real network. First, we chose a constant capacity (5, which is the average capacity extracted from the data) and only one type of company. We used also the real capacities of sectors, as described previously. However, the results are very similar, so we only show the first ones (constant capacity) here.

Maps of figures 1.32, 1.33, and 1.34 give an image on the resulting trajectories and corresponding occupation of the airspace for different values of the cost function parameters.

The ABM runs smoothly on these new networks, even though there is still a small issue with the altitude: since there is nothing related to altitude in the cost function, the flight can freely pass from a low sector to a high sector when there are one over the other.

Except for this point, we recover the main behaviors emulated by the cost function. When $\beta/\alpha \ll 1$, the companies do not care much about the time of departure compared to the desired time. Hence, they shift all their flight plans in time and take the same spatial path. Sectors are then very close to their capacity. When β starts to increase, the spatial dimension is relaxed, because it is now less costly to change the path compared to the shift in time. Hence, some alternative paths are used, and some sectors are well below their capacity. Finally, when $\beta/\alpha \gg 1$, the companies take all the same time of departure and only change the path. Hence, many more sectors are used.

Regulation and satisfaction

As for the generated network, one can also compute and plot the regulated values (flight plans and flights) as well as the satisfaction, to see how they are different. Figure 1.35 (obtained with constant capacities) shows that these metrics are similar to the ones obtained with a generated network.

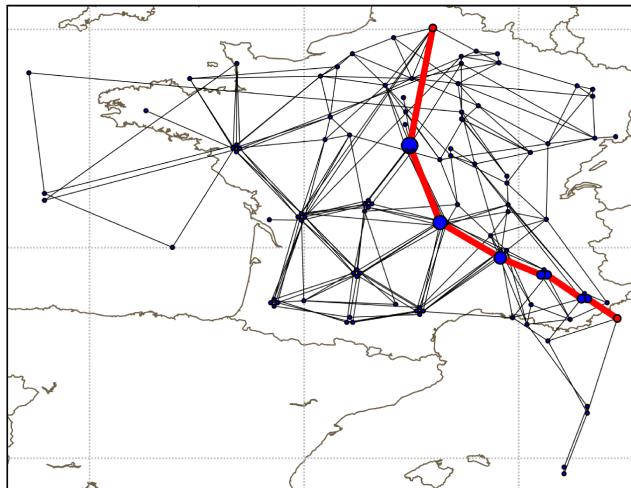


Figure 1.32: Trajectories and occupation of airspace resulting from a run of the ABM. The nodes in red are the “airports”, i.e. the departing and arrival sectors. All the other nodes have size corresponding to their averaged occupation. The red lines shows the trajectories actually flown by the flights; their thickness is proportional to the number of flights using them. This map has been obtained with constant capacity (equal to 5) and one type of company with $\beta/\alpha = 0.0001$.

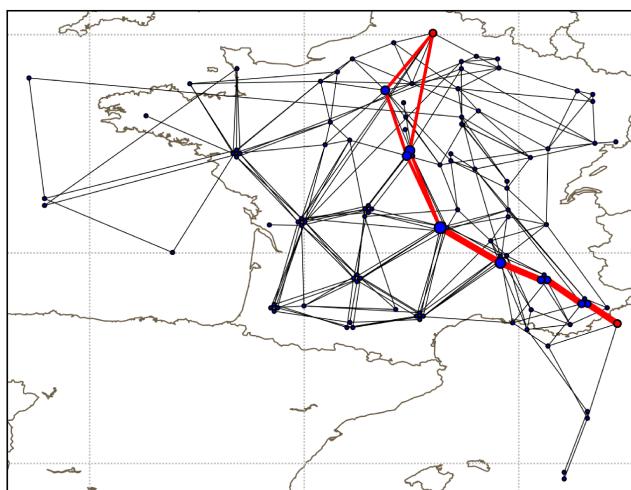


Figure 1.33: Trajectories and occupation of airspace resulting from a run of the ABM. See figure 1.32 for a complete description. The companies have all $\beta/\alpha = 1$.

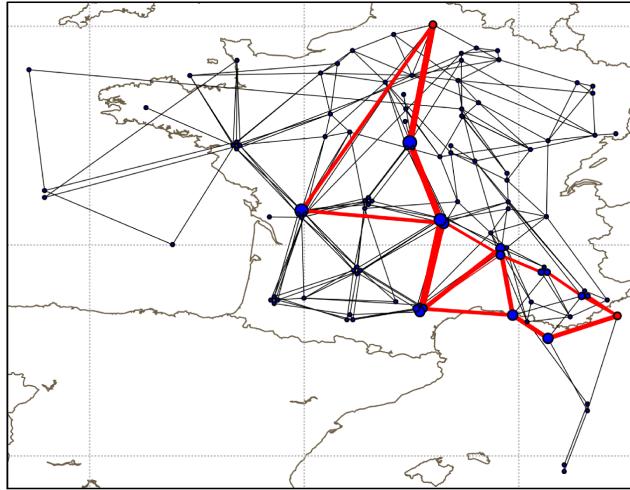


Figure 1.34: Trajectories and occupation of airspace resulting from a run of the ABM. See figure 1.32 for a complete description. The companies have all $\beta/\alpha = 10000$.

However, we see that the threshold, clearly visible on figure 1.11 for instance, is almost absent in the present case. It is not clear why at this stage, but it could be of great importance because raising this threshold would increase the global capacity of the network. On the other hand this observation is based on a small number of simulations and therefore deserves further investigations.

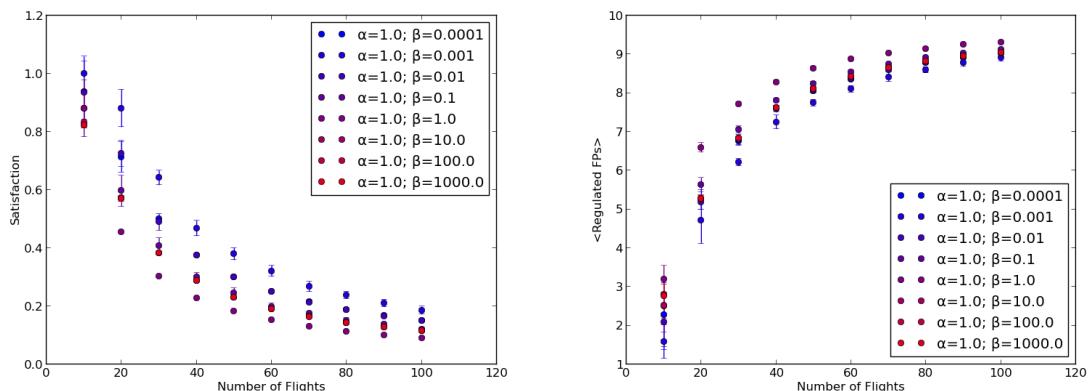


Figure 1.35: Left: satisfaction against number of flights for different parameters. Right: number of regulated flight plans against number of flights for different parameters. Results obtained with the real network, with constant capacities.

We also show on figure 1.36 the same plots obtained with the real capacities of every sectors. As one can see, the plots are really close to the ones with constant capacities. The satisfaction is globally smaller than before. This could be due to the fact that one or few sectors of the best path(s) have a small capacity. Hence, they act as bottlenecks and the sectors of higher capacities are far from being full.

Finally, we show in figure 1.37 the dependence of the satisfaction from the progressive change

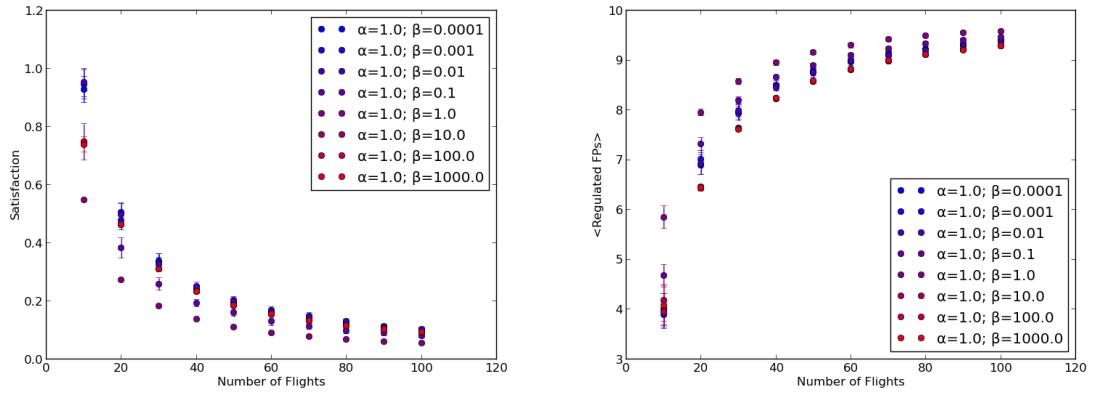


Figure 1.36: Left: satisfaction against number of flights for different parameters. Right: number of regulated flight plans against number of flights for different parameters. Results obtained with the real network, with real capacities.

of the cost function for a given number of AOs (50). As one can see, the evolution is clearly non-monotonic, with a drop around $\alpha = \beta$. Although this result cannot be considered general yet, since we use only a specific network with a specific pair of airports, it is striking to recover a non-monotonic behavior like with the generated network (see figure 1.20).

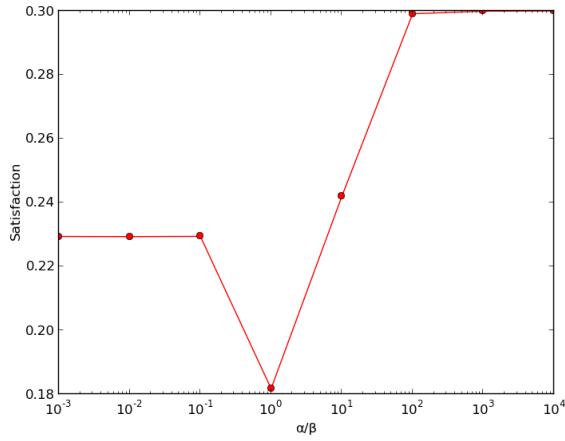


Figure 1.37: Satisfaction against the ratio α/β of the cost function for a given number of Air Operator (50).

1.6.7 Conclusions

This toy model 1.0 sheds some light on the main mechanisms of the ABM. The very first one is the existence a phenomenon of jamming, embodied by the dramatic increase of rejected flights after the number of flights reaches a certain threshold. This threshold depends on “static” parameters linked to the network itself, like capacities, but also on the behaviors of the air companies undergoing.

Indeed, not only the behavior of the individual companies matters, but also the collective one. Hence, if companies tend to choose the same time of departure, the strong structure of waves gives the advantage to company S. Moreover, companies tend to be more satisfied if they are surrounding by other types of company. Moreover, the non trivial variations of the satisfactions of each type of companies lead to the existence of plateaus and maximums for the global satisfaction. Hence, in general, the entire system is more satisfied with a diversity of behaviors than with a “monopoly”.

The shocks we implemented are a way of investigating how the structure of the network impacts the companies. The conclusion is that, apart from the obvious decrease on satisfaction for all types of companies when the number of sectors shutdown increases, company S tends to be more resilient than company R. Indeed, its satisfaction decreases less quickly than the R company's one, resulting in an increased advantage.

These results still hold in the general case, although they can be hidden by the network itself and the overlap between shortest paths. Our simulations with more airports show also the same kind of behavior, although the global satisfaction does not exhibit a maximum anymore (but still displays a plateau).

Finally, when setup on a real network, the ABM shows the same kind of mechanisms. Moreover, we see that the precise distribution of capacities does not have a big impact on the results. Model 1.0 might be too simple to grasp the full complexity of the real case though, which is why we will use the model 2.0 in the next section in order to investigate this case.

1.7 Results with model 2.0

We describe here the results of the model 2.0 of the strategic phase. We present here only the results obtained on the real airspace of France and we do not present the results on the artificially generated networks and airspaces. Moreover our simulations are obtained by calibrating the model on the real traffic present in a specific day, namely May 6th, 2010. More specifically, after having generated the real navpoint network with the data from the real flights, we first fix the number of flights in the simulations equal to those of the real one. Then we extract the pairs of navpoints (entry, exit) of the real flights and set them as airports for the flights¹³. The times of entries of every flight are also extracted from the database and we take them as desired times of departure for the air companies. Thus the wave structure of departing time is inherited from real data. We then apply the model of the interaction between air companies and the network manager, as described before, by considering both cases of pure and of mixed populations of companies.

This approach to modeling allows us to reduce significantly the number of parameters of the model. They are:

- The number N_{fp} of flight plans submitted to the NM by the AO. This parameter is set to $N_{fp} = 10$.
- The time shift τ used by companies to delay the departing time in case of rejection by the NM. We set this parameter to $\tau = 20$ minutes.
- The number N_{sp} of navpoint paths per sector path used by the AO in the selection of the best trajectories. We use this parameter for the calibration of the model on real data, but we also explore how results change when it is varied.
- The ratio(s) β/α characterizing the strategy of the company.
- The fraction f_S of companies of type S in the mixed population case.
- The standard deviation δt_0 of the noise added to the departing time.

As it should be clear, the first two parameters are fixed, and the last three are kept as variable to study how the output of the ABM changes when we vary them. Finally, N_{sp} is used to calibrate the ABM on real data.

1.7.1 Calibration of the model

1.7.1.1 Extraction of the data

In this deliverable we focus on the case of France. The choice of this country is mainly dictated by two reasons. First, it has an airspace which is almost convex, which helps in the treatment of data, since the number of flights entering (or exiting) more than once in the airspace is limited. Second, it is big enough to include several airports, many entry and exit points, and several zones of controls.

We chose the set of flights based on the “Strong” filter (see Deliverable D1.3), which in first approximation gives the commercial flights. For the date we have chosen, the 6th of May 2010, there are approximately 3800 such flights spending at least part of their flight time inside the airspace of France. We then restrict our attention to those flights that do not enter or exit the France airspace more than once. This leads us to 3584 flights.

The way we have built the network is the following: we selected a flight level, namely 350, and selected all the sectors present at this altitude. Then for each flight, whose trajectories are cut below

¹³For a setup in a real environment, the ABM takes “airport” in a broad sense.

flight level 240, we recorded the sequence of waypoints they were following. Finally, based on its coordinates, we associate each waypoint to a sector. Note that it is not necessarily the sector that the flight was passing through in reality, because we used the sectors present at FL 350, not at the real altitude of the flight. In this way, the “sectors” of our network can be understood as vertical stacks of real sectors. In fact, several of the sectors at this altitude have the same 2D boundaries, and are just slices of a bigger volume delimited by these boundaries.

The waypoints extracted in this way are also linked one with each other based on the real traffic. Specifically, we used last filed flight plans (M1 files) to build the waypoint network (see also Deliverable D1.3 for details). The crossing times between waypoints are also extracted from the data, by computing the average crossing times of the real flights.

The sector capacities have been computed in the following way: we considered all flights of the set and we computed the load of each sector. We remind that the load is the number of flights using the sector space in an hour. The capacity has then been defined as the maximum load detected during the period of time. Note that this is in fact a lower bound on the real capacity (assuming no violation of capacity ever occurs).

1.7.1.2 Statistics

As done in section 1.6.6, here we show some statistics on the relevant metrics for the model, except here we restrict our analysis to the French airspace. The left panel of Figure 1.38 shows the histograms of the crossing time between waypoints. Interestingly, this distribution has a similar shape as the one observed in section 1.6.6 for crossing times between sectors. In fact, as the central panel of Fig. 1.38 shows, a lognormal distribution with parameters $\mu = 1.35 \pm 0.03$ and $\sigma = 0.74 \pm 0.02$ fits remarkably well the distribution of crossing times. This similarity was not obvious a priori, since the sector network is different by nature from the waypoint network, even though both are quasi planar. In fact, we tested that planarity itself does not explain the shape of distribution. This can be understood also by the presence of tails on the right of the real distributions, corresponding to long edges impossible to have in a planar graph.

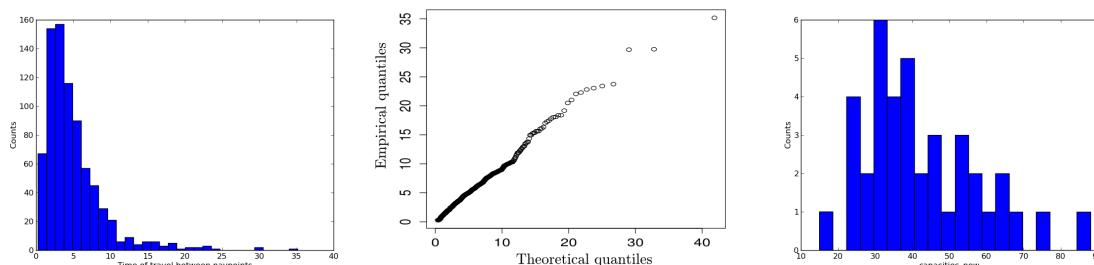


Figure 1.38: Histogram of the crossing times between waypoints (left) and the estimated capacities (right). Crossing times are in minutes. The central panel shows QQ plot of the crossing times between waypoints and the best fit with a lognormal.

We now consider the distribution of estimated capacities. The right panel of Figure 1.38 shows the histogram of the sample, which has mean value 42.7 (median equal to 39) and standard deviation 15.7. These values are quite close (but smaller, as expected) to the real capacities of the Italian airspace we obtained in the validation activity. The average capacity of the sectors in the airspace LIRCENTR is 48.2, whereas for the airspace LIRNORTH is 44.8. These results indicate that the capacity measure we use are quite close to the real one, even if slightly underestimated.

Moreover we remind that we cannot directly compare the right panel of figure 1.38 with its analog in section 1.6.6. Indeed, model 1.0 and model 2.0 use different proxies for capacities, the first based on the maximal number of flight simultaneously present in a sector and the second with the maximal traffic load (in one hour). However the two proxies are strongly related. Figure 1.39 shows the scatter plot of the two proxies of capacity and it is clear they are significantly correlated ($\rho = 0.77$). A best linear fit gives a proportionality constant of 2.8. Hence, apart from a proportionality constant, the proxy used in model 1.0 is a faithful representation of the true capacity.

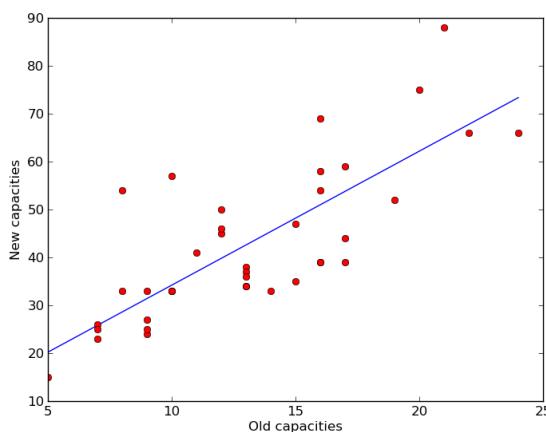


Figure 1.39: Scatter plot capacity proxy equal to the maximal number of flights simultaneously present in a sector (in abscissa) and capacity proxy equal to the maximal traffic load in one hour (in ordinate). Each circle represents a French sector. The blue line is a linear regression. The slope is 2.80 ± 0.16 and the ordinate at origin is 6 ± 30 . The Pearson correlation coefficient is 0.77.

1.7.1.3 Calibration of the model with real data

Given that we use a significant amount of real data to setup our model, we can use the remaining parameters of the model to calibrate it. Clearly we choose to use the unobservable parameters of the model for the calibration. These are the parameters which describe the strategic behavior of air companies. In a pure population setting, without departing noise, and having fixed $N_{fp} = 10$ and $\tau = 20$ minutes, the remaining behavioral parameters are β/α , N_{sp} , and f_S . We have found that the calibration of the model is not very sensitive to the ratio and therefore in this phase we will set $\beta/\alpha = 1000$. The only parameter on which we are performing the calibration is therefore N_{sp} , i.e. the number of shortest paths of navpoints per shortest path of sector, as described in section 1.3.4.

In order to calibrate the model we study how N_{sp} affects the distribution of four key navpoint network metrics, namely degrees, strengths, weights for the network, and topological length (number of navpoints crossed) for the flights. We search the value of N_{sp} that produces simulations reproducing as close as possible the real distribution of the above metrics.

In figure 1.40 we show an example of two of these distributions, the degree and topological length distribution, each of them for two extreme values of the control parameter N_{sp} , namely 1 and 10. The figure shows that the value of the control parameter has a strong influence on the distributions. Visual inspection of these graphs indicates that the choice $N_{sp} = 10$ gives a better agreement than $N_{sp} = 1$.

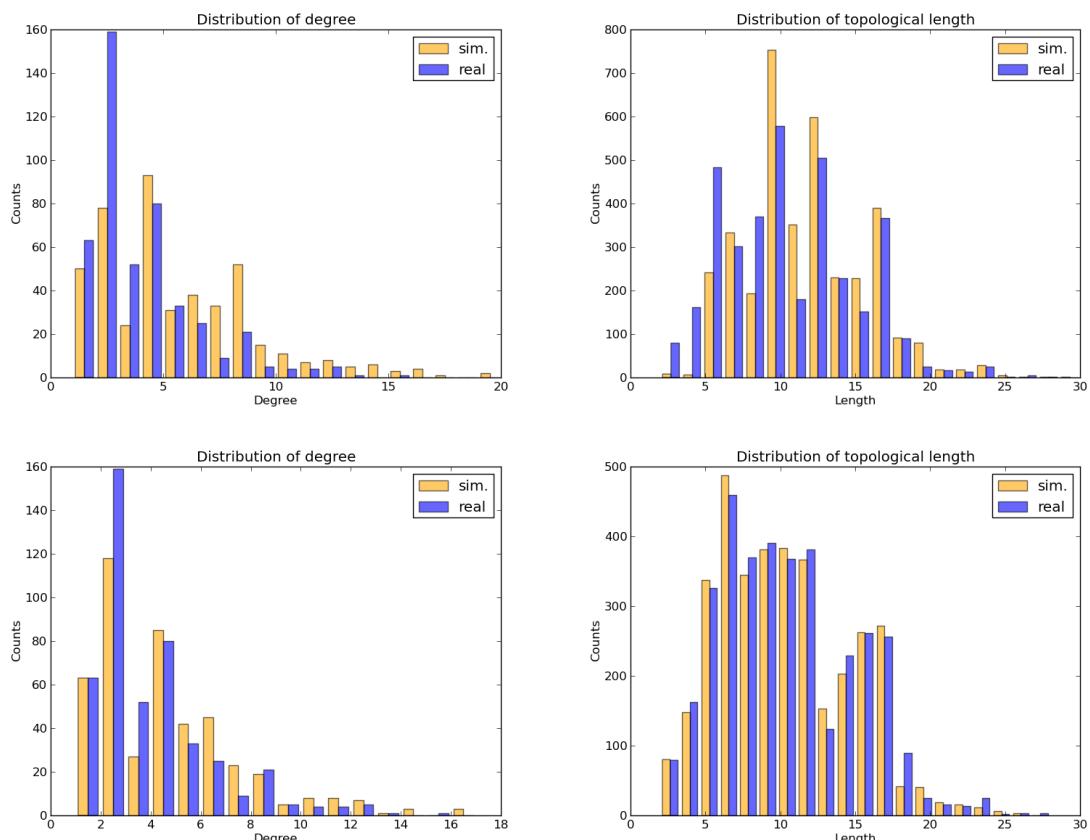


Figure 1.40: Distributions of degree (left panels) and topological length of flights (right panels) for $N_{sp} = 1$ (top panels) and $N_{sp} = 10$ (bottom panels). For each panel, the real and simulated distributions are shown.

