EXPLORATORY RESEARCH

SESAR JOINT UNDERTAKING

D5.1 First tactical model and results

Deliverable ID:	**D5.1**
Dissemination Level:	**PU**
Project Acronym:	**BEACON**
Grant:	**893100**
Call:	**H2020-SESAR-2019-2**
Topic:	**SESAR-ER4-08-2019**
Consortium Coordinator:	**University of Westminster**
Edition date:	**10 November 2022**
Edition:	**01.01.00**
Template Edition:	**02.00.05**

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Authoring & Approval

Authors of the document

Name / Beneficiary	Position / Title	Date
Gerald Gurtner / Uow	Project coordinator	10/11/2022

Reviewers internal to the project

Name / Beneficiary	Position / Title	Date
Tanja Bolic / UoW	Project Team	12/09/2022

Reviewers external to the project

Name / Beneficiary	Position / Title	Date

Approved for submission to the SJU By - Representatives of all beneficiaries involved in the project

Name / Beneficiary	Position / Title	Date
Gerald Gurtner / Uow	Project Coordinator	10/11/2022

Rejected By - Representatives of beneficiaries involved in the project

Name and/or Beneficiary	Position / Title	Date

Document History

Edition	Date	Status	Name / Beneficiary	Justification
00.00.01	29/07/2022	Internal	Gerald Gurtner / UoW	First draft
00.00.02	29/08/2022	Internal	Gerald Gurtner / UoW	Second draft
01.00.00	16/09/2022	Rejected	Gerald Gurtner / UoW	First submission to SJU
01.01.00	10/11/2022	Submitted	Gerald Gurtner / UoW	Second submission to SJU

BEACON

BEHAVIOURAL ECONOMICS FOR ATM CONCEPTS

This deliverable is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 893100 under European Union's Horizon 2020 research and innovation programme.

Abstract

The deliverable presents the results obtained with the help of the Mercury simulator in order to estimate the efficiency and equitability of hotspot resolution mechanisms defined in D3.1 and tested on a small-scale simulator in D4.2. To this end, fast-time games are defined, with a central optimiser implementing the mechanisms computing the final flight/slot allocation. The game is played by agents representing the airlines present in the regulation, and who are tasked with sending information to the central optimiser regarding their own costs, in order for the latter to find the best possible allocation cost-wise. The deliverable defines the various games possible, combining different types of central optimisers and different types of agents, taking into account various degrees of rationality and behavioural biases for the decisions of the latter. The deliverable presents a theoretical framework for these behaviours, highly simplified but implementable in simple simulations.

Different results with these simulations are presented. First, it is found that the approximation process used by airlines to communicate their costs to the optimiser has a major effect on the efficiency of the mechanisms. Second, it is found that, due to the variance of the regulation structure and the approximation issues, defining a performant rational agent is difficult. Despite this fact, the deliverable shows how one of the mechanisms (credit mechanism) was calibrated, including the rationality of agents in the corresponding game. It then shows the detrimental impact of the presence of rational agents – as opposed to honest ones – on the efficiency of two relevant optimisers, including the credit mechanism, as well as the impact of behavioural biases. Finally, it compares the aggregated gains potentially made with the various mechanisms in terms of efficiency and equity.

Table of Contents

List of Tables

List of Figures

1 Introduction

In this deliverable, we explore the mechanisms defined in D3.1 with more advanced cost functions and regulations than what was done in D4.2. Indeed, in this deliverable we use a tactical simulator called Mercury in order to extract realistic regulations that will then be used in quick simulations that we call games in the following. These games feature two main ingredients: a central optimiser, which can be thought of being part of the network manager, and the players, i.e. the airlines involved in the regulations. Hence, the main goal on this deliverable is to see if the new mechanisms defined for regulation resolution at arrival airports are good enough, as least in theory, taking into account fairly realistic behaviours from airlines.