In order to have a quantitative criterion to select the best agreement, we use the Kolmogorov-Smirnov distance (from the eponymous test). This statistical distance computes the biggest difference between the cumulative probability distributions of two different samples. Hence, its value lies between 0 (the cumulative distributions are exactly the same) and 1. We compute it for each of the four metrics above and for each value of N_{sp} , between 1 and 10. Figure 1.41 shows the results of the procedure. As we can see, two of the metrics are not really sensitive to this parameter: the distributions of weights and strengths are roughly the same regardless the value of N_{sp} . On the other hand, the distance between the degree distributions is dropping by almost 50% between $N_{sp} = 1$ and $N_{sp} = 10$. The distance between distributions of lengths drops from 0.2 to nearly 0. Hence, $N_{sp} = 10$ gives the best agreement between the simulations and the data, as qualitatively seen from Figure 1.40.

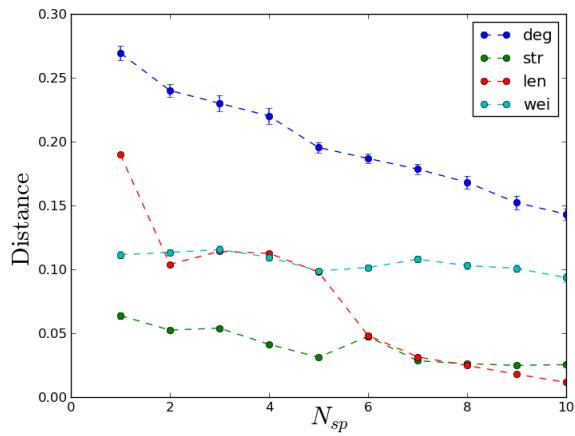


Figure 1.41: Kolmogorov-Smirnov distance between real and simulated distributions of degree, strength, weight and topological length of flights of the navpoint network as a function of N_{sp} .

As a consequence, we will consider this value in the following. Note however that this is a very peculiar value, for reasons we will explain in the following. Hence, we will also briefly investigate what happens when we take other values of the parameter.

1.7.2 Testing the model

Figure 1.42 shows the outcome of a simulation with the above setting by displaying the occupation of the edges of the navpoint network. The system is very well organized along major routes, which are usually straight. This particular simulation has been done with only one type of companies (type R, with a high ratio β/α). In this simulations, of the 3584 flights, only 289 flight plans have been rejected, and only 27 flights had all their flights plans rejected.

One important question at this point is the following: how much has the system been already regulated? Since we do not know the initial flight plans, one way to answer this question is to disturb the times of departure and see how much it affects the output. If the system is already heavily optimized, small deviations should increase drastically the number of rejected flights. On the other hand, if the system is far from its limits, it means that the times of departure we observe are the desired ones. Hence, some perturbations should not have any impact.

The noise we add is Gaussian, centered on this initial desired time with a variance δt_0^2 . In the case where $N_{sp} = 10$ we do not see any increase of the rejection with δt_0^2 . Hence the system seems

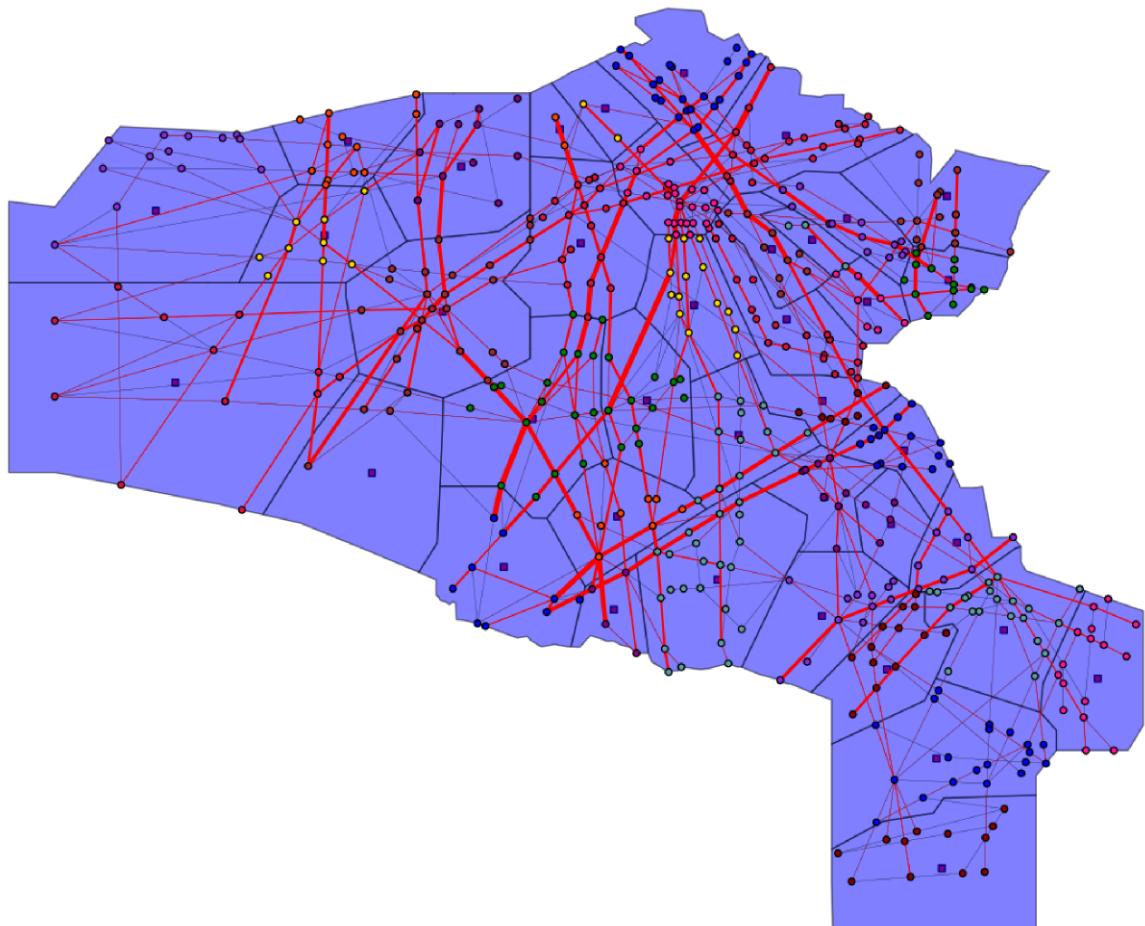


Figure 1.42: This map is the result of a run of the ABM on the real network. The blue patches represent the sectors and the round points are navpoints, their colors indicating the sector they belong to. The thickness of the red edges is proportional to the traffic.

to be far from its limit.

Note that this is not the case when we set $N_{sp} = 1$ for instance, as figure 1.43 shows it. Here, we see a clear increase of the rejection up to 12 minutes before it decreases again. This decrease can be understood with the following argument: when the flights are heavily shuffled, it helps breaking the natural “wave pattern” and daily pattern. On the other hand, the increase shows that the system is already optimized. However, the magnitude of the increase is quite small, which indicates a small amount of regulation. The relationship between this behavior and the calibration itself is unknown.

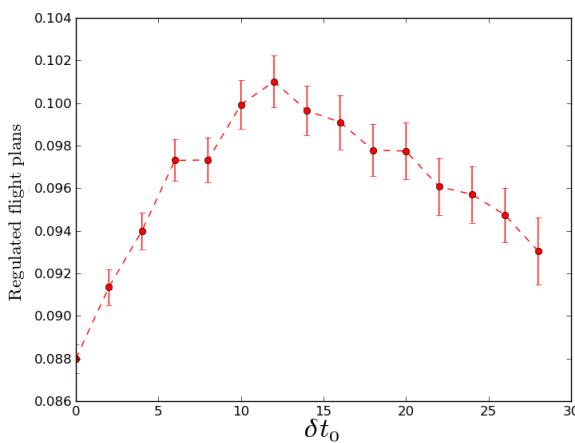


Figure 1.43: Fraction of flight plans, computed as the number of flight plans rejected divided by the total number of *flights*, against the standard deviation of the noise on desired departure times. The air companies have all a high β/α ratio. The error bars are the standard errors.

1.7.3 Pure population

1.7.3.1 $N_{sp} = 10$

We now investigate the results of the simulations where there is only one type of company. Figure 1.44 shows what happens when there is only one type of company, as a function of the logarithm of the ratio β/α . As we can see, there is clear effect on the rejected flight plans, going from 3% up to 8%. More precisely, starting from a pure population S (on the left), there is a constant increase of rejection until it hits $\beta/\alpha = 1$. Then we have a plateau in the region where AOs reroute their flight instead of delaying them. This last part is probably due to the fact that, given the small number of *flights* rejected, only the first flight plans of each flight are in fact rejected. Since the ratio plays a more crucial role in the last flight plans, which are not reached by the NM, the number is constant at the end.

The number of rejected flights follows a similar increase, although the absolute values are 10 times smaller than the rejected flight plans, as it is expected: it needs 10 flights plans to be rejected in order to have one flight rejected. Finally, the satisfaction is consistent with the other plots, showing a clear decrease followed by a plateau. Comparing these results with figure 1.20, we seem to be in an equivalent regime of model 1.0 where $\Delta t = 23$. The comparison is not valid in fact, for the following reason.

Figure 1.44 indicate clearly that companies of type S have an advantage on company of type R, which was happening previously when the waves were far one from each other, thus allowing the

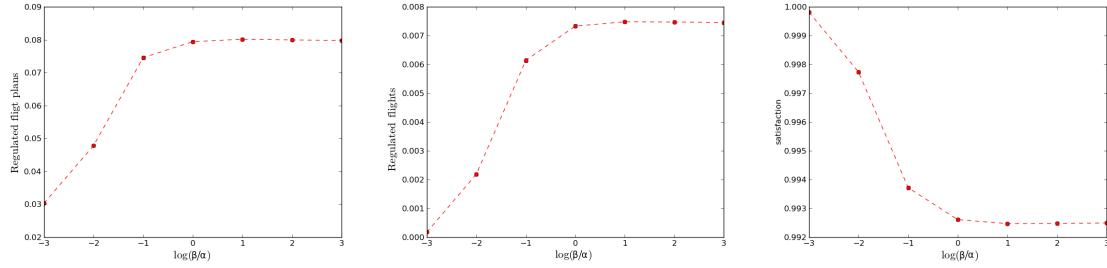


Figure 1.44: Variations of the fraction of rejected flight plans (left), the fraction of rejected flights (center) and the satisfaction (right) with the logarithm of the ratio β/α .

delaying the flight plans. However here it is not due to the time pattern. In fact, it is impossible for company R to have an advantage. Indeed, when we set $N_{sp} = 10$, it means that we compute 10 navpoint paths per sectors path. Since there are $N_{fp} = 10$ flight plans in total for a flight, then there is only one sector path available¹⁴. For company S, it does not change anything: it only has one path and then shifts in time. On the contrary, companies of type R have only one sector path. When their first flight plan is rejected, the second one – taking the same sector path – hits the same full sectors and thus is rejected too. In fact, since the departing time is the same in all flight plans, the entry and exit time of each sector will be very close in each plan. Hence, company R has *no advantage at all* for this choice $N_{sp} = 10$. Company S is necessarily doing better, whatever the setup.

For this reason, we also show the results with an opposite choice for the parameter: $N_{sp} = 1$.

1.7.3.2 $N_{sp} = 1$

In this setting, the flight plan submission is directly comparable to the one of model 1.0. Indeed, 10 sectors paths are now available to companies, exactly like in the previous model. Figure 1.45 shows the related plots. First, we notice that the absolute values are higher for the rejections: from 10% up to almost 20% for the flight plans for instance. More precisely, we see that, starting from a pure population of companies S, there is an exponential *decrease* of the number of rejected flight plans, until we reach a plateau where increasing the ratio does not affect the number anymore. Hence, it is exactly the opposite of the plot of figure 1.44. Company R has gained back its advantage, and it is now able to outperform company S.

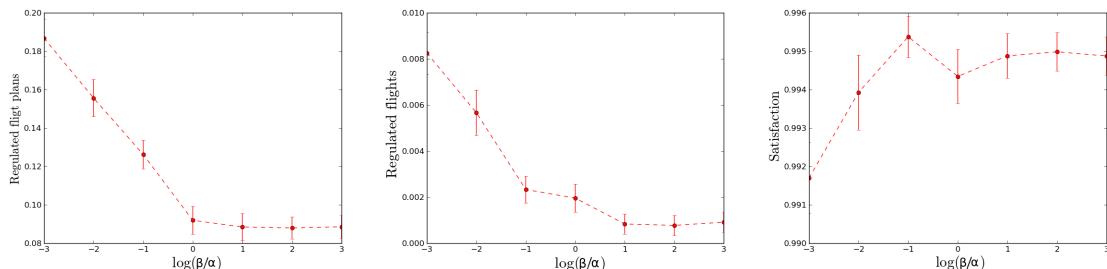


Figure 1.45: Variations of the fraction of rejected flight plans (left), the fraction of rejected flights (center) and the satisfaction (right) with the logarithm of the ratio β/α .

¹⁴To be precise this is not always true, since if there are not 10 paths inside the sector path, one considers another sector path.

The number of flights rejected follows a similar increase. The fact that when decreasing β/α , the increase of rejected flights is slower than the increase of rejected flight plans might be due to the fact that, when decreasing the ratio, the NM rejects more and more flight plans but does not reach the last one, which triggers the rejection of the whole flight. Finally, the satisfaction is consistent with the other plots, although it seems to display a non-monotonic behavior, with a drop in 0. In fact, this might be due to the following reason: for $\alpha = \beta = 1$, each deviation of the desired time or shortest path is costly. Hence, if a flight does not have its first flight plan selected, the satisfaction drops quite a lot. On the other hand, if $\alpha = 1$ and $\beta = 0.1$, the deviation from the desired time is not so costly. Hence, even if the first flight plan is not selected, the satisfaction does not drop very much.

This time, we can compare with figure 1.20, and we see that the setting is similar to $\Delta t = 0$, with almost no structure of waves. This is a result we are going to find in the following section too.

1.7.4 Mixed population

1.7.4.1 $N_{sp} = 10$

Here, as in 1.6.3.3, we focus on two kinds of companies, companies S and companies R. We fix N_{sp} back to the value 10.

Figure 1.46 shows the evolution of the measures of global satisfaction versus the fraction of company S present in the system. All panels show a somehow expected behavior. The values start from the extreme left values of figure 1.44 and ends up at the extreme right value. In between, the behavior is linear.

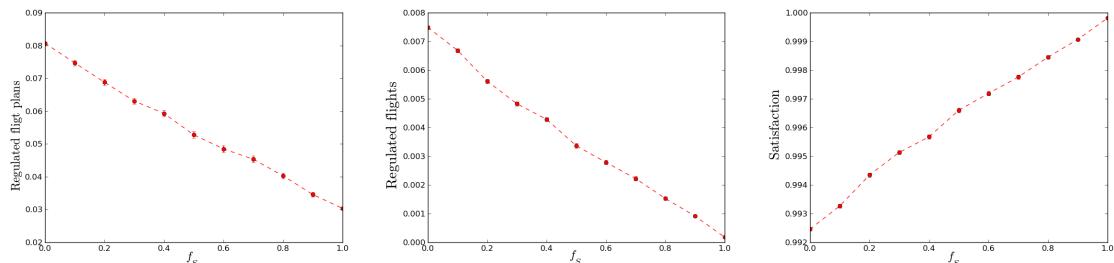


Figure 1.46: Evolution of the fraction of regulated flight plans (left), the fraction of regulated flights (center) and the global satisfaction (right) versus the fraction of company S in the system. Error bars are standard errors.

It is interesting also to investigate how much each company contributes to the overall evolution of the above metrics to inquire whether the observed patterns are the effect of the mixing parameter f_S . Figure 1.47 shows that in fact the number of rejected flight plans decreases for company S with f_S , thus displaying the opposite of the “the loner, the better” effect seen for model 1.0. The same conclusion holds for company R. However, the magnitude is quite small and thus the evolution seen on figure 1.46 is mainly due to the mixing parameter itself. Finally, note that the absolute value are twice as small for company S than for company R, as expected from the fact that the latter does not have any advantage in this setting.

1.7.4.2 $N_{sp} = 1$

Once again, we show the same results obtained for $N_{sp} = 1$ in order to see the differences. Left and center panels of figure 1.48 show the same kind of behavior than before. The rejections start

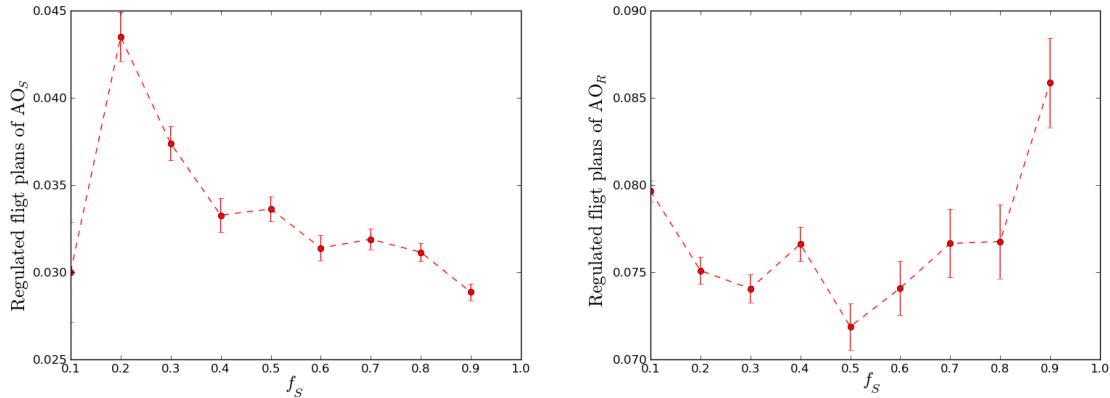


Figure 1.47: Evolution of the fraction regulated flight plans for company S (left panel) and company R (top panel). Error bars are standard errors.

from the extreme left values of figure 1.44 and end up at the extreme right values with a quite linear evolution. Of course, the extreme values are here different from those of figure 1.46, which ends up in an *increase* of the rejected flights with f_S instead of a decrease.

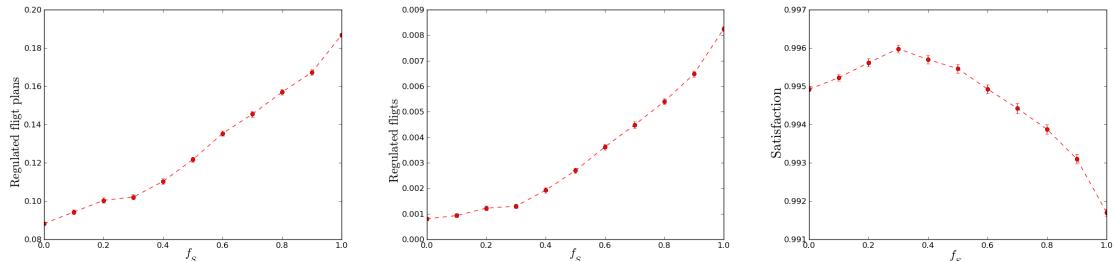


Figure 1.48: Evolution of the fraction of regulated flight plans (left), the fraction of regulated flights (center) and the satisfaction (right) versus the fraction of company S in the system. Error bars are standard errors.

On the other hand, the right panel is more interesting. In fact, it seems that the satisfaction reaches a maximum value at an intermediate value of the mixing parameter. This would mean that, in line with the case of section 1.6.3.3 (see fig. 1.22), the best overall satisfaction is achieved when there is a mix of companies, and not one major company. By comparing with figure 1.22, we see that it corresponds to the case where Δt is of order 1, i.e. an almost uniform pattern of departing time, as it was already noticed in the pure population case. This is somehow non surprising since here we included the entry/exit points in the analysis, on the contrary of the one of section 1.6.3.3 where there are only two airports. Hence, the departing times are much more spread in time. Nevertheless, it is a striking result that the multi-airports setting does not destroy the existence of a maximum.

Since we find a non-trivial pattern for the total satisfaction, we inspect the satisfaction of each type of companies separately, as it was done previously. Figure 1.49 displays the fraction of regulated flight plans and the satisfaction for each type of company.

First, we see a drop in the fraction of flight plans rejected of R for intermediate values of f_S , although by a very small amount. The two extremes seem roughly on the same level. The satisfaction

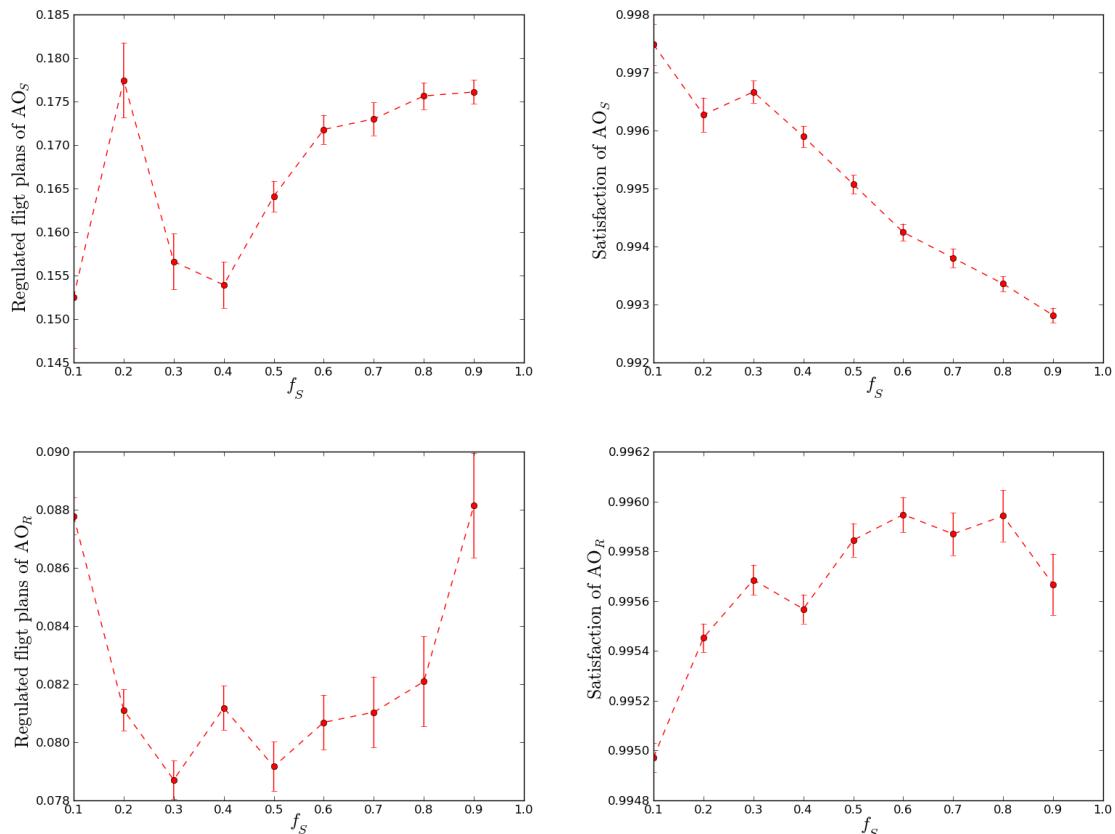


Figure 1.49: Evolution of the fraction regulated flight plans (left panels) and the satisfaction (right panels) for company S (top panels) and company R (bottom panel). Error bars are standard errors.

seems to increase, but again by a very small amount. Thus, companies R are almost insensitive to the level of mixing.

On the other hand, the fraction of rejected flight plans is not monotonic for companies S. From a small value, it starts increasing , reaching a maximum in 0.2, and then declines to a minimum around 0.4, before increasing again up to 1. The satisfaction shows quite the opposite behavior, although the whole decrease is bigger in magnitude. Note that, in line with results from model 1, the best situation for companies is the one where they are surrounded by companies of the other type. However, the not monotonic shape of their satisfaction leads to the one of the global satisfaction.

1.7.5 Simulations of scenarios where sectors are closed.

We now show an example of the use of the ABM as an empirically grounded scenario simulator by using the procedure of shocks we have presented before. We simulate the shutdown of some sectors on the real network, previously calibrated on the data, and see how companies react and how the sector load changes. We fix $N_{sp} = 1$ here.

First, figure 1.50 helps visualizing the impacts of a sector closure on the planned trajectories. As we can see, the traffic increases in the surrounding sectors because flights have to avoid the closed sector.

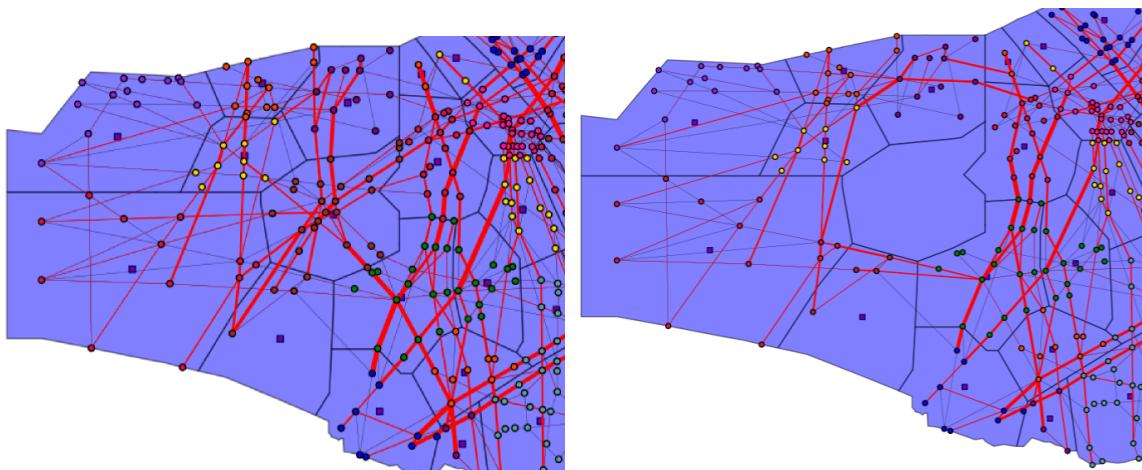


Figure 1.50: Portions of the French airspace where a sector has been shut down. Left: the initial traffic. Right: the traffic after the shock.

As done with model 1.0, we try to see if some types of companies are more resilient than others to sector closure. Figure 1.51 shows how the AO satisfaction depends on the number of shocks, for companies S and companies R (in a pure population setting). As we can see, the satisfaction decreases monotonically with exactly the shape for both companies. Hence, it seems that, in contrast with model 1.0, company S does not gain a relative advantage from the shutdown of sectors. The same results are obtained in the mixed population case and when we fix $N_{sp} = 1$ (not shown here).

We can also use this procedure as a scenario simulator to assess how much a sector is important for the airspace. To this end, we show a map on the left panel of figure 1.52 where we report the drop of satisfaction incurred by the *whole system* when each sector is individually closed. It is clear from this map that not all sectors are equally important to the airspace. In particular, the sector around Paris and the one in the South West seem crucial. This map, however, takes also into account the flights which were initially departing/arriving from these sectors and cannot depart/arrive anymore. Hence,

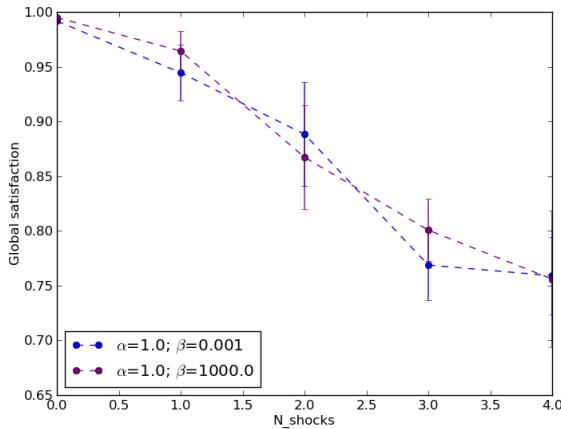


Figure 1.51: Satisfaction of companies S and companies R as a function of the number of closed sectors.

the satisfaction can drop significantly only because these flights are suppressed and not because the traffic is disturbed. For this reason, we provide on the right panel of figure 1.52 the corresponding map. This time we computed the satisfaction only on the flights which were not initially departing or arriving in the sector. The picture is now quite different. As intuitively expected, we find that central sectors play a bigger role now with respect to the peripheral ones. As a consequence, the sector in the South West does not play an important role anymore, whereas the sector around Paris is still crucial.

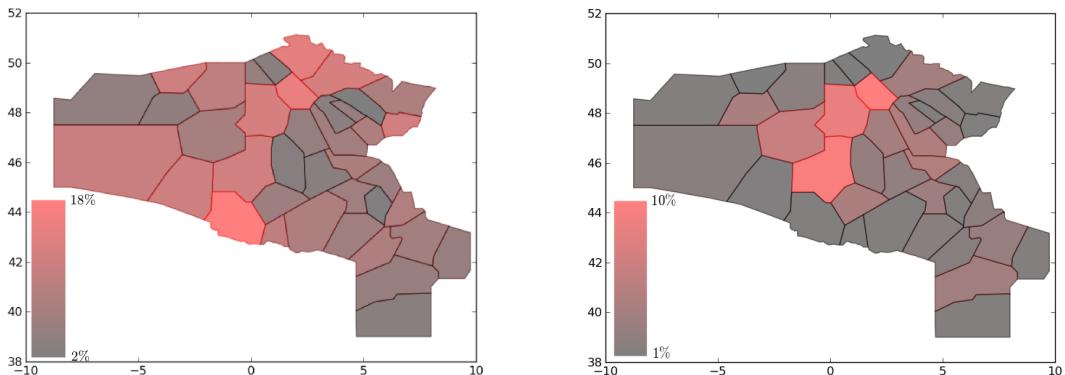


Figure 1.52: Drop of satisfaction of the whole system when a sector is closed. Left: satisfaction is computed on all flights. Right: satisfaction is restricted to flights which do not depart or arrive in the given sector.

1.7.6 Conclusions

The setup of model 2.0 in a real environment implies that the choice of parameters is drastically reduced. Hence, we are able to perform a quantitative calibration on some of the remaining parameters. We chose N_{sp} as the calibration parameter, which allows to keep the behavior of companies as variables. However, a different choice for N_{sp} is interesting for two different reasons. First, it allows us to directly compare the result of model 1.0 with these ones, to see if the mechanisms we have explained previously are still useful to understand what is going on in the new setup. Second, because it gives us a indication of what would happen if the companies were acting differently, by allowing longer paths for instance. The results could be a better repartition of the load in the airspace. Indeed, we find that it could allow a maximum of global satisfaction for the system involving two types of air companies. On the other hand, keeping N_{sp} to its calibrated value only allows a maximum of satisfaction made of one pure population.

Finally, we used the procedure of shocks as a scenario simulator where the companies have to react to the shutdown of a sector. Crucial sectors can be highlighted this way. This procedure should be used to see if different behaviors from the agents can relieve the pressure on these sectors and spread more efficiently the load on the whole airspace.

2 Tactical Layer

2.1 Introduction

The agents of the tactical layer of the ELSA agent based model are **aircraft/pilots** and **controllers** who are active at the level of **ATC sectors**.

In our agent based model we simulate the events that make M1 transform into M3, with the aim of investigating the issues that affect the predictability of the last filled flight-plan within the ATM system. The specific scientific questions we are investigating are:

- What are the issues that affect the predictability of the last filled flight-plan within the ATM system? How is the predictability affected by these issues?
- Does the European Air Traffic predictability improve by moving from a trajectory-based system to a business-trajectory-based system?

We plan to address these issues in two steps. In the current scenario Model A (A stands for Actual) we will address predictability issues by looking at the current scenario. In Model F (F stands for Future) we will consider a transition to the future SESAR scenario and try to see whether predictability improves moving to a business-trajectory regulation scheme.

2.2 General features

The interaction between the agents is needed in order to manage the tactical changes occurring in the system due to unforeseen events, i.e. weather events, congestions, limitation of sectors capacity, etc. Moreover, the ATC sectors are the places where flight trajectories are made conflict free. Consequently, we will try to model and simulate the events that make M1 transform into M3 by considering agents that operate at the level of ATC sectors.

2.2.1 Current scenario Model A

We have two types of agents : the **aircraft/pilots** and **controllers** active at the level of **ATC sectors**. These agents interact with each other in order to resolve two main issues:

- due to the fact that the M1 flight trajectories are not conflict-free pilots/aircraft and controllers have to interact in order to make them conflict-free.
- further interactions are due to the fact that during the en-route phase of the flight there may happen weather events or congestion events that require a modification of the flight trajectory.

In the current version of the ABM model we are giving to aircraft/pilots agents the opportunity of modifying the aircraft velocity in order to solve safety events. We are aware that this strategy is

actually used by controllers only in specific cases. However, we decided to explore this possibility because this will nevertheless be a possible strategy adopted by companies and controllers in the future SESAR scenario. Indeed, we have constructed the code in a modular way that allows to exchange the order of the strategies adopted by the controllers. In fact, we can easily modify the code in such a way that controllers first check for the possibility of doing re-routings and then check for the possibility of changing velocities. Another major feature of the model is to take into account the possibility that the conflict resolution module changes the aircraft flight levels in order to solve the conflicts.

In the simulations of the current version of the ABM model included in this deliverable the controllers only perform two types of actions: they give directs in order to facilitate the airlines operations and they resolve possible conflicts between trajectory pairs by issuing horizontal re-routings or vertical flight changes. In fact, we do not use the velocity change module. This is an oversimplification of the controllers behavior, but we consider it to be useful as we are trying to capture the essential elements of controllers behavior without making the model overly complex. We are planning to consider the velocity change module in the simulations of the future SESAR scenario. The re-routings we are considering here are tactical changes of trajectory mainly issued at the level of sectors. Therefore we will refer to these horizontal trajectory changes as to *vectoring*.

The model assumes that M1 trajectories are not conflict free. Thus one main task to be performed within the model is to deconflict trajectories. Moreover, we simulate a shock in the system and see how the system reacts to it. Specifically we simulate a shock in an area around a navigation point. We assume that the shock lasts for a certain time span. Operatively, this means that for a certain time span a certain area of the ATC sector can not be crossed by flights. This might correspond to a situation where an extreme weather event occurs as well as to a situation when a certain area is highly congested and therefore the air traffic must be deviated. As a result, another tasks of the model is to change one or more flight trajectories in order to avoid the shocked areas. The way we plan to model this step is to deviate the flight trajectory along new navigation points that are external to the restricted area and with the constraint that (i) we want to minimize the length of the deviated trajectory and (ii) the deviated trajectory must be conflict free. We will perform different simulation experiments changing the statistical features of the shocks.

These tasks are accomplished as follows

- we have checked that M1 trajectories are not conflict-free, see below section 2.3.8. This implies that a lot of the modeling efforts have to be devoted to the efficient resolution of conflicts. In the D2.1 draft model we had already an algorithm whose task was to solve conflicts. However, this algorithm was thought to solve the few conflicts that may arise when a planned trajectory is modified due to the existence of a shocked area. This algorithm was quite inefficient to solve conflicts when all possible trajectories must be compared with each other. The original algorithm for generating M3 trajectories has been modified in order to take into account three different situations:

1. *there is the onset of a shocking area.* In this case we select a new navigation point for each flight trajectory such that the new trajectory has the minimal distance from the planned one. However, we allow for changing the aircraft velocity in the modified trajectory segments, if necessary. This enables us to select trajectories characterized by better fitness measures. This algorithm is therefore essentially based on vectoring, possibly augmented with the possibility of changing velocities.
2. *there is a possible conflict of trajectories that nevertheless do not intersect one with each other* - We are implementing a two-level algorithm. This algorithm mixes together the search of a new navigation point along the lines discussed below with an approach that, based on a genetic algorithm, searches the optimal velocity in the mutated trajectory

segment, as to ensure that the new trajectory exists. The advantage of this two-level algorithm is that it is fast. Indeed, it must be taken into account that aircraft can change their velocity within a range that depends on the flight level, the type of aircraft and other variables. As a preliminary choice we will consider velocity changes of the order of 5 % of the previous en-route velocity.

3. *there is a possible conflict of trajectories that intersect with each other* - When solving conflicts between trajectories that intersect with each other we use the algorithm illustrated above, which proves to be efficient also in this case. From a logical point of view there is no difference from the previous case. The algorithm treats these two situations in the same way. We keep this case distinct from the previous one to emphasize that the present case usually occurs in the planned trajectories, while the previous one usually occurs when one of the two conflicting trajectories have already been deviated.
- Based on the inputs from the ENAV operational experts, we have introduced a new feature in the model. When a trajectory is deviated, then it is not sent back to its planned trajectory. Rather, it is sent to the planned exit navigation point of the sector. This helps us in implementing a feature already planned in D2.1, i.e. the fact that airplanes are given directs within a sector.

2.3 Implementation of the Model

The code that implements the model is written in Phyton [6]. However, some modules have been written in C [7] in order to improve the computational efficiency of the ABM. Below we describe the modules that compose the tactical layer of the Agent based model.

2.3.1 Navigation Points module

Given the sector, we have populated it with navigation points. On one hand, part of the navigation points selected are real ones, i.e., those crossed by the flights according to their M1 last-filled flight-plan. On the other hand, other navigation points are generated randomly from an uniform distribution (e.g., 1000) and we have considered only those falling inside the polygon of the given sector (i.e., we have excluded those on the boundaries). These new navigation points could be seen as !-points in the M3 flight plan. However, not all of them will be really used in the flights deviations. Only a set of them will be selected. All the not used ones will be eliminated from the analysis after all the flights in the sector will be checked. An illustration of all the navigation points we use is provided in figure 2.1. Those colored in blue are the real ones, in red are the randomly generated ones. We have not generated navigation points on the boundaries because these are sensible points and need to be treated separately. Indeed, since these points connect two or more sectors, it is required coordination between the reference controllers. Usually, controllers give directs inside their area of reference changing the navigation points a flight has to pass through, while they rarely substitute those navigation points that are located on the boundaries and that can affect the air traffic management of the neighbor controller. In other words, different aircraft can cross sector borders in different places. However, a specific aircraft, even though it was re-routed, will cross the border in the planned navigation point.

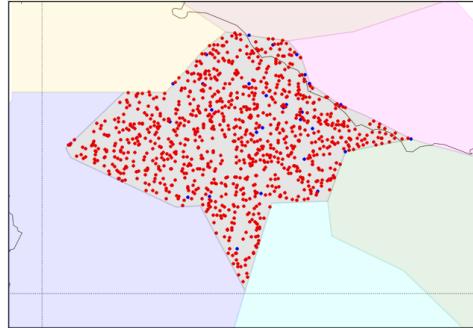


Figure 2.1: Visualization of the navigation points active during the whole AIRAC 334 in the selected sector. The points in blue are the real one derived from the M1 last-filled plan, those in red are randomly generated.

2.3.2 Flight List module

Once the sector has been populated with navigation points, we create a list FL_k of flights active in that time-step. Such list will be reshuffled in the next time-step. Within this list we check whether the flights are crossing a shocked area and whether or not they conflict with other trajectories. Specifically, the i -th aircraft in the list will be checked against all other $j < i$ flights. When a trajectory modification is needed, it will affect the i -th flight. The reason for shuffling the list at each time step is in order to avoid that the trajectory modifications are implemented always on the same aircraft.

2.3.3 Conflict detection module

In order to check for conflicts between two flights, we use a data structure that considers the aircraft localized inside the trajectory segments travelled within the given time-step Δt . For this purpose, we introduce a finer subdivision of the time-step into N elementary time increments δt and compute the real space-time position of the aircraft at each elementary time increment, by assuming a constant velocity, see Fig. 2.2. The conflict detection algorithm will have to simply calculate the positions of the aircraft for each of the elementary time increments and then compute the distances between the two aircraft at these positions. Suppose we are now checking if the i -th flight trajectory is conflicting with all other f_j trajectories, with $j < i$. We are therefore considering a number i of flight trajectories. For each of them we have an array \mathcal{P}_j , $j = 1, \dots, i$ of positions computed according to the algorithm illustrated above. For each of the elementary time-increments, we compute an array of distances d choosing as a value for each array element the minimum distance between the i -th aircraft and all the other aircraft in the list FL_i with $j < i$. From the distances array it is possible to estimate a *fitness value* to maximize. By assuming that the safety distance threshold is d_{thr} , and $X = \{d_i\}$ is the subset of distances below d_{thr} , the fitness value is defined as $\mathcal{F}_1 = \sum_{i \in X} (d_{thr} - d_i)$. If this value is different from zero then there is a conflict and the algorithm proceeds to the next module that makes the trajectories conflict free. As a result the computational time increases linearly with the number of aircraft. Moreover, this method allows us to introduce a simple way to manage possible changes of aircraft velocity. Of course the conflict detection module is not in place when the

necessity for vectoring is due to the existence of a shocked area.

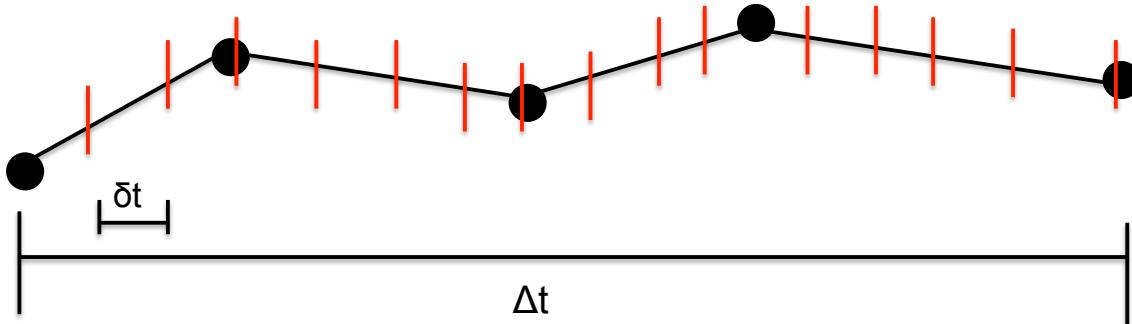


Figure 2.2: The management of the trajectory is done within a $\Delta t = 240$ secs time horizon. The flights are sampled every $\delta t = 1/3\Delta t$ secs by linearly interpolating if necessary.

2.3.4 Conflict resolution module

After the check for conflicts has been done, this module searches for a new conflict-free trajectory. It is conceived as a three-step algorithm that acts on the velocities of the aircraft, the search of a new trajectory (vectoring) and the change of flight level, in case conflict exists. The order by which the three steps are applied might be changed. Here we describe them.

2.3.4.1 Vectoring sub-module

The first step of the module we present here is the one that performs the vectoring. The procedure is illustrated in Fig. 2.3. We first identify the two navigation points B and A which are before and after the conflict area (crossed circle in the figure), respectively. The idea is to (i) keep B , (ii) substitute A and (iii) eliminate all the other subsequent navigation points but the last one L . To do that, we take the previously generated temporary navigation points T_k (squares in the figure) and we order them with respect to the angle that the segment connecting B and T_k forms with the original trajectory. We select the temporary navigation point such that the new trajectory has the smallest length. Let us call α the angle between the planned and re-routed trajectory. We admit angles such that $\alpha \leq \alpha_M$ where α_M is the maximum admissible angle, see section 2.3.8.

Having this new navigation point we compute again all the distances with the $j < i$ trajectories and compute again \mathcal{F}_1 , see Fig. 2.4. If $\mathcal{F}_1 = 0$, then we select this navigation point, otherwise we go to the navigation point with the second smallest difference in length. This procedure is iterated until we find $\mathcal{F}_1 = 0$. If we find only navigation points with an angle $\alpha > \alpha_M$, the algorithm exits this module and go to the next one.