In fact, we are interested in this deliverable in exploring the efficiency of the central optimiser when faced with different behaviours from the players, but also in the behaviours themselves. For this, we play, fast-time, a series of “games” which are always played the same way: we first select how the regulation will be solved (for instance, basic UDPP), then the type of agents that will play the game: rational, honest, etc. We then sample regulations from a dataset we built (from historical data), ask the players to play the game, which implies for them to communicate some information on their cost functions to the central optimiser. The central optimiser can then compute the final allocation, based on its type and the information received. One can then compare this allocation to the allocation that would have been applied without any mechanism, the “first planned first served” (FPFS) one. Indicators related to how well the solving algorithm performs can then be computed. Indeed, the impact is typically estimated in terms of efficiency (how much the airlines gain in terms of cost) and equity (how much they gain with respect to other airlines).

The simulations can also estimate the impact of the mechanisms when faced with agents applying different strategies, including agents that may be prone to behavioural biases for instance. They can also estimate the impact for different players, for instance major players vs. low-volume users. Hence, we are able in this deliverable to include a rational agent in the midst of honest agents, which may improve the efficiency for this particular agent, but decrease the efficiency for others. Since the regulations are more realistic than those used in D4.2, in particular in terms of cost structure of the airlines, the picture painted by the simulations presented in this deliverable may be considered as more accurate and may help a policymaker to make an informed decision for a future implementation of these mechanisms.

The deliverable is structured in the following way:

Section 2 presents the definition of the mechanisms and the different agent types used in the simulations. We also highlight in this section the most important experiments to be performed and why we perform them.

Section 3 presents a theoretical framework for rational and agents with behavioural biases. The final implementation is a simplified version of the general case, due to the impossibility to derive a closed form for the optimal strategy in the credit mechanism.

Section 4 presents the results of the simulations. We explore several types of results. First, we explore how much the approximation used by the airlines when communicating their cost impacts the results of the simulation. The next step is to examine the impact of the decisions of isolated agents on their own cost. Due to various reasons, among which the approximation process and the stochasticity of the regulation structure, agents struggle to properly map their actions to an expected gain in terms of cost.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

This has a direct impact on the quality of the rational agent, which is supposed to be able to do this map particularly well in order to make the best decision. After that, we move on to see how having one or more rational players in the simulations impact the final allocation given by a specific mechanism. After doing the same for behavioural biases, we conclude this section by showing a high-level comparison of the goodness of the different mechanisms in terms of efficiency and equity.

In conclusions, we make considerations about the goodness of the framework used in this deliverable, but also its limitations, in particular in terms of definition of rationality and generalisation of the results, linked to the specific dataset used to produce the results. We suggest some possibilities regarding how this line of research could be improved and expanded, including some ideas that will be explored in D5.2 We save operational conclusions for D6.2, where the whole concept will be assessed from multiple angles, including the operational one.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

2 Mechanisms and agents

This section presents the experiments that we run in order to estimate the impact of the mechanisms in different environments.

In order to do this, we use Mercury to build a regulation dataset comprising regulation information (duration, number of airlines etc.) and airline information (cost functions). These regulations are all generated for a single airport, Charles de Gaulle, based on the traffic dataset prepared for BEACON (see D2.1 and D2.2 for more information). This dataset is then used in a series of “games” played by the airlines involved in the regulation, that are further explained in this section and the following one.

These games have the following features:

- The agents type: can be honest (giving their best approximation of the real cost function to the central optimiser), rational, bounded etc. We divide the agents in two categories:
    - the main player, i.e. Air France, who is the major airline at this airport volume-wise.
    - the other players, i.e. all the other airlines present at the airport. Note that based on the regulation, not all airlines are present in all regulations.
- The mechanisms themselves, for instance UDPP, by which the final allocation for the regulation is computed.

With this procedure, and by combining the agent types and mechanisms, we are aiming at having a full picture of the mechanism’s impact, from a cost efficiency and equity point of view. Note that this is an ‘isolated’ view, meaning that the knock-on effects of the final allocation is unknown, as well as the exact impact on other performance indicators like passenger delay.