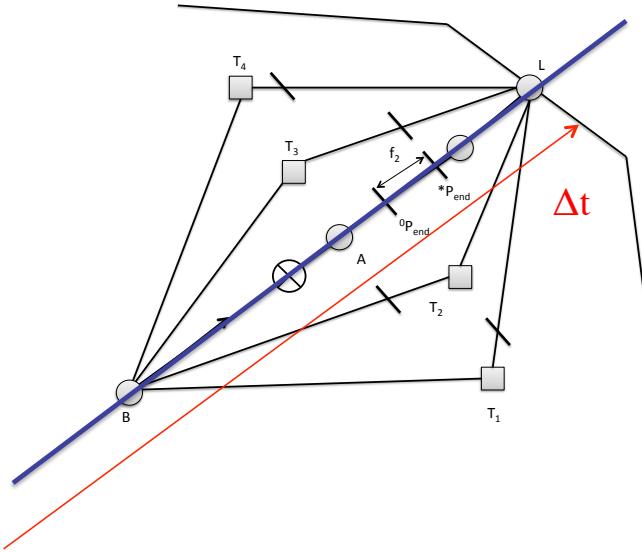


Figure 2.3: The Fig. illustrates the techniques of rerouting after the rescheduling of velocity. Every possible rerouting is tested with the genetic algorithm for the velocity. This would define a point P_{end} . The route selected will be the one such that minimizes the fitness \mathcal{F}_2 , i.e. it minimizes the distance between the point P_{end}^* of the original route and the P_{end} of the new route. The crossed circles identifies either the shocked area or the trajectory conflict. Circles identify true navigation points. Squares identify randomly generated navigation points.

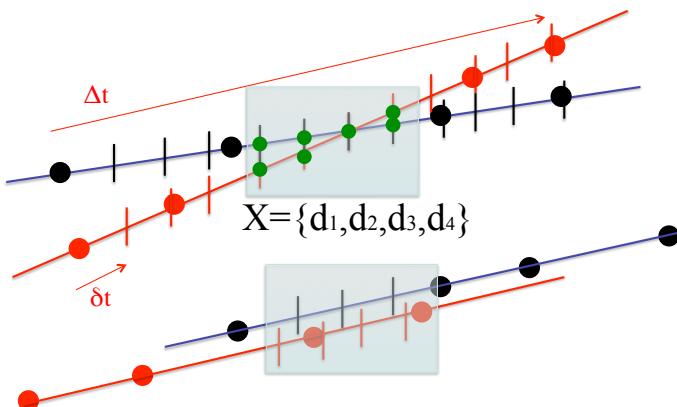


Figure 2.4: Illustration of the way the fitness function \mathcal{F}_1 is computed. The green points are those whose distance is below threshold.

2.3.4.2 Flight level change sub-module

The second step of this module involves changes of flight level. The model implements one central flight level and ten other possible flight levels, five above and five below the central one. Flight levels are further separated considering the direction in which they are flown. All flights are initially considered to be active in the planned (M1) flight level. Therefore they can move upwards or downwards whenever the vectoring is not feasible. The choice of the new level is done by considering

the one where there is less probability of having conflicts. This is assessed by computing the sum of the \mathcal{F}_1 functions for all the flights in the two external flight levels and choose the flight level with the smallest one. If the two sums are equal then we choose the upper flight level. If no level is available then the algorithm exits this module and go to the next one.

2.3.4.3 Velocity change sub-module

The third step for changing the flight velocity is mainly based on a genetic algorithm [5] that minimizes two fitness functions by using a mix of cross-over and mutation operators, see appendix D for further details about the considered operators. Suppose we are now checking if the i -th flight trajectory is conflicting with all other f_j trajectories, with $j < i$. We are therefore considering a number $i - 1$ of flight trajectories. For each of them we have an array \mathcal{P}_j , $j = 1, \dots, i - 1$ of positions computed as illustrated above. When the conflict detection module identifies a subset of points within the j -th array \mathcal{P}_j such that these points are below the safety distance threshold, the ABM tries to solve the conflict using a new module that implements a genetic evolutive algorithm. This module is written in C language in order to make it computationally efficient. In our data structure we use a characteristic speed for each elementary segment between two navigation points. Assuming that before going out from the sector the aircraft will run into $N_{navigationpoint}$ navigation points, the aircraft will therefore change $N_{navigationpoint} - 1$ velocities during his trajectory. In this step we first generate a population N_{pop} of velocities arrays each of length $N_{navigationpoint} - 1$. We generate velocity arrays \mathcal{V}_s , $s = 1, \dots, N_{pop}$ filled with v_s velocities randomly selected under the condition that $v_{min} < v_s < v_{max}$, where v_{min} and v_{max} are the minimal and maximal velocity accepTable for the considered aircraft flying in the considered sector. For each element of this population we compute the array of positions \mathcal{P}_s , $s = 1, \dots, N_{pop}$ with dimension N . For each \mathcal{P}_s , $s = 1, \dots, N_{pop}$ the algorithm computes the \mathcal{F}_1 fitness value, according to the procedure illustrated in the vectoring module, see Fig. 2.3. The population of possible solutions (velocities arrays) will be sorted according to this fitness value. Another target to pursue is to minimize the delay with respect to the time scheduled in the file $M1$. To this end, we sort the elements with degeneration in the fitness value associating an estimator of this delay. Assuming that P_{end} and P_{end}^* are the end points of the aircraft according to the considered population and the scheduled flight plan, respectively, the $\mathcal{F}_2 = d(P_{end}, P_{end}^*)$ will be the second fitness to be minimized. The genetic algorithm searches for the optimal solution by applying to the population of velocities three different operators. Each one has a characteristic probability to be applied. Moreover, in order to avoid the selection of elements of the population that are too similar with each other, these operators will not be applied to the best element, but according to a probability function (a Gaussian distribution defined on the positive real axis, peaked in zero and normalized to unity in the interval from 0 to N_{pop}). Each new element generated with these operators will take the place of the worst element in the population. These operators are applied on a random basis with different probabilities as to ensure that local minima are avoided.

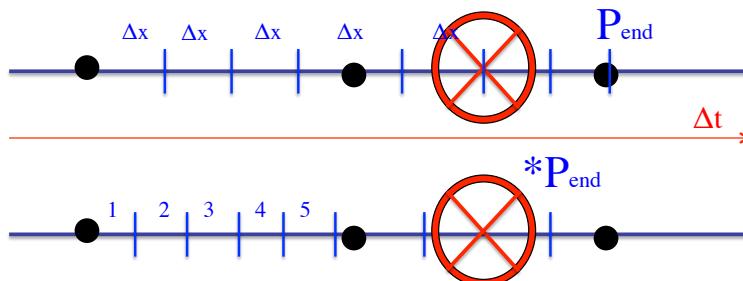


Figure 2.5: Illustration of the way the velocity change sub-module. The crossed circles identifies either the shocked area or the trajectory conflict.

2.3.5 Directs module

Following the suggestion from validation experts we have implemented in the model a module that assigns directs to the aircraft when traffic conditions are not so heavy.

The flight trajectory of the aircraft is changed with probability p by allowing that the aircraft directly reaches the planned exit point from the sector, whenever this does not imply a trajectory conflict. In the case of direct the aircraft remains on the same flight level.

2.3.6 Expected results

For each simulated flight we will monitor two variables. First we consider the number $A_{c,f}$ of actions that any controller has performed for each flight $f = 1, \dots, F$ and for each of the three types of possible actions: $c=1$ vectoring, $c=2$ flight-level change and $c=3$ velocity change. This is simply be the number of changes (velocity, position, flight level) operated on the planned trajectory for each type of action. We also consider as positive actions $A_{1,f}^{(+)}$ the ones when the controller gives a direct. We consider as negative actions $A_{1,f}^{(-)}$ all the others. As a second variable we consider the time $D_{3,f}$ needed by any aircraft to cross the sector of each flight $f = 1, \dots, F$. This must be compared with the planned time $D_{1,f}$. The delay will then be defined as $d_f = D_{3,f} - D_{1,f}$.

We believe that the model is now realistic and relatively efficient from a computational point of view. It will allow us to perform different types of simulation experiments aiming at understanding to which extent the sector capacity can be improved. In fact, having an automated algorithm for conflict resolution might lead to an improvement of the capacity performances within sectors. These capacity stress-test will be done by increasing the number N of flights entering the sector in a certain time window and investigating up to which value of N the algorithm is able to find a solution with trajectories conflict free. We believe that investigating capacity issue in connection with the optimal generation of conflict free trajectories is a relevant issue, especially in view of the new SESAR scenario.

Moreover, the modular structure by which the model has been implemented will allow us to easily switch from the current to the future SESAR scenario by eliminating some module, such as the Navigation points module and/or changing the order by which some tasks are executed, i.e. some strategies are adopted.

2.3.7 Input from Validation experts

The sector chosen for the calibration of the tactical layer was the sector LIRROV in the Rome ACC, see Fig. 2.6. An operational expert working at the Rome ACC has been interviewed regarding the

features of such sector and most of the operative knowledge acquired has been integrated into the tactical layer.

The sector is crossed by North-South and East-West overflight traffic, as presented in Fig. 2.6. The flights inside the sector are mainly commercial, with few exceptions of military flights that however behave like commercial flights. As a result of these traffic pattern, several critical areas emerge from the crossing of these traffic flows, highlighted by red circles in Fig. 2.6.

The sector can operate in two main configurations summarized in Table 2.1. During summer the traffic load is usually high so the sector operates with the configuration “B” in which it is vertically split in order to increase its capacity. On the other hand, during winter when the traffic is lower, it usually operates in configuration “A” where it is composed by just one volume and lesser capacity.

The strategies used for conflict resolution are considering both horizontal and vertical actions. In the first case one of the aircraft involved in a possible conflict is deviated from its original route to achieve horizontal separation, while in the other case a small variation in flight level of just 10 FL is used to achieve vertical separation. Combinations of these two strategies are also possible. Despite the fact that horizontal deviations are more convenient in terms of fuel consumption, a small vertical deviation is usually preferred. Moreover in order to reduce the amount of traffic to be managed and the delay generated by their action, controllers usually send aircraft directly to the exit point of the sector after any deviation and whenever possible. Another possibility for the controllers to reduce the traffic load of the sector is to apply a “direct”, i.e. to send an aircraft directly to a point in the next sector of its flight plan. However, since directs require the coordination of the controllers in the involved sectors and thus an increase of their workload, they are considered unlikely events. The tactical layer implements all these finding regarding the traffic patterns inside the sector. Main routes and critical areas are reproduced and also the seasonality of traffic has been considered in the calibration.

Adverse weather conditions occur on a daily basis and are not a negligible effect inside the sector and the system in general. This kind of events does not represent a challenge for the controllers that are always supposed to be capable of handling them. We have been able to identify two major classes of perturbations depending on their dimension:

- Small shocks ($\approx 5 \text{ NM}$ of radius), with a fast dynamics and a short lifetime ($\approx 1 \text{ h}$), usually occurring during summertime.
- Large shocks (around $60 \text{ NM} \times 20 \text{ NM}$), which can be considered static and with a lifetime that goes from 8 h to 10 h . This kind of perturbations represents big storms occurring during winter.

While the only possible way to manage a small shock is to avoid it, it is possible that the biggest one could be crossed by an aircraft instead of being avoided and thus generating a small delay instead of a large one. These kinds of shocks have been implemented into the tactical layer, following the discussion with the operational expert. Correlations between size, lifetime and dynamics of the events have been introduced, so that both small dynamical shocks and large static ones are investigated. In both cases shocks can be avoided using horizontal deviations as well as vertical ones if the considered shock does not affect all the flight levels. Since aircraft might fly through a large perturbation instead of completely avoiding it, a probability of being crossed is assigned to each large perturbation. Moreover in order to simulate correctly the seasonality of the disturbances, small shocks are more likely to occur when simulating summertime while large shocks occur more frequently when simulating winter time. After any redirection, due to separation or adverse weather conditions, controllers try to sent the aircraft directly towards its exit point in the sector.

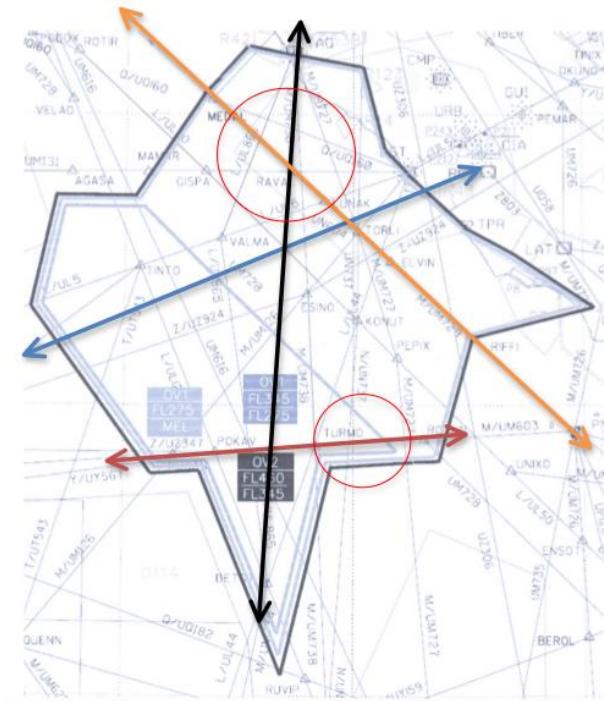


Figure 2.6: Projection of the LIRROV sector on a map including the navigation points. Major routes and their directions are indicated by the blue, red, orange and black arrows while the critical areas emerging from their intersection are marked with red circles.

Configuration	Sector/s	Heights	Capacity
A	<i>OV1 + OV2</i>	<i>MEL – 460 FL</i>	52
B	<i>OV2</i>	350 + FL	44
B	<i>OV1</i>	<i>MEL – 345 FL</i>	44

Table 2.1: Operation configurations of sector LIRROV. In configuration A the two sub-sectors OV1 and OV2 are joined. In configuration B the two sub-sectors OV1 and OV2 are kept distinct although they are opened simultaneously.

2.3.8 Calibration of the model

The main issues we have considered to calibrate the model are:

- the aircraft velocity.
- the angle α of deviation from the planned trajectory when rerouting occurs.
- the probability p by which directs are issued.

The input to our model has been given by all flights of the LIRROV sector in day 6 May 2010, fulfilling the following constraints: (i) each navigation point of the route inside the sector is higher than at least 240 flv and (ii) the trajectory of the aircraft intersects the sector in two distinct point. The number of the flight trajectories fulfilling such constraints is: N=452 for day 06 May 2010.

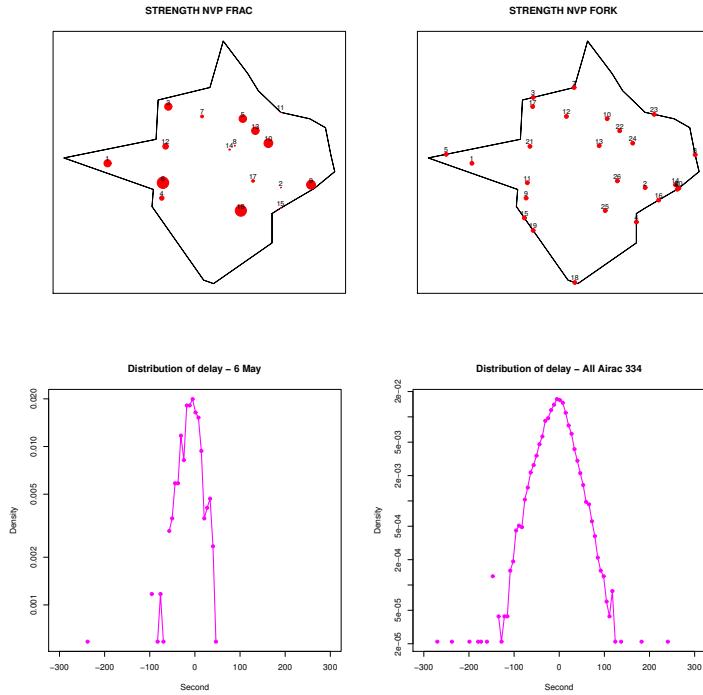


Figure 2.7: Main metrics used to calibrate the model: the frac (top-left panel), fork (top-right panel), average delay of the considered trajectories on 6 May 2010 (bottom-left panel) and average delay of the considered trajectories on the whole AIRAC (bottom-right panel).

The analysis of the simulated trajectories has been done on the smaller subset of $N_a = 380$ aircraft fulfilling the STRONG filter prescriptions stated in deliverable D1.3. This filter involves all flights: 1) with Landplanes, 2) scheduled, 3) having a IATA code, excluding flights having a duration shorter than 10 minutes. In the selected day 6 May 2010 we have 14 navigation points inside the sector and 11 on the boundaries, see Fig. 2.7 below.

In order to calibrate the above parameters we mainly consider the following metrics: **frac** (top-left panel of Fig. 2.7), **fork** (top-right panel of Fig. 2.7), delay computed in the same day of the considered M1 flight trajectories (bottom-left panel of Fig. 2.7) and delay computed over all AIRAC (bottom-right panel of Fig. 2.7). Frac and forks were defined in deliverable D1.3. We recall here the definitions for the reader's convenience: **frac** is the fraction of flight trajectories actually passing over a given navigation point (M3 trajectories), with respect to the planned case (M1 trajectories); **fork** is the fraction of flights for which a deviation begins after this point; this point is the last common point in M1 and M3 for this piece of trajectory.

In the two top panels of Fig. 2.7 we show the geographical location of the real navigation points obtained from the real M1 flight trajectories of day 6 May 2010. The size of the circle is proportional to the number of frac (top-left panel of Fig. 2.7) and fork (top-right panel of Fig. 2.7), respectively.

The delay is defined as the difference of time used by the aircraft to cross the sector (M3) minus the planned one (M1).

Frac, forks and delay obviously are different in different days of the AIRAC. In order to check the variability of fracs, we computed the correlation matrix (unweighted) between the fracs of two different days of the AIRAC 334. The correlation was computed by considering the configuration

of values of the frac for each navigation point inside the sector in the selected day as a vector of 14 elements (which is the number of navigation point inside the sector) and by computing the unweighted Pearson correlation coefficient between the vectors of different days. Looking at the distribution of all correlation coefficients (left panel of Fig. 2.8) we can observe that high value of correlation are reached (mean 0.088 std 0.352). The central panel of Fig. 2.8 shows a contour plot of the correlation between any pair of days in AIRAC 334. The white color is assigned when the absolute value of the correlation coefficient is smaller than $1/\sqrt{14}$. The panel shows that days with similar fracs geographic profile can be observed. In the right panel of Fig. 2.8 we have ordered days according to the results of a hierarchical clustering procedure applied to the correlation matrix. In this case blocks of days where fracs behavior is similar can be observed around the main diagonal.

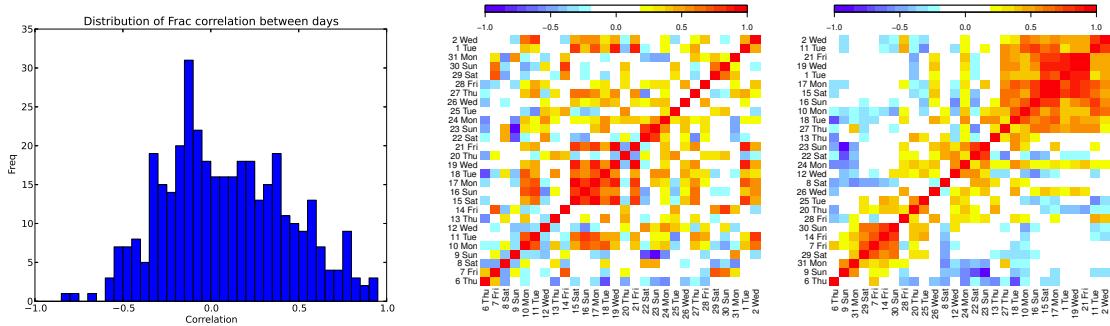


Figure 2.8: Correlation between fracs profile observed in the different days of AIRAC 334. The left panel shows the distribution of all correlation coefficients. The central panel shows a contour plot of the correlation between any pair of days in AIRAC 334. The white color is assigned when the absolute value of the correlation coefficient is smaller than $1/\sqrt{14}$. The right panel shows a contour plot of the correlation coefficients after we have ordered days according to the results of a hierarchical clustering procedure applied to the different days of the correlation matrix.

2.3.8.1 Aircraft velocity

The analysis of real data (AIRAC 334 - Sector LIRROV) shows a relevant difference between the planned velocity of aircraft and the actual one.

In order to investigate this aspect, we consider all trajectory segments¹ inside the sector LIRROV and calculate the percentage difference of velocity between M3 and M1 trajectories, i.e. $v_r = \frac{V_{M3} - V_{M1}}{V_{M1}}$. As shown in Fig. (2.9), the distribution of v_r shows exponential tails. This has been estimated by using the Kolmogorov-Smornov test. In fact we compute the K-S statistics for all subsets $S_x = \{x_i : x_i > x_{min}\}$ obtained by considering different values of x_{min} . We then choose the first value of x_{min} with a p-value larger than 0.01, where the p-value was obtained by adapting the methodology illustrated in Ref. [14]².

¹Pairs of navigation points of the routes $\text{navigationpoint}_i - \text{navigationpoint}_{i+1}$

²<http://cran.r-project.org/web/packages/poweRlaw/vignettes/poweRlaw.pdf>

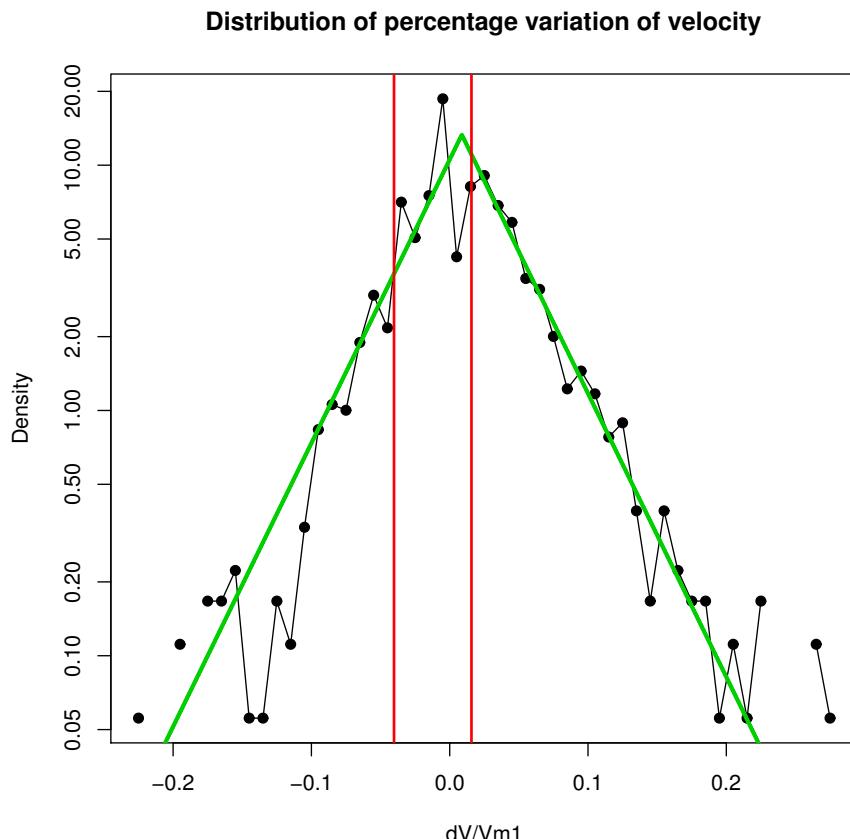


Figure 2.9: Distribution of percentage difference of velocity (M3,M1). The two vertical red lines indicate the best x_{min} for the two tails, selected by using the procedure sketched above.

We therefore decided to include in the ABM this inaccuracy on the estimated velocity. Actually, for each aircraft, we modify the planned velocity in each segment according to a laplacian distribution with the same parameters observed in the real one. As a result the distribution of delays shows a good agreement with real data, as shown in Fig. 2.10. Also in this case we performed the K-S test obtaining $K - S = 0.0422$ and p-value $p = 0.4131$. It is worth noticing that the main differences between model and real data are in the left tail of the distribution.

We investigated the relation between incorrect estimation and aircraft models. In fact, for the whole AIRAC 334, we computed the delay acquired in LIRROV by all aircraft fulfilling the STRONG filter mentioned above. We partitioned all these flights into 46 categories, according to the type of aircraft used in the specific flight. We calculated the distribution of delay for each of 46 partitions of different aircraft models. In Fig.2.11 we show the average and standard deviation of these delays.

For each of the 1035 possible pair of aircraft type we calculated the Kolmogorov-Smirnov test, and only 36 pair of aircraft type resulted to have the same distribution of delay ($p-value > 0.01$ $K - S < 0.1$). Therefore in general different aircraft types have different delay distributions. To compare this value with a random null hypothesis we performed a bootstrap (without repetitions) taking into account that we have an heterogeneous system in term of partition size, i.e. the number of flights with a certain aircraft type. This was made by a random resampling of the partitions preserving the

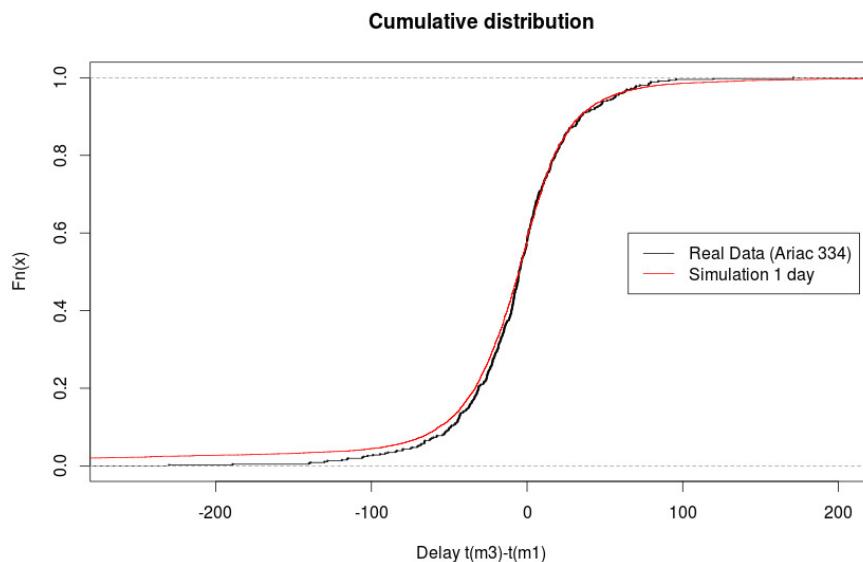


Figure 2.10: Cumulative distribution of delay in the simulated and real data when the distribution of velocities in each trajectory segment is calibrated with velocities maintained by aircraft obtained from real M3 data.

same size and extracting value from the entire sample. We performed 3000 bootstrap replica and for each one we calculated the number of distribution that K-S test recognize as identical. From this analysis we obtained 103 average number of pairs with identical distribution, with standard deviation 13.5. Therefore the observed value of 36 pairs observed in real data is statistically significant. We can thus conclude that there is a relation between aircraft model and incorrect estimation of velocity.

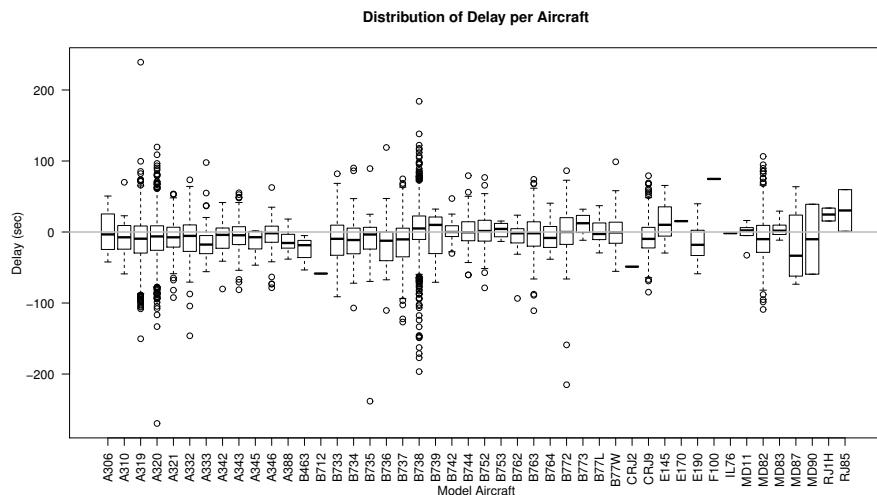


Figure 2.11: Box-plot of delay in real data for different aircraft type. The shown error bars are the standard deviations. The boxes correspond to the first and third quantile.

2.3.8.2 Deviation angle α

As mentioned above we performed simulations to select the best value of the maximum angle α_M of deviation between trajectories when vectoring occurs.

In order to calibrate the model, we used a set of angles $[5^\circ, 10^\circ, 20^\circ, 30^\circ]$ and for each simulation we compared the value of frac for the navigation points inside the sector with real data. We first tried to compare the results of our simulation with the average value of frac in all AIRAC 334, see bottom-left panel of Fig. 2.12. We then compared the results of our simulation with the frac observed in the same day of the simulation, see bottom-left panel of Fig. 2.13. In both cases, for each navigation point we computed the Pearson correlation coefficient between the frac measured in the simulations, see the top-left panel of Fig. 2.8. We also considered the weighted correlation, where the weight of each navigation point was given by the number of aircraft that cross the selected navigation point, see top-right panels in Fig. 2.12 and Fig. 2.13.

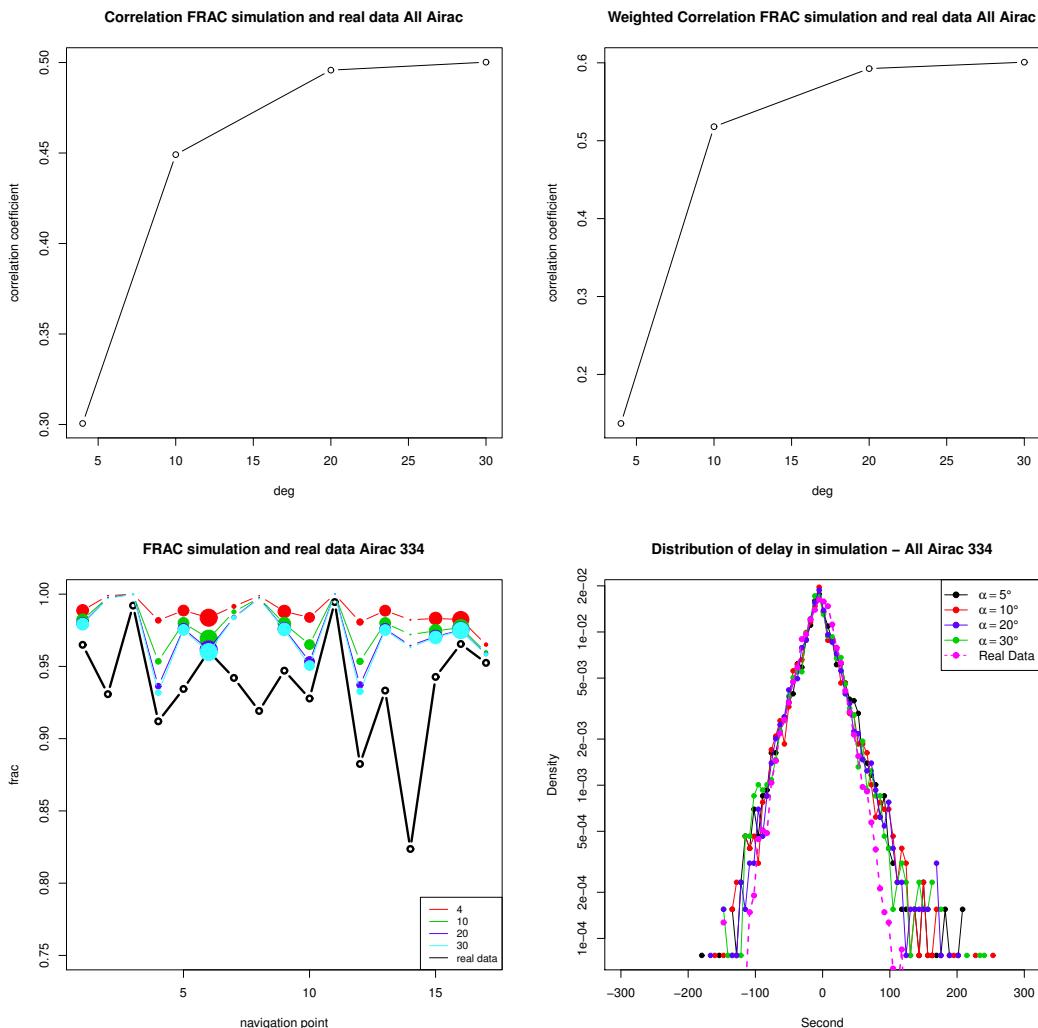


Figure 2.12: Simulations with different maximum angle compared with real data relative to the whole AIRAC 334. In the bottom-left panel we show fracs and in the bottom-right panel we show the pdf of delays for different values of α . In the top panels we show the Pearson correlation coefficient (left panel) between the frac measured in the simulations and the weighted correlation coefficient (right panel), were the weight of each navigation point was given by the the number of aircraft that cross the selected navigation point.

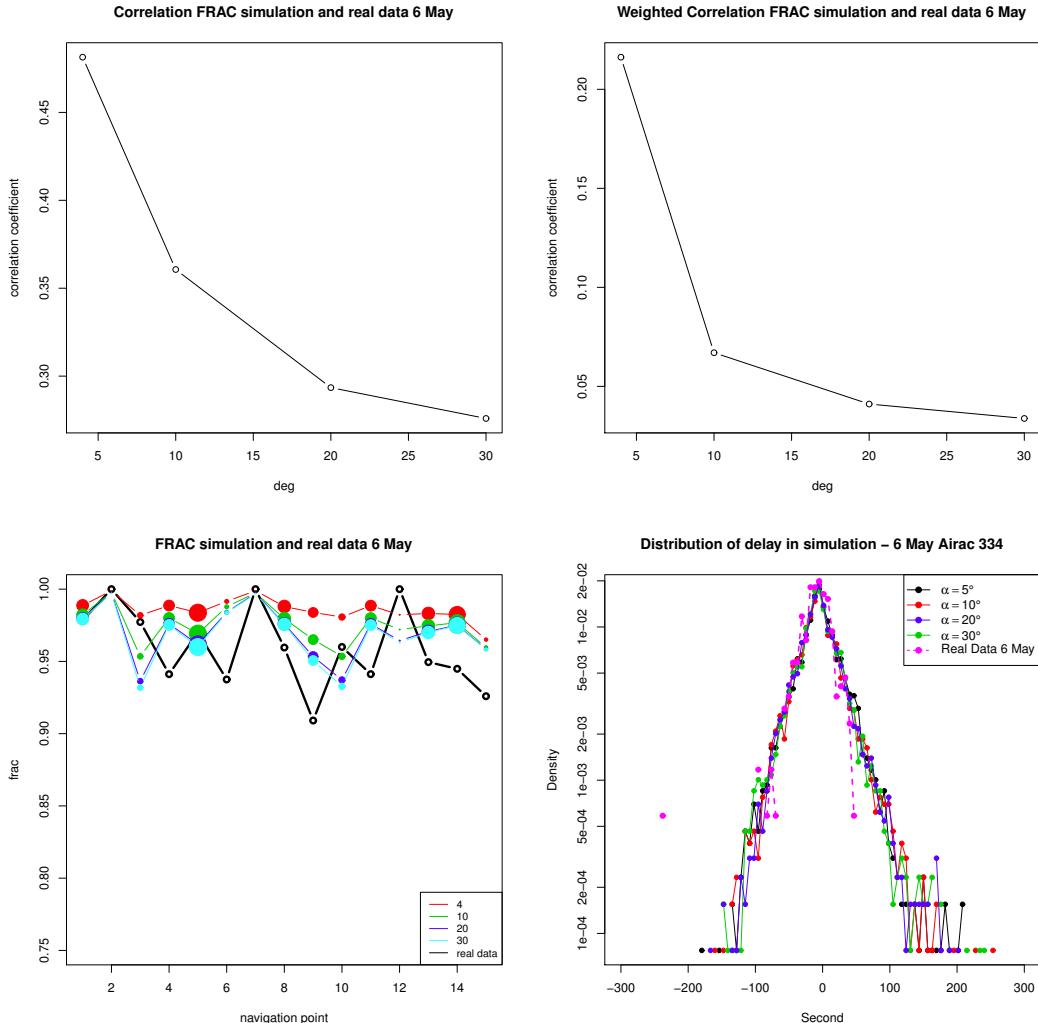


Figure 2.13: Simulations with different maximum angle compared with real data relative to the same day of the simulation. In the bottom-left panel we show fracs and in the bottom-right panel we show the pdf of delays for different values of α . In the top panels we show the Pearson correlation coefficient (left panel) between the frac measured in the simulations and the weighted correlation coefficient (right panel), were the weight of each navigation point was given by the the number of aircraft that cross the selected navigation point.

In the bottom-right panel of the two figures we report the pdf of delays for different values of α . In all cases the distribution shows exponential tails as in the real data. The mean and standard deviation of delays is reported in Table 2.2. The results obtained by using the model are slightly different from those obtained for real data. However, in both cases we have negative delay, meaning that the model is able to generate surrogate trajectories that give an overall anticipation of time at which the aircraft leaves the sector, as in the real case. It is worth noticing that data from the simulations are more similar to those obtained for the whole AIRAC, rather than those relative to the single day. We believe that this is due to the fact that there might be specificities of the considered

day, obviously not captured by the model, that averaged out when considering the whole AIRAC.

data	mean	std
05°	-3.54	36.1
10°	-3.55	36.1
20°	-3.49	36.2
30°	-3.49	36.2
All Airac	-6.05	31.0
6 May	-11.29	27.8

Table 2.2: Average delay in real data and simulations for different values of α . Here the directs module is switched off.

Based on the above results we selected the value $\alpha_M = 20^\circ$ as the maximum angle of deviation between trajectories when vectoring occur.

2.3.8.3 Directs probability

As mentioned above we performed simulations to select the best value of the probability p by which directs are issued.

In order to calibrate the model with respect to the probability by which directs are issued, we used a set of possible probabilities in the range $[0, 0.1]$ and for each simulation we compared the value of frac for the navigation points inside the sector and the delay with real data, as shown in Fig. 2.14. In all cases the delay distribution shows exponential tails as in the real data.

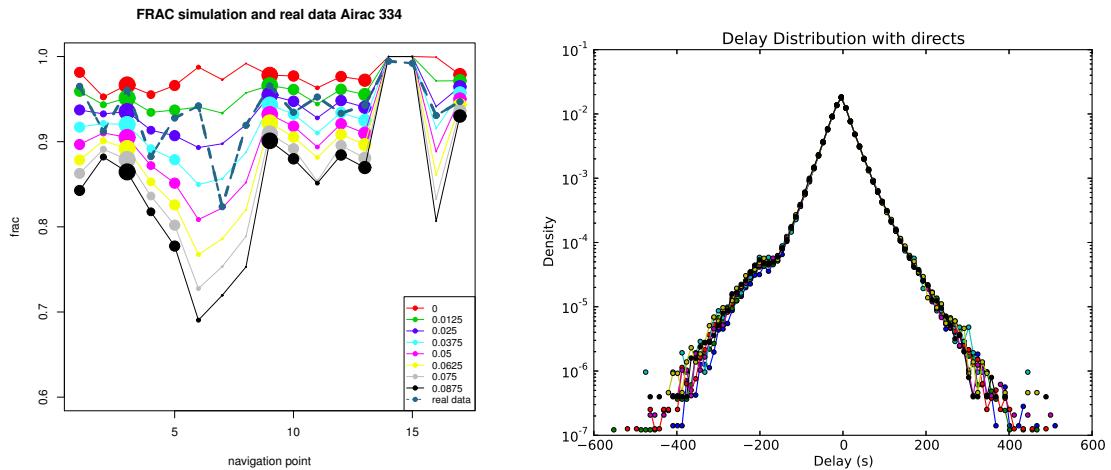


Figure 2.14: Simulations with different values of the probability p by which directs are issued. In the left panel we show fracs and in the right panel we show the pdf of delays for different values of p .

The mean and standard deviation of delays is reported in Table 2.3. The results obtained by using the model are very similar than those obtained for real data, thus improving the results of Table 2.2. Also in this case data from the simulations are more similar to those obtained for the whole AIRAC, rather than those relative to the single day.

data	mean	std
0.0125	-4.73	38.1
0.0250	-5.06	38.6
0.0375	-5.36	39.0
0.0500	-5.54	39.1
0.0625	-5.42	39.1
0.0750	-5.54	39.5
0.0875	-5.60	39.3
All Airac	-6.05	31.0
6 May	-11.3	27.8

Table 2.3: Average delay in real data and simulations for different values of p and $\alpha = 20^\circ$.

Based on the above results we selected the value $p = 0.10$ for the probability by which directs are issued.

2.4 Results

The Agent Based Model illustrated above has been therefore used to perform simulations of the current scenario to investigate how the presence of shocked areas affects the performance of the system. We have simulated the case when areas of the sector are closed to air traffic due, for example, to adverse weather events, or military manouvers, etc.

In order to determine the spatial localization of shocked areas, we fix the center C of the sector and we identity the maximum circle centered in C and fully contained in the sector. Let us call R its radius. We assign the center of the shocked area by assigning its polar coordinates from C , i.e. the distance r from C and an angle θ measured from some reference line. The angle θ is always extracted from a random uniform distribution. We have then the three following possibilities:

- **UNIFORM** - The distance r is obtained extracting two numbers x,y from a random uniform distribution in the range $[-R, R]$ and accepting the value $r = \sqrt{x^2 + y^2} < R$.
- **GAUSSIAN** - The distance r is extracted from a normal (gaussian) distribution in the range $[0, R]$. The distribution is centered in zero. We consider only the right side of the distribution. The standard deviation of the distribution can assume the following values: $\sigma = 10, 15, 20, 25, 30, 35, 40, 45$. This is done for each value of n .
- **POWER-LAW** - The distance r is extracted from a power-law distribution $\mathcal{P} = Nr^{-\gamma}$ for $r \in [x_{min}, R]$ with $x_{min} = 0.2$ NM. The γ parameter can assume the following values: $\gamma = 0.01, 0.05, 0.1, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8$. This is done for each value of n .

We show below a few results obtained by using our model. The selection of the results we want to present here was done with the aim of showing how by changing some of the ingredients of the model we can affect its general performances and to illustrate the main mechanisms of the model.

2.4.1 Poissonian temporal occurrence of shocked area

A first set of simulations has been obtained by assuming that shocked areas are randomly localized over all flight levels and in spatial positions that are extracted from (i) a random uniform distribution, (ii) a gaussian distribution and (iii) a power-law distribution. In all these cases the shocked area is circular with a radius of 5 NM, see section 2.3.7. Other radius values are considered in Appendix E.

The total area of the sector is about 56747 km^2 . Therefore a shocked area of radius 5 NM corresponds to 0.5% of the sector planar extension. The value of R in real data is about $R = 14 \text{ km}$. The duration of each shock is equal to the size of the time-step Δt . The time at which each shocked area occurs is extracted from a random uniform distribution with an average number n of shocked areas per time-step. This results in a Poisson distribution of the temporal occurrences of the shocks. The considered values of n are: $n = 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5$. The directs module is switched off. The velocity module is switched off. The only possible actions are vectoring and flight level change. For each set of parameters we perform 2000 simulations. Each set of 2000 simulations lasts from 2 to 5 hours, depending on the value of n .

In the top panels of Fig. 2.15 we show the average (top-left panel) and the standard deviation (top-right panel) of delay for the three considered distributions. In the bottom panels of Fig. 2.15 we show the average number of reroutings and flight level changes for the three distributions considered. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ . In order to make a meaningful comparison, on the horizontal axis we show the average size of the filled surface for each set of parameters. The average filled surface per time-step per flight-level has been computed by using the python library called `shapely.geometry`. The results of Fig. 2.15 show that the largest delay, number of vectorings and flight level changes are obtained when shocks' localizations are extracted from a uniform distribution. We interpret this result by considering that when shocks are uniformly distributed there is more room for performing trajectory variations and therefore the model tries many more vectorings and flight level changes. This results in an overall lower negative delay, with respect to the other two distributions, see top-left panel. In the case of gaussian and power-law distribution we are assuming that shocks are localized. Therefore there is less room for performing trajectory variations. Accordingly, the number of vectorings and flight level changes is smaller than the previous case and this results in an overall increase of delays. However, there are differences between the gaussian and power-law case. In fact, the delay obtained with the gaussian is higher than the one obtained for the power-law case and consistently we have lower vectorings and flight level changes. This is so because in the case of gaussian distribution the shocks are more localized than in the case of power-law, due to the fact that power-law distributions have fatter tails.

To better explain this point we have introduced a proxy of porosity of the sector, i.e. of the available space within the sector when shocks occur. Porosity is here defined as the ratio a between the surface *not occupied* by shocks and the surface of the sector. In Fig. 2.16 we show the radial porosity [15] for the three different considered distributions. In fact, we have considered 10 values of radius $r_i \in [0, R], i = 1, \dots, 10$ and we have computed the porosity a in the annulus identified by two radii r_i and r_{i+1} , i.e. in the the region bounded by two concentric circles of radius r_i and r_{i+1} . The different lines in the three panels refers to the different values of n considered. The right panel is relative to the gaussian distribution and shows that porosity approached unity (i.e. full space available for aircraft) earlier than in the power-law case (central panel). This means that for small values of r the available space is smaller in the gaussian case with respect to the power-law case, thus indicating that shocks are more concentrated in the central region of the sector. Interestingly, for the uniform distribution, porosity stay well below unity in all considered values of n .