In the following, we sometimes use the terms “one-sided” and “multi-sided” games. By using the former, we usually focus on the behaviour of one player against all the others, considered as background “noise”. For instance, if all players are honest, one might wonder about the effect of introducing a rational agent in the middle (see below for the exact meaning of these terms). In this case, we will consider the experiment as a one-sided, game, with a focus on the distinctive, rational player. On the other hand, if we switch all agents to a rational behaviour, we will speak of a multi-sided game, considering all players equally. This is merely a different point of view on similar situations.

However, it is interesting to note that the full study of these games would be fairly different. For instance, computing the rational, optimal strategy of a player in a single-sided game (where other players do not change their strategy) is fairly easy, even analytically (see section 3). On the other end, in multi-sided game, the return (profit, or avoidance of cost) of a player depends in general on the strategy of other players. This is the domain of game theory, and in general the game must be analysed in terms of Pareto front (if it exists), which is a lot harder, especially in a stochastic game like the one defined here.

In the following, we use the word “rational“ rather abusively, first because we only have an approximation for the single-sided optimal strategy, but also because we use this behaviour afterwards in a multi-sided game, where it might not represent the best strategy. This is merely to illustrate what would happen if everybody used a more aggressive profit-maximisation strategy, even slightly off, as opposed to the honest one.

2.1 Mechanisms

The mechanisms tested for this deliverable have been introduced in deliverables D3.1 and D4.2, see Table 1. They all rely on the same high-level procedure: a hotspot resolution process is started. The NM, or a part of it that in this deliverable we also call central planner or central optimiser, sends the relevant regulation information to the airlines having flights involved in the regulation. The airlines can then take actions, in the form of prioritisations, parameters related to their cost functions, etc, communicated to the NM. The latter will then use this information to compute the final positions of each flight in the queue. Different indicators like the final cost for the airlines are then computed. The exact way the central optimiser solves the regulation depends on the mechanism activated, as explained briefly in Table 1.

Table 1: Mechanism tested in D5.1

Mechanism	Description	Comment
UDPP	Based on their own true cost functions, airlines set priorities to their flights. The NM then 'merges' the priorities to have the final allocation.	Use standard UDPP concepts like Selective Priorities.
UDPP+ISTOP	After applying UDPP, the airlines give some parameters to the NM, used to approximate their cost function. The NM suggests two-airline swaps that are beneficial to everybody. The airlines always accept the suggestions.	The approximation is the same used across different mechanisms, see below.
UDPP+ISTOP_TRUE	Idem above, except the airlines provide the real cost function to the NM. Used for benchmarking.	The efficiency of this mechanism should be between UDPP and NNBOUND.
NNBOUND	The airlines give some parameters to the NM, used to approximate their cost function. The NM then performs an optimisation, seeking the minimum total cost across airlines, while keeping changes of cost above zero for each airline (i.e. nobody can lose from the final allocation).	Low volume users will not be affected by this mechanism, since they cannot lose from it and there is no intra-airline suitable swaps for them.
NNBOUND_TRUE	Idem above except that airlines give their true costs to the NM. Used for benchmarking.	
GLOBAL	The airlines give some parameters to the NM, used to approximate their cost function. The NM then performs an optimisation, seeking the minimum total cost across airlines.	Idem NNBOUND but without the "no-negative gain" constraint.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

GLOBAL_TRUE	Idem above except that airlines give their true costs to the NM.	Best possible outcome from the social point of view (total cost). Equity likely to be very low.
CM	The airlines give some parameters to the NM, used to approximate their cost function. Each airline pays a certain number of virtual credits based on which parameters they chose. The NM then uses the parameters to rebuild cost functions and finds the global optimum (total minimum cost).	Except for the approximation, this mechanism should be one of the best from the efficiency point of view, while achieving equity on the long run. Can be considered as a simple extension of GLOBAL.

Note that, contrary to previous deliverables, we examine different flavours of some mechanisms: indeed, for ISTOP, GLOBAL, and NNBOUND, we want also to test the drop of efficiency of the mechanisms when approximated costs are used instead of real ones. This is crucial because of the credit mechanism, which by design uses approximated costs. The approximation is typically done using single or multi-steps functions. See section 4.1 for the type of functions used for the approximation.

Note also that the auction is not included in the mechanisms tested in this deliverable. It was decided after the results from D4.2 and the human in the loop simulations to drop it. This is due to the complexity of playing the corresponding game, the difficulty to implement it, and the related problem of being able to automatise the resolution. More consideration about this mechanism can be found in D6.2 . While in D4.2 we used a pretty simple procedure to simulate rational and less rational agents, it looks like the full study of this mechanism is beyond the scope of the project. However, some preliminary work has been performed with the auction, and some considerations on rationality in this mechanism can be found in Appendix B.

Finally, note that the only mechanisms that needs calibration is CM. Indeed, all the other mechanisms have no free parameter, on the contrary of CM. More details on these parameters can be found in section 3 and the calibration per se in section 4.3.

2.1.1 Credit mechanism: implementation details

We describe a little bit further the CM in this section. The heart of the mechanism is airlines providing a certain number of parameters to the central optimiser. The central optimiser then rebuilds airline's cost function based on these parameters. As an example, we can take the 'jump2' archetype function, described by two parameters – a margin and a jump. Given these two parameters, the central optimiser will consider the cost of this particular flight as 0 if its delay is smaller than the margin, and equal to the jump value otherwise, effectively using a step function as an approximation to the real cost function of the airline.

Once the central optimiser has collected all the parameters and rebuilt all the cost functions, it uses the GLOBAL algorithm, i.e. it solves for minimum total cost across airlines. On top of that, the airlines spend a certain number of credits based on the parameters that they communicated. For instance, using a value V for a jump will translate into a credit cost of (V-default_jump) x price_jump.

'default_jump' is the "free" jump parameter that the airline can use without spending credits. Any value bigger than that will decrease the number of credits available to the airline, and any value will add the credits. The exact number of credits gained or spent is driven by the 'price_jump' parameter. Note that for each parameter passed to the optimiser, there should be a default value and a price, which should be calibrated properly. This calibration procedure implies to set the values of all these parameters in order to reach the best outcome for the mechanism, both in terms of stability, efficiency, and equity. See section 4 to see how the calibration was performed.

Finally, the airlines usually have access to a certain amount of credits at the beginning of the simulation, and carry their credits from regulation to regulation throughout the simulation. The initial endowment of credits should also be carefully chosen, as is highlighted in section 4.

2.2 Agent types

Different agents will be tested with the mechanisms, in order to see the impact of the mechanisms in different environments. The agent types are described in Table 2.

Table 2: Types of agents.

Agent type	Description	Comment
Random	A random agent provides random information to the NM regarding their cost function, usually based on uniform distribution.	This is only used for testing or calibration.
Honest	Agents are said to be 'honest' when they communicate to the NM either their true costs or the best approximation they have of their true costs.	Honest agents are not rational in the sense that they do not try to have the best allocation for themselves.
Rational	Agents are said to be rational when they communicate costs to the NM designed to minimise their expected cost in the mechanism.	The agents in the model are not fully rational from an economic point of view, because we do not have a closed, exact form for their expectation. Instead we rely on different approximations, see section 4.
Bounded	Short for "agents with bounded rationality", these agents include two "distortions" in their decision-making process, based on prospect theory (PT) and hyperbolic discounting (HD).	Note that bounded-rationality in general is much wider than just a distortion of profit-seeking optimiser's decision making processes, for instance using heuristics, rules of thumbs etc.
Bounded-simple	Due to the complexity of defining a rational agent, the above agent is hard to implement. A more simple bounded-rationality framework was used for some results.	While the agent above is defined as a distorted version of the rational agent, notably with respect to its utility function, this agent is based on the honest agent and uses a

		rule of thumb to approximate the behavioural distortion.