For each of the three considered distributions, in Table 2.4 we show the total number of actions that have been done on all the flight trajectories, either vectoring or flight level change. We can interpret this numbers as a proxy of controllers workload. Data are ordered as Fig. 2.15, i.e. the third entry in the first column of the Table corresponds to the third point of the green curve in the bottom panels of the figure. Each entry of the Table corresponds to the sum of the values relative to the corresponding points in the bottom panels of the figure. We can see that usually flight level changes are issued in the model more frequently than vectorings. Moreover, we can see that for

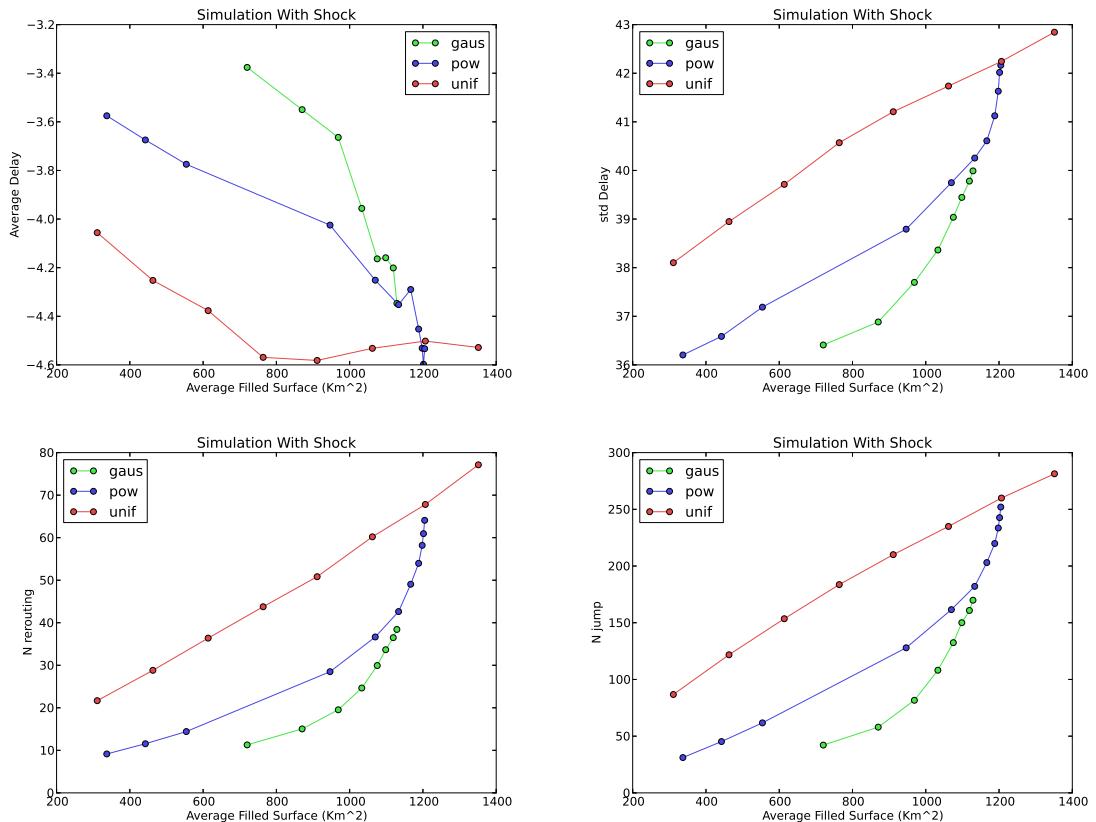


Figure 2.15: Shocked areas of radius 5 NM. In the top panels we show the average and the standard deviation of delay for the three considered distributions. In the bottom panels we show the average number of reroutings and flight level changes for the three distributions considered. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ . The directs module is switched off.

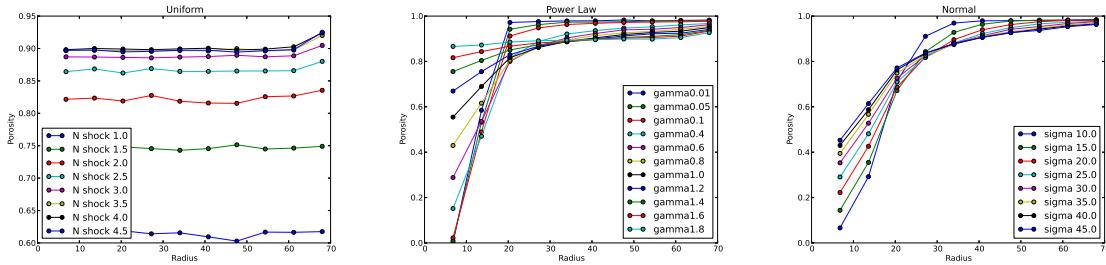


Figure 2.16: Shocked areas of radius 5 NM. The Fig. shows the radial porosity [15] for the three different considered distributions: uniform in the left panel, power-law in the central panel and gaussian in the right panel. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ .

a given value of filled surface in the sector the workload proxy is usually higher when shocks are distributed (uniform distribution) rather than concentrated (gaussian distribution).

Gaussian filled surface	Gaussian actions	Power Law filled surface	Power Law actions	Uniform filled surface	Uniform actions
720.2	53.4	337.0	40.3	311.0	108.4
870.3	73.0	442.4	56.8	462.9	150.6
968.9	101.2	554.1	76.1	613.8	189.9
1033.1	132.8	946.5	156.4	763.9	227.4
1075.2	162.4	1069.9	198.2	911.2	260.9
1098.4	183.7	1133.6	224.7	1062.0	295.1
1119.1	197.3	1166.6	252.1	1206.6	327.8
1129.0	208.3	1188.3	273.8	1351.3	358.4
		1197.9	291.7		
		1201.5	303.6		
		1204.8	316.1		

Table 2.4: Shocked areas of radius 5 NM. The entries of the second, fourth and sixth column in the Table are the total number of actions that have been done on all the flight trajectories, either vectoring or flight level change, for the three different considered distributions. These entries of the Table are a proxy of controllers workload. The entries in the first, third and fifth columns are the filled surface corresponding to the different parameters used in the simulations. The directs module is switched off.

Each flight trajectory can be managed by the controller more than once. Therefore the numbers shown in the bottom panels of Fig. 2.15 and in Table 2.4 do not correspond to the number of re-routed individual aircraft or to the number of individual aircraft that changed flight level. In fact, in Table 2.5, for the uniform distribution, we show the number of aircraft that have anyway changed at least once their trajectory either horizontally or vertically. For each of the 2000 simulations we compute the averages over all 452 aircraft and then we compute the average (first and third column) and standard deviation (second and fourth column) of these averages over all 2000 simulations. The same data are shown for the power-law distribution in Table 2.6 and in Table 2.7 for the gaussian distribution. When we consider the filled surface associated to the different parameters considered,

we see that the number of re-routed aircraft and the number of aircraft that changed flight level is smaller in the case of gaussian distribution and larger in the case of uniform distribution, with the power-law distribution being an intermediate case. For example, for a filled surface values around 1100km^2 , we get an average number of vectorings equal to 192.6 ± 10.8 for the uniform distribution, a value in the range $[137.0 \pm 9.39, 152 \pm 9.7]$ for the power-law distribution and 127.3 ± 9.36 for the gaussian distribution. These results indicate that when we have relatively small shocked areas (about 5 NM), the larger the concentration (gaussian case) the smaller number of vectorings. Therefore we have less optimized trajectories, which results in a larger delay, as also shown in Fig. 2.15.

parameter n	filled area	average vectoring	std vectoring	average level change	std level change
1.0	311.0	78.4	8.42	24.1	4.59
1.5	462.9	107.8	9.36	32.0	5.40
2.0	613.8	133.0	9.93	39.8	5.867
2.5	763.9	156.1	10.2	47.4	6.43
3.0	911.2	175.4	10.4	54.8	6.76
3.5	1062.0	192.6	10.8	63.5	7.21
4.0	1206.6	209.6	10.2	71.0	7.59
4.5	1351.3	223.7	10.3	79.5	7.87

Table 2.5: Shocked areas of radius 5 NM. The entries of the Table are the average number of aircraft that are re-routed or change their flight level, for the uniform distribution. The directs module is switched off.

parameter γ	filled area	average vectoring	std vectoring	average level change	std level change
0.01	337.0	28.1	3.84	9.91	3.08
0.05	442.4	39.9	5.15	12.5	3.37
0.10	554.1	54.1	6.22	15.8	3.86
0.40	946.5	109.9	9.32	31.3	5.32
0.60	1069.9	137.0	9.39	40.1	5.72
0.80	1133.6	152.9	9.87	46.1	6.42
1.00	1166.6	169.0	10.1	52.4	6.62
1.20	1188.3	181.3	10.3	57.4	7.08
1.40	1197.9	191.4	10.3	62.0	7.19
1.60	1201.5	198.0	10.4	64.8	7.33
1.80	1204.8	204.1	10.6	67.9	7.41

Table 2.6: Shocked areas of radius 5 NM. The entries of the Table are the total number of aircraft that are re-routed or change their flight level, for the power-law distribution. The directs module is switched off. Data are obtained for $n = 4$.

parameter σ	filled area	average vectoring	std vectoring	average level change	std level change
10	720.2	35.1	4.15	12.0	3.324
15	870.3	48.6	5.32	16.2	3.73
20	968.9	69.0	6.82	21.3	4.394
25	1033.1	92.4	7.993	27.1	4.72
30	1075.2	112.9	8.53	32.9	5.31
35	1098.4	127.3	9.36	36.9	5.72
40	1119.1	135.8	9.43	39.9	5.72
45	1129.0	143.0	9.79	42.0	5.93

Table 2.7: Shocked areas of radius 5 NM. The entries of the Table are the total number of aircraft that are re-routed or change their flight level, for the gaussian distribution. The directs module is switched off. Data are obtained for $n = 4$.

In Fig. 2.17 we show the frac metric for the three considered distributions. Fracs are compared with the empirical fracs observed in day 06 May 2010 (top panels) and for the whole AIRAC (bottom panels). Fracs obtained by using the gaussian distribution are always above 0.90 which is the typical value of fracs obtained when averaging over all the AIRAC. In the case of power-law distribution we get values below 0.90, with the only possible exception of the magenta and orange curve which are obtained with $\gamma = 1.6$ and $\gamma = 1.8$, respectively. This would suggest that real data are more compatible with a concentrated distribution of shocked areas rather than with a sparser one.

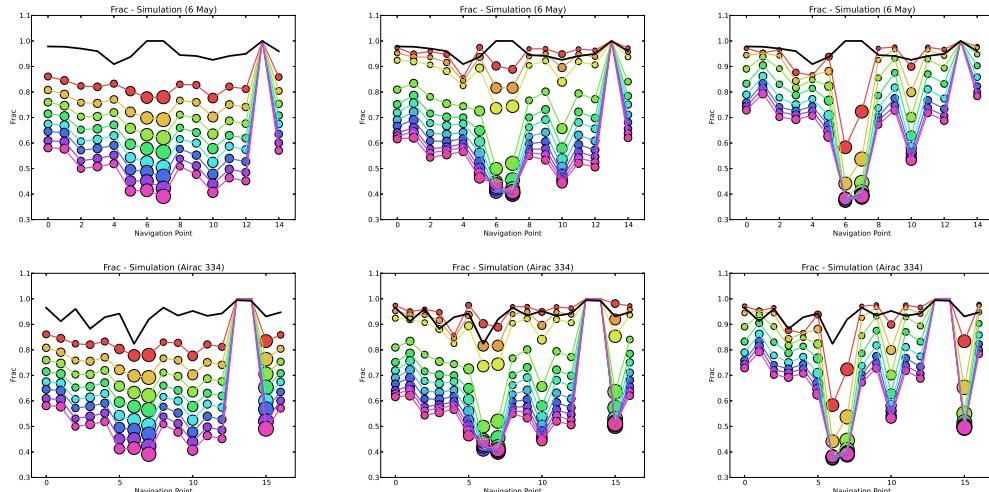


Figure 2.17: Shocked areas of radius 5 NM. The Fig. shows the frac metric for the three different considered distributions. In the top panel simulated data are compared with fracs (black curve) relative to 06 May 2010. In the bottom panels data are compared with fracs measured over all the AIRAC 334. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. The two left panels refer to the uniform distribution. The two central panels refer to the power-law distribution and the two right panels refer to the gaussian distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ . The directs module is switched off.

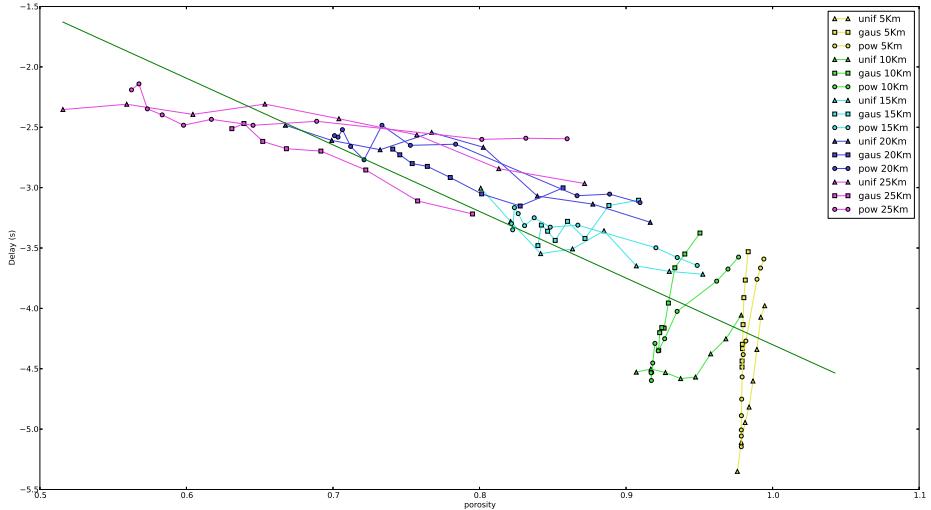


Figure 2.18: Average delay against porosity, for all simulations we have performed so far, i.e. with different values of n , σ or γ and different values of the extension of the shocked area, see appendix E.

In Fig. 2.18 we show the average delay against the porosity, for all simulations we have performed so far, i.e. with different values of n , σ or γ and different values of the extension of the shocked area, see appendix E. In the Fig. the different simulations detailed in the legend are shown with different colors and symbols. We also show a reference straight line. This Fig. seems to suggest that the larger the porosity the smaller the delay (the greater the anticipation), which seems to be the key result of our model.

2.4.2 Temporal occurrence localized in specific time intervals

In the last set of simulations we used again an uniform distribution for the geographical localization of the shocks but we temporally distributed them in a short time window of three hours. We then moved this window over the 24 hours of the considered day.

We tested how the system reacts when the shocking areas are temporally concentrated. In fact, the number of active aircraft changes across the day-time and it has two peaks: one around 8 AM and one around 9 PM.

It is obvious to assume that the predictability of the system decreases when the shocks are distributed during the congested time in the sector.

However in contrast with our expectation the higher negative delays are reached when the shocks are concentrated, i.e. when the sector is congested, see Fig.2.19. This effect is probably related to the rerouting strategy implemented in the model. In our model the controller uses to give directs to solve conflicts. The effect of directs is expected to be relevant because real flight trajectories are not optimized in this respect³. However, this effect is not simple to compare with real data because the ABM was implemented with rules that do not perfectly match the real behavior of the ATC. In fact

³the situation might be different in the new SESAR scenario with the introduction of business trajectories

the air traffic controllers usually give directs across sectors, while in our model the aircraft have to enter and exit from the same point of the sector.

This effect on delay would suggest that an increment of the ATC operations might improve the efficiency of the system even when available sector space is reduced and the traffic congestion increases.

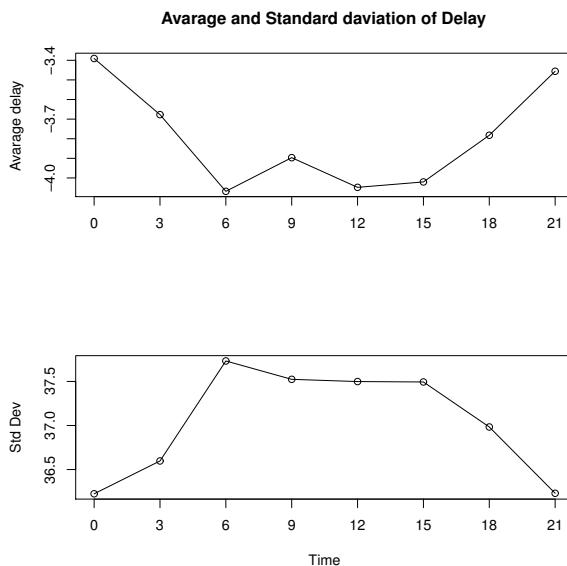


Figure 2.19: The Fig. shows the average (top panel) and standard deviation (bottom panel) delay for the uniform distribution when shocks are temporally concentrated in a time widows of 3 hours.

This result parallels the one of section 2.4.1. We can perturb the system in two different ways: either increasing the temporal concentration of shocks or by increasing their spatial concentration. In both cases, the model is able to improve the efficiency of the system by generating trajectories with overall negative delays.

2.4.3 Conclusions

The main results obtained with our model is that, counter-intuitively, the larger the porosity the smaller the delay (the greater the anticipation) we observe in the simulated trajectories. As also indicated in appendix E, this result seems to revert when we start considering shocked areas with larger radius.

A clear-cut interpretation of this counter-intuitive result is not straightforward, also because we do not know whether in the real data considered for the calibration of the model there were extreme weather events or areas of the sector closed to air traffic for military reasons or the like. In any case, we believe that these results well indicate how our model works. For relatively small shocked areas the model optimizes the flight trajectories giving rise to overall negative delays. This effect is also present when we do not have shocked areas and the model is therefore simply making trajectories conflict free. Whenever the model modifies a flight trajectory, it is therefore able to provide an optimization in terms of delay. Small shocks uniformly distributed leave more room for optimization rather than small concentrated shocks. We must however admit, that this optimization might be

done at the cost of changing the flight trajectory many many times, see Table 2.4, which might not be acceptable for airlines. However, we do not have data relative to the controllers workload to compare with. On the other hand, when individual shocks are large ($> 15 \text{ km}^2$) the model capacity to optimize trajectories degrades, see Appendix E. In this case, large shock uniformly distributed leave less room for optimization rather than large concentrated shocks. In fact, large concentrated shocks tend to overlap with each other and therefore to increase the number of shocks only marginally reduces the available space for aircraft, in this case.

Having stated that there are two possibilities: (i) either our model is badly calibrated due to lack of empirical data, or (ii) the result obtained for relatively small shocked areas (radius below 10 NM) indicate that the automated air traffic management strategies proposed in the model might even outperform the human based strategies, at least as far as delays are concerned.

We believe that the present version of the model is a good starting point for performing scenario simulations able to investigate the new SESAR scenario of the ATM world. To reach this goal, we are aware that there are improvements that need to be implemented in the model:

- The calibration and validation activities of the current version of the model we have performed so far would need to be further extended, in the direction of better comparing the simulations results with real data in terms of *controllers workload* and *weather events*. This would necessarily need to have access to a new type of data;
- One key issue in the future SESAR scenario is that ATM will move from a network-based (routes and navigation points) to a trajectory-based scenario. The other key issue is the implementation of larger sectors. This would in principle allow to increase the overall capacity of the system of accommodating a larger number of aircraft. One preliminary issue is therefore that of understanding to which extent the capacity in the current scenario can be still increased by simply optimizing the current (network-based) management procedures. In fact, we want to perform capacity stress-tests by trying to push the capacity of a set of N sectors to their limits and comparing the results, in terms of average delay and number of actions to be performed by the controller. We will then compare these results with those obtained by considering one larger sector comprising the N smaller ones. This will allow us to assess how the implementation of larger sectors will structurally change the ability of the system to manage a larger number of aircraft. We will therefore be able to disentangle the effect due to the implementation of larger sectors with respect to the effect due to the implementation of the trajectory-based scenario.
- The current version of the model models the processes that take place at the level of a single sector. We believe that a multi sector extension of the model is necessary if we want to investigate issues related to the delay propagation across sectors. This would be also relevant in view of the new SESAR scenario. Will the new SESAR scenario be beneficial for reducing such delays? To which extend the introduction of larger sectors will help in this direction?
- The implementation of the multi-sector extension will also give us the possibility of considering different typologies of sectors, characterized by different types of traffic. Examples include sectors where trajectories are almost always parallel to each other, sectors where trajectories mainly intersect with each other, sectors which have a small (large) number of confining sectors. We will then perform a calibration activity trying to use standard complexity indicators of sectors to characterize them and then we will perform simulations trying to investigate how delays propagate from sector to sector depending on the typology of traffic in the considered sectors.

In summary, we think that two types of improvements are needed. On one side, even at the level of a single sector, we need improvements on the understanding of controllers workload, the

characterization of weather events and capacity stress-tests. More generally, we believe that it will be necessary to implement a multi-sector extension of the tactical layer of the model. This needs to be firstly done at level of the current ATM scenario, trying to investigate different typologies of sectors and how these impact on the delay propagation. Then one can move at the level of the future SESAR scenario, in order to investigate whether the new scenario will bring the desired improvement of capacity.