In the model, we sometimes isolate a player from the others, in general the major airline in the Charles de Gaulle regulations, Air France. In this case we can apply a different type for this player than for the others.

2.3 Experiments

We are interested in combining the agent types and different mechanisms in order to study extensively the impact of the latter based on the environment. Table 3 shows a summary of the experiments and main rationale for them. In the table, “NI” stands for “Not Interesting”, (usually a trivial combination) “NA” for “Not Applicable, “FR” for “Future Research” (outside of BEACON scope).

Table 3: Rationales for experiments in D5.1.

Agents\mechanisms	UDPP	UDPP_ISTOP	UDPP_ISTOP_TRUE	NNBOUND	NNBOUND_TRUE	GLOBAL	GLOBAL_TRUE	CM
All Honest	Best baseline for intra-airline optimisation	Comparison with CM	Impact of approximation	Comparison with ISTOP and CM	Impact of approximation	Comparison with ISTOP and CM	Impact of approximation	Comparison with all other mechanisms
All Rational	Idem honest	FR	NA	FR	NA	Impact of gaming	NA	Impact of gaming
All Bounded	NI	FR	NA	FR	NA	Impact of PT	NA	Most realistic assessment of biases. (Note: HD only applied here).

All honest except AF rational	NI	FR	NA	FR	NA	Impact of rationality in single-sided game	NA	Impact of rationality in single-sided game
All honest except AF bounded	NI	FR	NA	FR	NA	Impact of PT and HD in multi-sided games	NA	Impact of biases in single-sided game
All honest except AF random	NI	FR	NA	FR	NA	Studying impact of decisions.	NA	NA (because positive credits reserves cannot be enforced)

In summary, we are performing experiments:

- in order to see the impact of some actions of the agents on the system and on themselves,
- in order to see the impact of rationality and bounded-rationality, either in single-sided games or multi-sided ones.
- in order to see the impact of the function approximation
- in order to compare mechanisms.

Note that for UDPP, having agents that are not honest is not interesting, as least not in our framework. First, rational agents coincide clearly with honest agents in this case[1]. Second, the behavioural biases

[1] : There is a small caveat. Indeed, this mechanism includes flight protection, which may protect a slot for a flight in exchange for another good slot for another airline. In this case, one may imagine that airlines may or may not

Co-funded by the European Union

that we consider should not modify the agent decisions in this case. Indeed, PT is a monotonous modification of the utility function of the agents, which means that preference orders should be preserved when using it, and thus the priority given to each flight. HD does not even have a role in UDPP.

The experiments are all run on a laptop using a python implementation, with a dataset built from SQL. The number of experiments used for extracting a given indicator in a specific case is typically around 100 in the results section.

2.4 Indicators

In this deliverable, we are using mainly two indicators and small variation of them:

The relative drop of **true** cost with respect to FPFS, either overall or broken down by size of airlines. In the following we will refer to it as the **efficiency** of the mechanism[2].

The **equity** of the final allocation across different airlines, itself broken down in two flavours:

- the absolute equity (EQ1)
- the relative equity (EQ2)

To find more details about these two indicators, see D3.1. Here we give the exact definition used in this deliverable:

$$EQ1 = 1 - \sum_i \sum_j abs(c_i - c_j) \Big/ \sum_i \sum_j abs(c_i + c_j)$$

$$EQ2 = 1 - \sum_i \sum_j abs(c_i/n_i - c_j/n_j) \Big/ \sum_i \sum_j abs(c_i/n_i + c_j/n_j)$$

with c_i the drop of cost of flight i with respect to FPFS and n_i the number of flights of the same airline in the regulation.

---

trigger flight protection in order to decrease the profit (increase) of other airlines. Hence, a rational agent would behave slightly differently from an honest one. But first, this effect is likely to be very small, and second, we consider that the objective function of the airline does not include other airlines' costs. This may seem obvious at first sight, but in general in economics, firms try to gain an advantage over their competitors, so a competitor's loss is their gain. In any case, we do not take into account any of that in this deliverable.