Bibliography

- [1] B. Delaunay, *Sur la sphère vide*, Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk, **7**, 793-800, (1934).
- [2] P. Erdős and A. Rényi, *On the evolution of random graphs*. in: Publications of the Mathematical Institute of the Hungarian Academy of Sciences **5**, 17-61 (1960).
- [3] E.W. Dijkstra, *A note on two problems in connexion with graphs* . Numerische Mathematik **1**, 269-271 (1959).
- [4] J.Y. Yen, *Finding the k Shortest Loop-less Paths in a Network* . Management Science **17** (11), 712-716 (1971).
- [5] David E. Goldberg *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley Professional, 1 edition, 1989)
- [6] G Van Rossum; Fred L. Drake Jr *Python reference manual* (Centrum voor Wiskunde en Informatica, Netherlands, 1995)
- [7] Brian W. Kernighan; Dennis M. Ritchie *The C Programming Language* (Prentice Hall, England, 1988)
- [8] D. Delahaye, S. Puechmorel *Air traffic Complexity: towards intrinsic metrics* in: Proc. 3rd USA/Eur. Air Traffic Manage. R&D Semin., Napoli, Italy, Jun. 2000
- [9] Banavar Sridhar, Kapil S. Sheth, Shon Grabbe *Airspace Complexity and its Application in Air Traffic Management* in: 2nd USA/Eur. Air Traffic Manage. R&D Semin., Orlando,USA, December 1998
- [10] I. V. Laudeman, S. G. Shelden, R. Branstrom, and C. L. Brasil *Dynamic Density: An Air Traffic Management Metric* in: NASA/TM1998-112226
- [11] Jonathan M. Histon, R. John Hansman, Blake Gottlieb, Howard Kleinwaks, Sarah Yenson, Daniel Delahaye, Stephane Puechmorel, *Structural considerations and cognitive complexity in Air Traffic Control* in: Proceedings of The 21st Digital Avionics Systems Conference, 27-31 Oct. 2002.
- [12] G. B. Chatterji, B. Sridhar *Measures for air traffic controllers workload prediction* in: Proceedings of 1_{st} AIAA, Aircraft, Technology Integration, and Operations forum, 2001
- [13] Complex World - SESAR WP-E Research Network for the theme “Mastering Complex Systems Safely” , *The ComplexWorld Position Paper* , Version: August 2012, (Restricted audience)
- [14] A. Clauset, C. R. Shalizi, M. E. J. Newman, *Power-law distributions in empirical data* SIAM Review, **51**, 661-703, (2009)

- [15] G. E. Mueller *Radial porosity in packed beds of spheres* Powder Technology, **203** (3), 626633, (2010)
- [16] C. Bongiorno et al. *An Agent Based Model of Air Traffic Management*, in: Proceedings of the Sesar Innovation Day, 2013.

A Topology properties of the networks

In this annex we compile we obtained on some topological measures of the generated networks.

The capacity of filling the airspace with many flights departing at the same (or close) time and connecting the same airports depends crucially on the number of different ways of connecting the two airports with paths of the same or similar length. This is due to the fact that AOs considers the path length as a key variable in the cost function. However two shortest paths can have a significant fraction of nodes (sectors) in common with the result that they cannot be filled as if they were without overlaps. For this reason, before running the simulations and in order to understand how much the topology could affect the dynamics of the flight allocation, we studied the properties of the paths which can be found between two nodes. It is important to notice that this analysis is of great importance in the unweighted model, because when the crossing time (or length) of each sector is the same there will be several paths connecting two airports of *exactly* the same length. In the weighted case with real numbers as weight the probability that two paths have the same length is vanishingly small and this is not anymore a constraint.

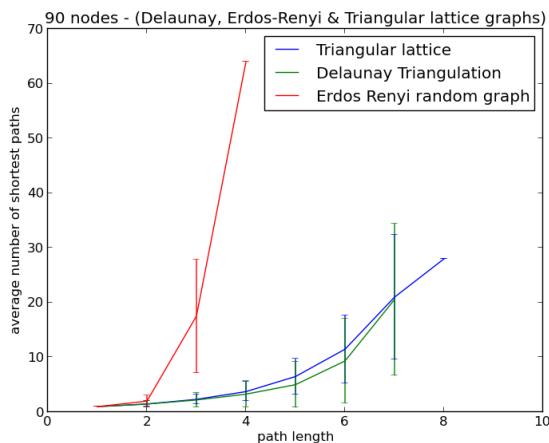


Figure A.1: Average number of shortest paths as a function of the path length for the Delaunay triangulation, triangular lattice and Erdos Rényi random graph. The order (number of nodes) of the graph is 90.

We started by computing and plotting, for a given graph topology, the average number of shortest paths between two nodes as a function of the path length between them (see Fig. A.1). We observe that while the Delaunay triangulation and the triangular lattice exhibit similar behavior, the Erdos Rényi random graph has a much larger number of shortest paths for a given distance as compared to planar graph. Moreover we find that the maximum length of the shortest paths of the Erdos Rényi

random graph is 4, with a large number of them sharing that length. This fact can be understood qualitatively by recalling that for large networks the graph diameter D – the length of the longest shortest path – in an Erdos Rényi random graph is very short ($D \sim \log N$, where N is the number of nodes in the graph).

We then plot the “overlap” between the shortest paths as a function of the path length (Fig. A.2). Overlap is defined as the number of common nodes, excluding the starting and the ending ones, in two shortest paths. We find a small overlap among the shortest paths in the Erdos Rényi graph as compared to the planar graphs. In the presence of a high overlap between shortest paths, if a shared node is overloaded, a large amount of FPs can be rejected, while if the overlap is low it is more difficult to have “bottlenecks” in the airspace.

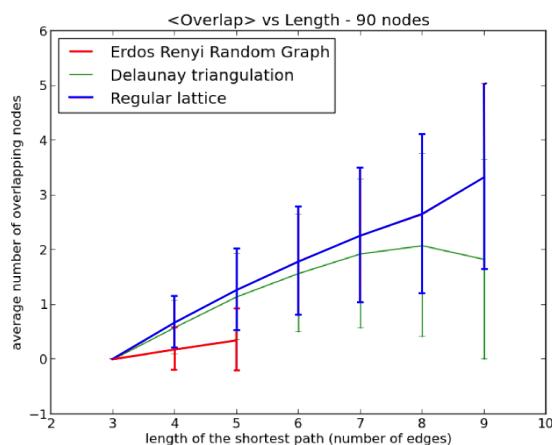


Figure A.2: Average number of overlapping nodes found among the shortest paths (starting and ending nodes of a path are not taken into account) connecting two given nodes, as a function of the path length.

B Analytical study of the rejection

Here we present the first results of a semi-analytical study to predict the rejections of flight plans when they are submitted to the NM.

B.1 Probabilities without overlap

B.1.1 First rejection

We consider only two airports having N_{sp} shortest paths (same length) between them. First we assume that the paths are completely distinct, with no overlap (except for the two airports). We also assume that all nodes between the airports have the same capacity c and that the flight plans are all starting at the same time. The question is then: how many flight plans can I fill before I have to reject one because one sector is full? Since every nodes have the same capacity, we can put a maximum of cN_{sp} flight plans before the rejection. However, if we fill them randomly, the average is smaller.

We note x_i the number of flight plans allocated to the i^{th} path and $\mathbf{x} = (x_1, \dots, x_{N_{sp}})$. If there were no restriction on capacity, the probability of having \mathbf{x} with n flight plans would follow the multinomial distribution:

$$P_{mul}(\mathbf{x}, n) = \frac{n!}{x_1! \dots x_{N_{sp}}!} p^n, \quad (\text{B.1})$$

where $p = 1/N_{sp}$ is the probability for a flight plan to be allocated to one path. The probability we are looking for can then be calculated as follows. At step $n - 1$, the probability of having a state \mathbf{x} is given by $P_{mul}(\mathbf{x}, n - 1)$. Given this state, the probability that a path is overloaded at the next step is equal to $p \times l$ where:

$$l = \sum_{j=1}^{N_{sp}} \delta_{x_j, c}$$

is the number of path already reaching the capacity c at step $n - 1$ (δ is the Kronecker symbol). Hence, if we sum on all the possible states at step $n - 1$, *i.e.* those where all the x_j are smaller or equal to c , we get:

$$\begin{aligned} P_{free}^1 &= \sum_{x_1=0}^c \dots \sum_{x_{N_{sp}}=0}^c P_{mul}(\mathbf{x}, n - 1) \times p \sum_{j=1}^{N_{sp}} \delta_{x_j, c} \\ &= \frac{(n-1)!}{N_{sp}^n} \sum_{x_1=0}^c \dots \sum_{x_{N_{sp}}=0}^c \frac{1}{x_1! \dots x_{N_{sp}}!} \delta_{n-1, \sum_{j=1}^{N_{sp}} x_j} \times \sum_{j=1}^{N_{sp}} \delta_{x_j, c}. \end{aligned}$$

However, the previous probability P_{mul} is not strictly valid here because at step $n - 1$ one has to consider, as all possible cases, only the ones where the x_i are smaller than c . To this end, we must divide the probability P_{free} by the probability that all x_i are smaller than c at step $n - 1$, which is:

$$P_{res}^1 = \frac{(n-1)!}{N_{sp}^{n-1}} \sum_{x_1=0}^c \dots \sum_{x_{N_{sp}}=0}^c \frac{1}{x_1! \dots x_{N_{sp}}!} \delta_{n-1, \sum_{j=1}^{N_{sp}} x_j}. \quad (\text{B.2})$$

It follows that the probability that a flight plan is rejected at step n is:

$$P_{rej}^1(n, N_{sp}, c) = \frac{P_{free}^1}{P_{res}^1}$$

Now we want the probability $P_{jam}(n, N_{sp}, c)$ that the rejection occurs at the n^{th} flight plan. For this, we must have a rejection at the n^{th} and no rejection before. Hence we have:

$$P_{jam}^1(n, N_{sp}, c) = P_{rej}^1(n, N_{sp}, c) \times (1 - P_{rej}^1(n-1, N_{sp}, c)) \times (1 - P_{rej}^1(n-2, N_{sp}, c)) \dots (1 - P_{rej}^1(1, N_{sp}, c)). \quad (\text{B.3})$$

Finally, the average number of flight plans before the first rejection is:

$$\langle n \rangle^1 = \sum_{n=0}^{cN_{sp}+1} n P_{jam}^1(n, N_{sp}, c) - 1 \quad (\text{B.4})$$

B.1.2 Second rejection and higher

The computation of the second, third, etc. follows the same line. Now, if we want to consider the probability of second rejection at step n , we need to have the probability of a state \mathbf{x} at step $n - 1$ where exactly one of the component is equal to $c + 1$, which is $N_{sp} P_{mul}(x_1 = c + 1, x_2 \dots x_{N_{sp}}, n - 1)$. Then we need to multiply by pl , where l is the number of components which are equal or greater than c , which is: $l = \sum_{j=2}^{N_{sp}} \delta_{x_j, c} + 1$. Finally, we have:

$$P_{free}^2 = \frac{(n-1)!}{N_{sp}^{n-1}} \frac{1}{(c+1)!} \sum_{x_2=0}^c \dots \sum_{x_{N_{sp}}=0}^c \frac{1}{x_2! \dots x_{N_{sp}}!} \delta_{n-1, \sum_{j=2}^{N_{sp}} x_j + c + 1} \times \left(\sum_{j=2}^{N_{sp}} \delta_{x_j, c} + 1 \right),$$

which would be the probability without constraint on the capacities. We also have:

$$P_{res}^2 = \frac{(n-1)!}{N_{sp}^{n-2}} \frac{1}{(c+1)!} \sum_{x_2=0}^c \dots \sum_{x_{N_{sp}}=0}^c \frac{1}{x_2! \dots x_{N_{sp}}!} \delta_{n-1, \sum_{j=2}^{N_{sp}} x_j + c + 1}. \quad (\text{B.5})$$

It follows that the real probability of rejection is:

$$P_{rej}^2(n, c, N_{sp}) = \frac{P_{free}^2}{P_{res}^2}.$$

The probability that the second rejection occurs at the n^{th} step is then the probability to have a first rejection at the k^{th} step, where $k < n$ and that there were no second rejection between. Hence, we have:

$$P_{jam}^2(n, c, N_{sp}) = P_{rej}^2(n, c, N_{sp}) \times \sum_{k=0}^{n-1} \left(P_{jam}^1(k, c, N_{sp}) \prod_{j=k+1}^{n-1} (1 - P_{rej}^2(j, c, N_{sp})) \right)$$

B.2 Probability with overlap

We are now interested in calculating the same probabilities when the paths are overlapping. For the sake of simplicity, we show here a “mean-field” approximation of these probabilities.

First, let us note Q_{ij} the probability that paths i and j have at least one common node. We note also $Q = \langle Q_{ij} \rangle$ its average, and $k = Q(N_{sp} - 1)$ the average number of “neighbors” (overlapping paths) of a given path. Then, in average, everything looks as if the capacity of the nodes were lowered by $q = Q(N_{sp} - 1)n/N_{sp}$, where n is the number of flight plans already filled. For instance the probability P_{free}^1 now reads:

$$P_{free}^1 = \frac{(n-1)!}{N_{sp}^n} \sum_{x_1=0}^{c-q} \dots \sum_{x_{N_{sp}}=0}^{c-q} \frac{1}{x_1! \dots x_{N_{sp}}!} \delta_{n-1, \sum_{j=1}^{N_{sp}} x_j} \times \sum_{j=1}^{N_{sp}} \delta_{x_j, c-q}. \quad (\text{B.6})$$

All the previous probabilities are modified accordingly.

B.3 Numerical results

We now show some plots displaying the previous quantities, computed numerically. Note that even numerically and even for small values of c , N_{sp} , Q and n the calculations can take quite a long time due to the multiple sums and products (the *number* of sums, as well as their length, varies with these parameters, in particular c). We also performed some basic simulations to see if our results were correct. They are performed very simply, by throwing randomly the flight plans on a certain number of paths, and measuring the times of rejections.

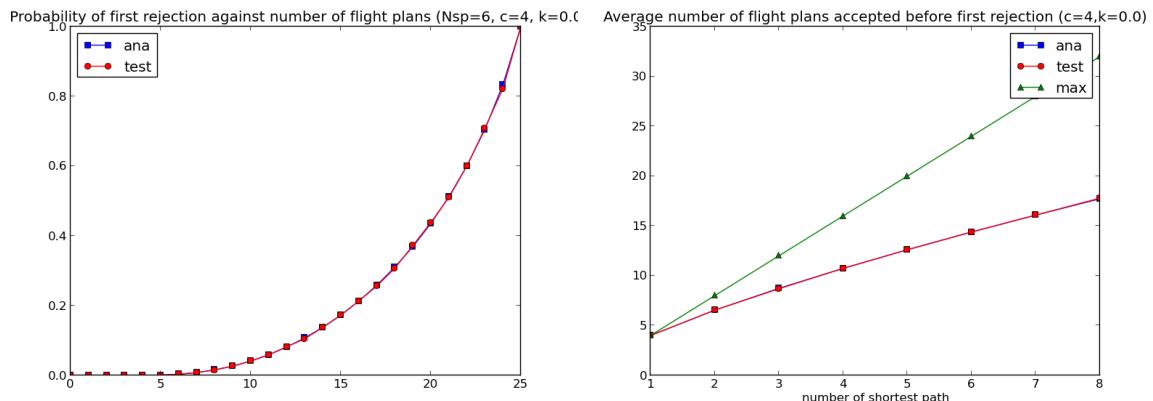


Figure B.1: Left: probability of first rejection as a function of the number of flight plans submitted for $N_{sp} = 6$, $c = 4$. The blue line is the analytical prediction, the red one comes from simple simulations. Right: average number of flight plans submitted before first rejection as a function of the number of shortest paths, for the analytical expression and the simulations. The green line is the maximum one could obtain (planifications).

As one can see on figure B.1, the agreement between simulations and analytical results is really good. Moreover, we see that, as expected, the average number of accepted flights before first rejection is much smaller with “random” strategy than it is with a “planning” strategy (in green).

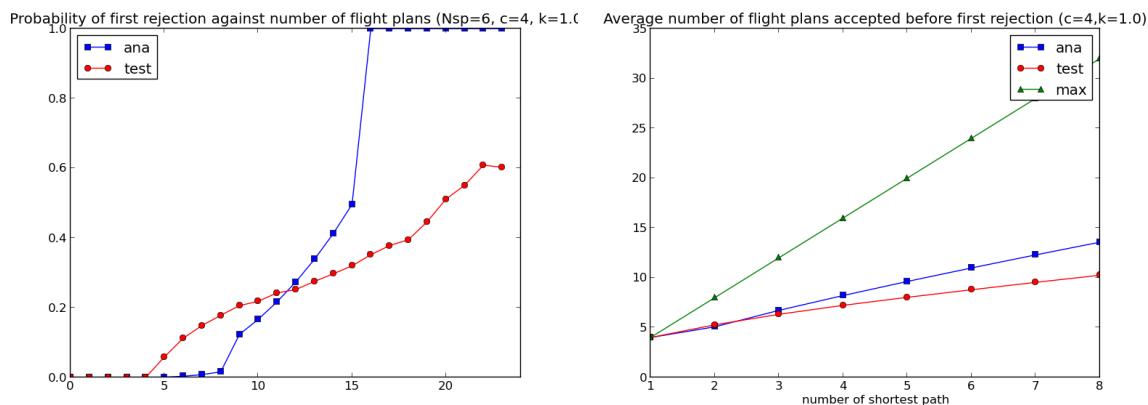


Figure B.2: Same figure than figure B.1 with an overlap $Q = 1/N_{sp}$ ($k = 1$).

On the contrary, figure B.2 shows that the mean-field approach described previously does not seem to be enough to grasp the overlapping phenomenon, since the analytical results and the simulations are significantly different. Figures B.3 and B.4 show more precise graphs where we see the evolution of the probability of rejection against the overlap Q and the number n of flight plans submitted. In particular, we see that the evolution with the overlap is much smoother in the simulations than in the analytical results. This is due to the fact that the parameter q in equation B.6 has to be taken as the closest integer in fact, which leads to results “step by step”.

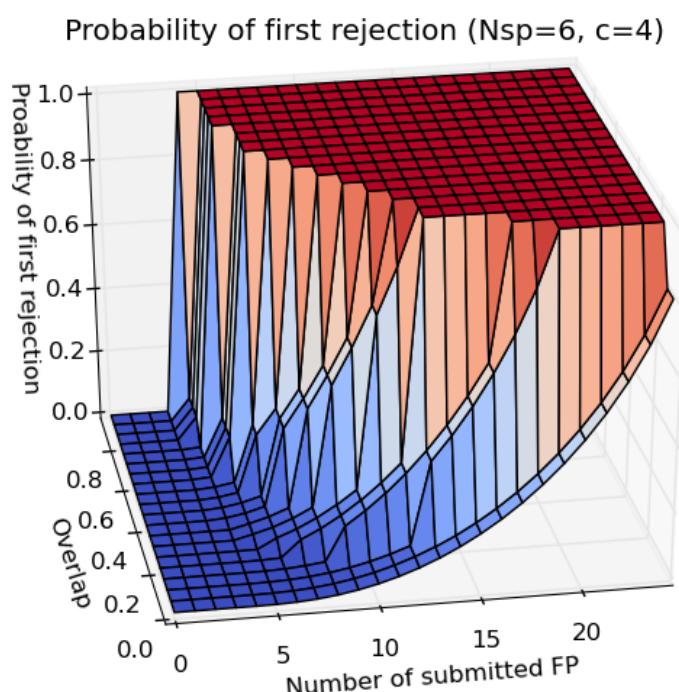


Figure B.3: Evolution of the probability of first rejection against overlap and number of flight plans submitted.

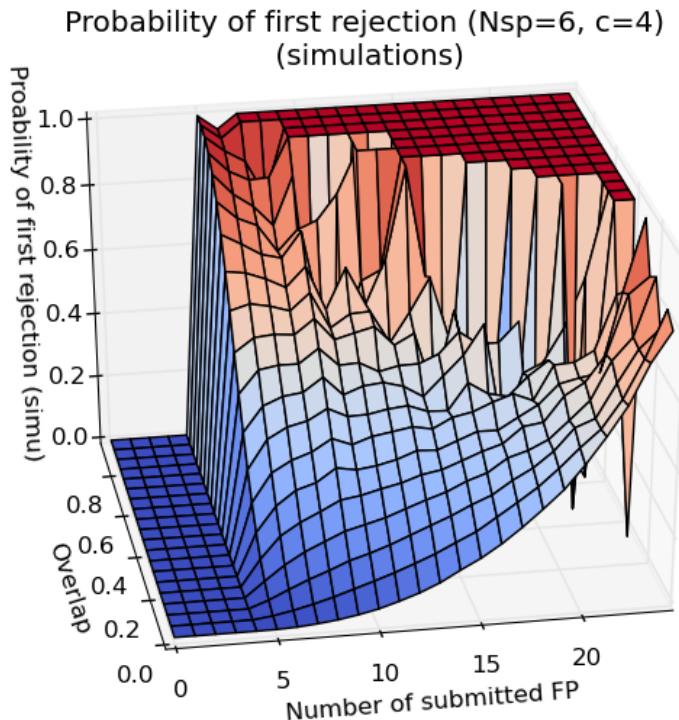


Figure B.4: Evolution of the probability of first rejection against overlap and number of flight plans submitted in the simulations. The noise becomes very big for high Q and n (overlap and number of flight plans), because the occurrences of a high number of flights plans rejected are rare and thus the statistic poor.

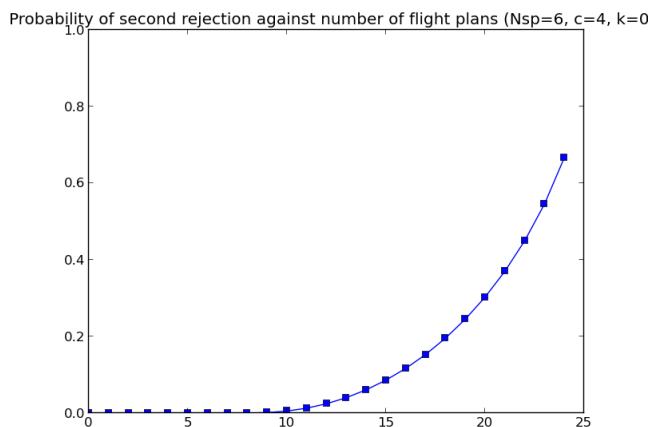


Figure B.5: Probability of second rejection against number of flights plans submitted.

C Probability to shift/reroute

In this annex we present the details of the calculation of the probability to pick a longer path rather than shift the flight plan in time, presented in section 1.6.1.2.

We remind that the probability that the flight plan with cost $\alpha L_1 + \beta$ is included into the list of flight plans is equal to the probability that

$$\alpha L_n > \alpha L_1 + \beta \rightarrow L_n - L_1 > \frac{\beta}{\alpha}. \quad (\text{C.1})$$

Remember that L_1 and L_n are the minimal and maximal weighted length in a sample of n . Since the weights are randomly assigned, the problem is to find the max and the min in a sample of random numbers. In order to simplify calculations, we assume that these random numbers are independent. This is not true because of the overlap between the edges of the paths.

Under this assumption and considering that the weights are Gaussian distributed with mean μ and standard deviation σ_0 , we can conclude that $L_i \sim \mathcal{N}(\mu\ell, \sigma_0^2\ell)$, i.e. L_i is also Gaussian. For a Gaussian variable $\mathcal{N}(m, \sigma)$ it is

$$P(x < y) = \frac{1}{2} \operatorname{erfc} \left[\frac{m - y}{\sqrt{2}\sigma} \right], \quad (\text{C.2})$$

thus for a sample of n independent and identically distributed variables the maximum has distribution

$$P(x_{max} < y) = [P(x < y)]^n = \left(\frac{1}{2} \operatorname{erfc} \left[\frac{m - y}{\sqrt{2}\sigma} \right] \right)^n. \quad (\text{C.3})$$

We want to compute the “typical” value for the max and we consider the median $\Lambda_{max} = \text{Median}[x_{max}]$, defined as $P(x_{max} < \Lambda_{max}) = 1/2$. Solving this equation we get

$$\Lambda_{max} = m - \sqrt{2}\sigma \operatorname{erfc}^{-1}(2^{1-1/n}), \quad (\text{C.4})$$

where erfc^{-1} is the inverse erfc function. One can proceed similarly for the minimum obtaining

$$\Lambda_{min} = m - \sqrt{2}\sigma \operatorname{erfc}^{-1}(2 - 2^{1-1/n}). \quad (\text{C.5})$$

Thus the typical value of the max and min weighted path is

$$L_1 \simeq \mu\ell - \sqrt{2}\sqrt{\ell}\sigma_0 \operatorname{erfc}^{-1}(2 - 2^{1-1/n}) \quad (\text{C.6})$$

$$L_n \simeq \mu\ell - \sqrt{2}\sqrt{\ell}\sigma_0 \operatorname{erfc}^{-1}(2^{1-1/n}). \quad (\text{C.7})$$

The range of the weighted length in equation 1.6 is

$$\sqrt{\ell}\sigma_0\sqrt{2}(-\operatorname{erfc}^{-1}(2^{1-1/n}) + \operatorname{erfc}^{-1}(2 - 2^{1-1/n})) > \beta/\alpha \rightarrow \sqrt{\ell}\sigma_0 g_n > \beta/\alpha. \quad (\text{C.8})$$

where g_n is a function of the number of flight plans (for example $g_5 = 2.26$, $g_{10} = 2.997$, and $g_{20} = 3.648$). Finally we have a time shifted flight plan if

$$\sigma_0 > \frac{\beta}{\alpha} \frac{1}{g_n \sqrt{\ell}}. \quad (\text{C.9})$$

D Details of the Genetic algorithm

Below we illustrate the three operators used in the genetic algorithm and their role.

1. **Mutation operator 1** - The first operator acts a change of a v_i randomly selected in the speed array with another v_i^* under the condition $v_{min} < v_i^* < v_{max}$ and therefore it modifies all positions after the selected segment¹, as illustrated in the left part of Fig. D.1.
2. **Mutation operator 2** - The second operator is applied to contiguous speeds v_x and v_y and acts as a change of them by a small percentage p_x and p_y , where

$$p_y = \frac{t_y}{\frac{p_x-1}{p_x}t_x + t_y} \quad (\text{D.1})$$

where t_x is the time used to travel the x segment and t_y is the time used to travel the y segment. This relation is obtained to leave invariate the position in the array P for the other segments. The schematic behavior of this operator is illustrated in the right part of Fig. D.1.

3. **Role of Mutation** - These two operators have a different level of intrinsic randomness. While the second one generates an element very similar to the original, the first one might generate an element very different. This peculiarity can be useful to avoid the uniformity of elements in the generated populations. In fact, at each iteration we estimate a global parameter of this uniformity defined as $U = \left(\frac{\sigma_{\mathcal{F}_1}}{\langle \mathcal{F}_1 \rangle} \right)^2 + \left(\frac{\sigma_{\mathcal{F}_2}}{\langle \mathcal{F}_2 \rangle} \right)^2$. A good rule that ensures to hold a medium level of U during the evolution of the algorithm is to increase the probability to apply the first operator with the increase of U and increasing the probability to apply the second operator with the decreasing of U . This will be achieved by applying the two operators with probabilities $p_2 = \frac{2p_{max}}{\pi} \arctan(\alpha U)$ for the second one, and $p_1 = \frac{2p_{max}}{\pi} (1 - \arctan(\alpha U))$ for the first one, where p_{max} and α are tunable parameters.
4. **Cross-over operator** - This is the third operator we consider in the algorithm. The idea behind it is that two elements of the population have a good fitness because they have inside of their position array \mathcal{P} some subsets of points with high performance. Therefore one assumes that a mixture of these elements might further improve the fitness of the system. This operator must be applied paying attention to generate a child element not so different from parents. This operator is applied to the positions P_i related to the velocity v_i . It selects two elements of the population \mathcal{P}_M and \mathcal{P}_F (mother and father), under the assumption that the velocity changes only at the Navigation Points². It therefore selects two Navigation Point N_x and N_y of the trajectory, and identifies the joint points on P_M and P_F . Let us suppose that $\{p_0 \dots p_m\}$ is the first subset of points before N_x of P_M , and $\{p'_{t+1} \dots p'_N\}$ is the last subset after N_y of P_F . Under the condition $m < t$, the new element generated with this procedure will be

¹Each v_i has a related segment between two navigation points.

²We are planning a more general rule to generate a set of m points in which the aircraft could change his velocity

$\{p_0 \dots p_m, p''_{m+1} \dots p''_t, p'_{t+1} \dots p'_N\}$, to obtain the velocity for the subset $\{p''_{m+1} \dots p''_t\}$ it is necessary to estimate the time t_x and t_y in which the aircraft runs respectively into N_x and N_y . The velocity needed for matching these two subsets will be $v_2^* = \frac{d(N_x, N_y)}{t_y - t_x}$ if $v_{min} < v_2^* < v_{max}$ this solution will be accepted.

5. **Role of Cross-over** - The cross-over operator is introduced to ensure a fast convergence of the algorithm. This operator will be applied with a characteristic probability. The Cross-Over operator selects two elements of the population according with the Gaussian distribution (see Mutation).

The genetic algorithm runs for a certain number of iterations until the two fitness values change for an amount smaller than a pre-determined value with respect to the previous iteration. When the genetic algorithm ends all iterations, it selects the first element of the population labeling it with the value $\mathcal{F}_2 = d(P_{end}, P_{end}^*)$.

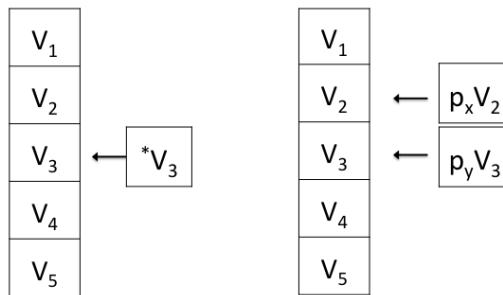


Figure D.1: On the left the figure illustrates the first mutation operator. It modifies a velocity on an element of population v . As a result all points in P after v_2 will be modified. On the right the figure illustrates the second mutation operator. It modifies two contiguous velocity on v according to Eq. (D.1). One velocity is increased and the other is correspondingly decreased. As a result all points before v_2 and after v_3 do not change. Further details are given in the text.

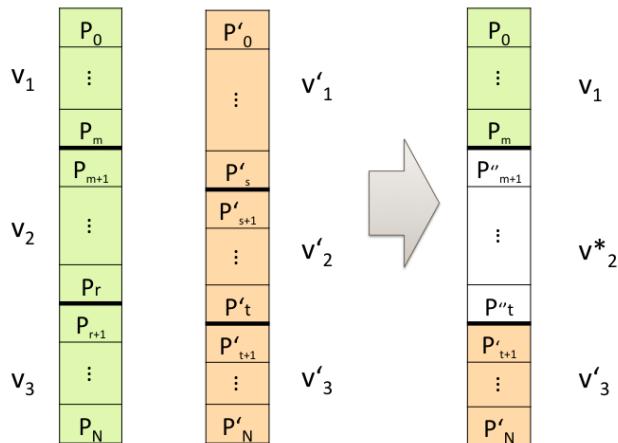


Figure D.2: The figure illustrates the Crossover operator. It acts on two father and mother elements and generate P , the child element. As much as in the biological/genetic crossover P inherits some genetic information from both parents.

E Other simulations

In Fig. E.1 and Fig. E.2 we show the same metrics already seen in Fig. 2.15 and Fig. 2.16 respectively, for the case when the radius of the shocked area is equal to 5 km rather than 10 km. The behavior of the considered metrics is qualitatively the same in the two cases.

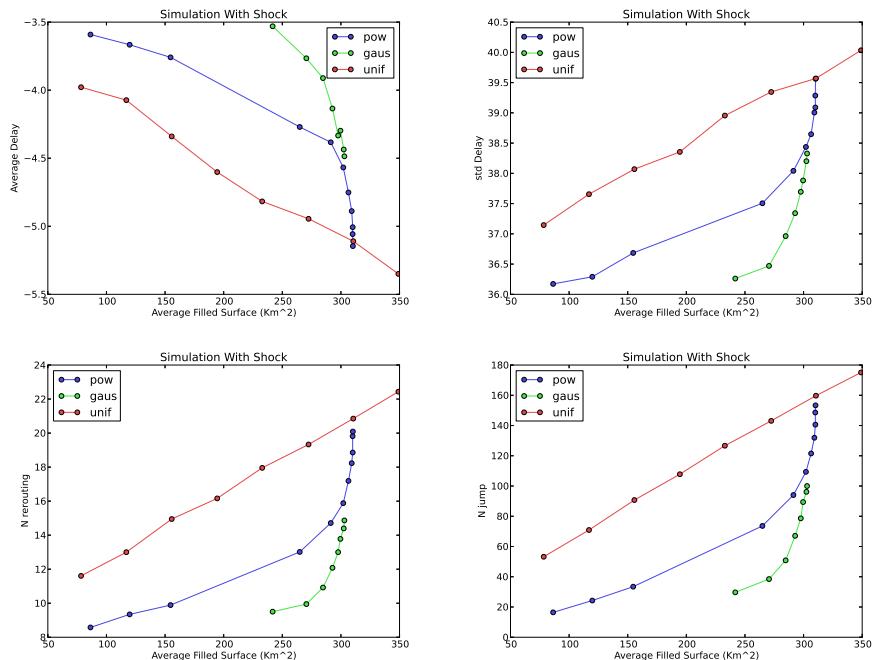


Figure E.1: Shocked area of radius 5 km. In the top panels we show the average and the standard deviation of delay for the three considered distributions. In the bottom panels we show the average number of reroutings and flight level changes for the three distributions considered. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ .

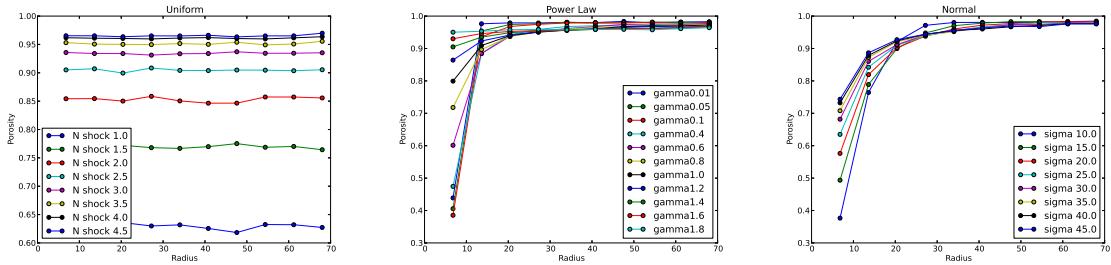


Figure E.2: Shocked areas of radius 5 km. The figure shows the radial porosity [15] for the three different considered distributions: uniform in the left panel, power-law in the central panel and gaussian in the right panel. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ .

In Fig. E.3 and Fig. E.4 we show the same metrics already seen in Fig. 2.15 and Fig. 2.16 respectively, for the case when the radius of the shocked area is equal to 15 km rather than 10 km. The behavior of the considered metrics starts to be different in the two cases, as far as the average delay is concerned. In fact we can observe that the slope of the curves relative to the uniform and power-law distributions is the opposite with respect to the previous case.

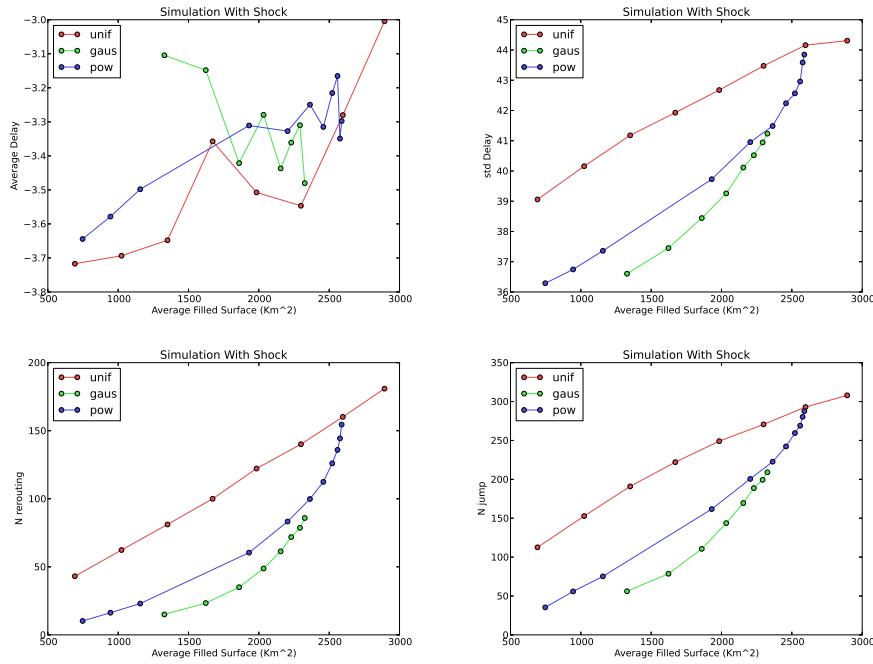


Figure E.3: Shocked area of radius 15 km. In the top panels we show the average and the standard deviation of delay for the three considered distributions. In the bottom panels we show the average number of reroutings and flight level changes for the three distributions considered. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ .

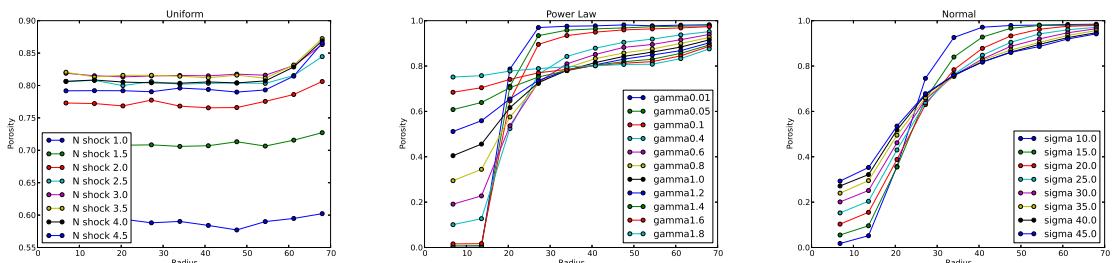


Figure E.4: Shocked areas of radius 15 km. The figure shows the radial porosity [15] for the three different considered distributions: uniform in the left panel, power-law in the central panel and gaussian in the right panel. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ .

In Fig. E.5 and Fig. E.6 we show the same metrics already seen in Fig. 2.15 and Fig. 2.16 respectively, for the case when the radius of the shocked area is equal to 20 km rather than 10 km. The behavior of the considered metrics is different in these two cases, as far as the average delay

is concerned. In fact we can observe that the slope of all three curves is the opposite with respect to the case in Fig. 2.15 and Fig. 2.16. We can observe that the number of flight level changes and vectorings starts to be very similar.

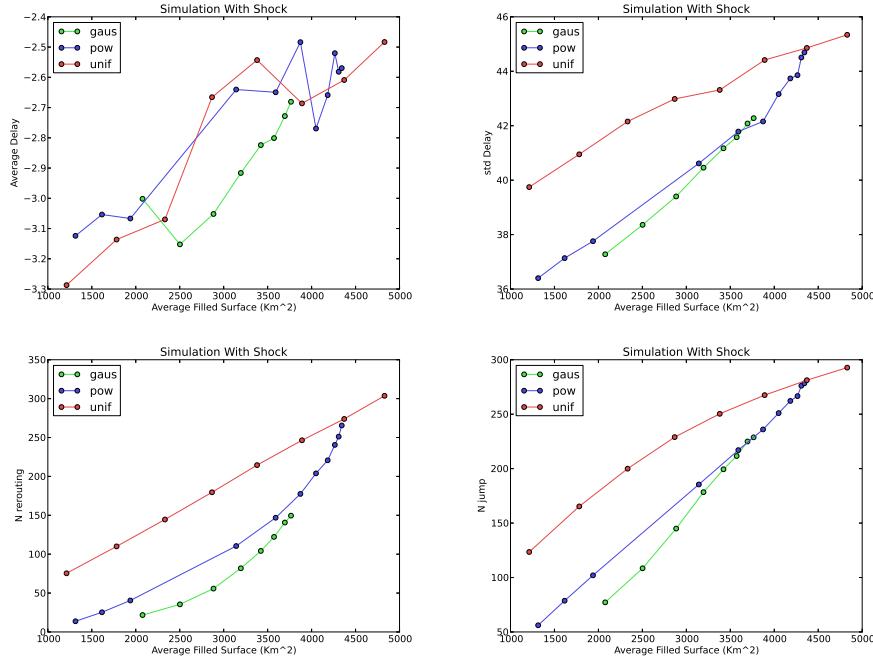


Figure E.5: Shocked area of radius 20 km. In the top panels we show the average and the standard deviation of delay for the three considered distributions. In the bottom panels we show the average number of reroutings and flight level changes for the three distributions considered. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ .

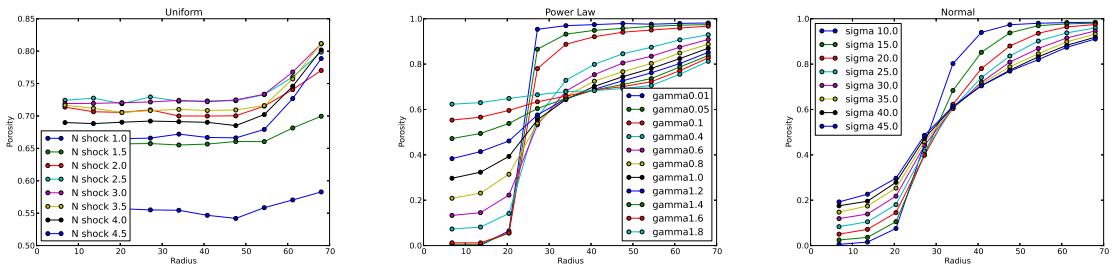


Figure E.6: Shocked areas of radius 20 km. The figure shows the radial porosity [15] for the three different considered distributions: uniform in the left panel, power-law in the central panel and gaussian in the right panel. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ .

In Fig. E.7 and Fig. E.8 we show the same metrics already seen in Fig. 2.15 and Fig. 2.16 respectively, for the case when the radius of the shocked area is equal to 25 km rather than 10 km. Again, the behavior of the considered metrics is different. In fact, we here confirm the behavior already observed in Fig. E.5 and Fig. E.6. We can observe that the number of flight level changes is here slightly smaller than the number of vectorings, which is the opposite of what we observe in Fig. 2.15.

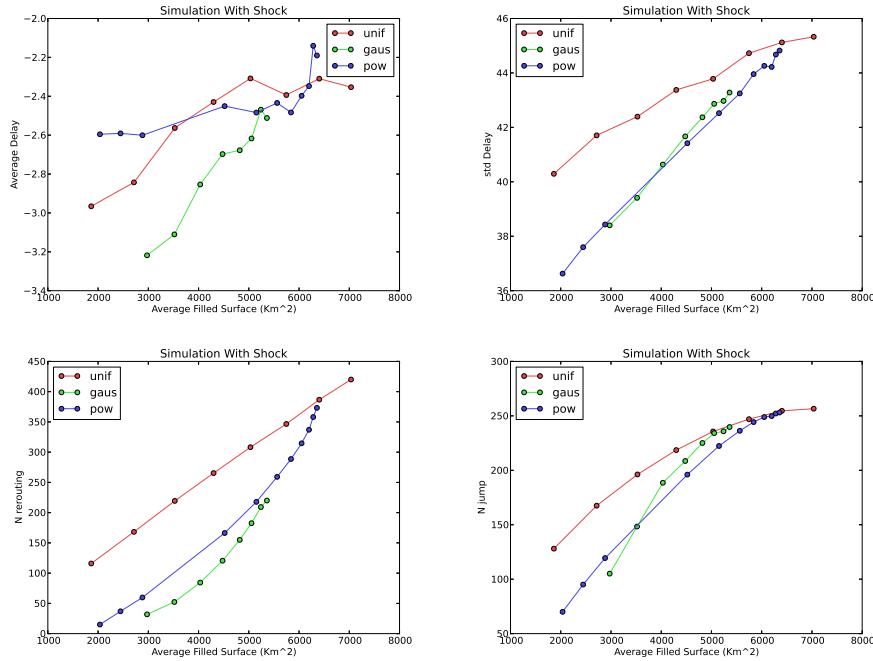


Figure E.7: Shocked area of radius 25 km. In the top panels we show the average and the standard deviation of delay for the three considered distributions. In the bottom panels we show the average number of reroutings and flight level changes for the three distributions considered. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ .

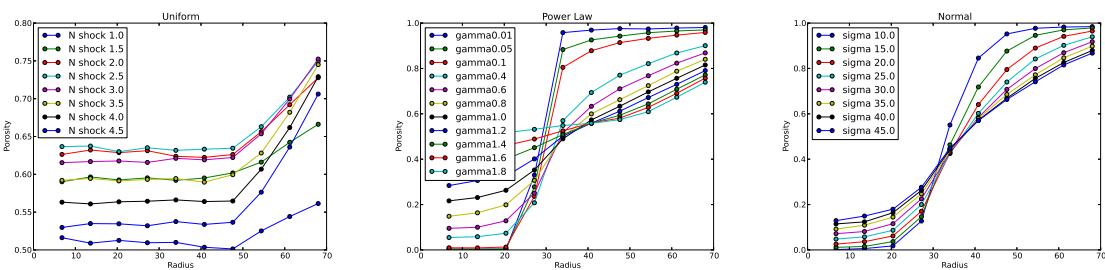


Figure E.8: Shocked areas of radius 25 km. The figure shows the radial porosity [15] for the three different considered distributions: uniform in the left panel, power-law in the central panel and gaussian in the right panel. These data are shown for 8 different values of the average number n of shocks per time-step mentioned above in the case of the uniform distribution. In the case of power-law and gaussian distribution we have fixed $n = 4$ for all the mentioned values of γ and σ .