[2] Note another definition of the efficiency could be to compare the drop of cost with respect the best case, i.e. with the global algorithm for instance. However, this definition would require to compute the ideal case in every case, which in some case may not be feasible or have any physical meaning (for instance using bounded agents with GLOBAL, see section 3).

3 Agent definition

In this section we discuss the notion of rationality in the specific context of BEACON and we present the analytical work that led to the model implementation, the results of which are presented in section 4.

3.1 Rationality and BE in CM

Rationality in economics has a very strong meaning: it refers to the behaviour of an agent making the best decision it can, given the information it has. The best decision is usually computed thanks to a utility function, sometimes the profit itself when dealing with companies. Indeed, given a utility function u and the space of actions A (or decisions) available to an agent, rationality is defined as taking the action that verifies:

$$a^* = \max_{a \in A} u_e(a)$$

where u_e is the expected gain in utility.

3.1.1 Rationality

Trying to apply this definition to the CM case, for simplicity, we consider first the case where the decision is made on only one parameter, the jump parameter, and where the agent has to make a decision on only one flight. If one knows the probability of having a delay δt as a function of the jump parameter Δj, then the expected utility can be written as:

$$\mathrm{u_e(\Delta j)} = -\int \mathrm{p(\delta t, \Delta j)\, C(\delta t)\, d\delta t},$$

where δt is the delay allocated to the flight, C its cost function, and p the probability of having a delay δt with a parameter Δj. Indeed, the basic assumption is that modifying the parameter will modify the position of the flight in the queue, everything else being constant.

The delicate part is to estimate $\mathrm{p(\delta t, \Delta j)}$, in particular because of the stochasticity of the regulations, see section 4.2. Indeed, it is hard to find a probability function that fits all possible flights in all possible regulations. Here we proceed with a significant approximation, to be able to compute a closed form.

First, we assume that the cost function in reality is a step function, $\mathrm{C(\delta t) = J_0 \theta(\delta t - m)}$, with m the real margin (I.e. based on true cost function). Second, we assume a simplified form for the probability:

$$\mathrm{p(\delta t, \Delta j)} = \frac{1}{A}\, \mathrm{e}^{-\frac{\mathrm{\delta t - \mu(\Delta j)}}{A}}$$

With only the expected delay varying as a function of Δj. A is a free parameter, which drives the variance of the probability function.

This allows for a simple form of the expectation:

$$\mathbf{u_e(\Delta j) = -J_0 \exp\left(-\frac{m - \mu(\Delta j)}{\mathit{A}}\right)}$$

(1)

Note that if μ is a decreasing function, then the expected cost decreases (as expected). Thus, trivially, if the player can set the jump parameter freely, they set the highest one possible (which will be important for section 3.1.2). For the same reason, in a one-shot game, even with credits, the player will set the highest parameter possible available with the amount of credits they have, since the latter cannot be spend in the future. Only when the player knows that there is a future regulation will they spend less than their available credits.

We now introduce the price of parameter p_j, such that the player will spend $\mathrm{c} = \mathrm{p}_j\,\Delta\mathrm{j}$. We also assume that μ is a linear function, and we write the expected utility as:

$$\mathrm{u_e(c)} = -\mathrm{J_0 e^{-(a\,c\,+\,b)}}$$

with $\mu(\mathrm{x}) = -\mathrm{a_1 x} - \mathrm{a_2}$, $\mathrm{a} = \frac{\mathrm{a_1}}{\mathrm{A\,p_j}}$, and $\mathrm{b} = \frac{\mathrm{m+a_2}}{\mathrm{A}}$. The parameters a_1 and a_2 are two additional parameters driving the response of the average delay as a function of Δj.

Assuming that the player is myopic and can only see the next two iterations, they will have to minimise the total expected utility:

$$\mathrm{u_e(c)} = -\mathrm{J_0 e^{-(a\,c\,+\,b)}} - \mathrm{J_0{}' e^{-(a\,(\tilde{c}-c)+\,b)}}$$

Considering a Poisson law for the appearance of the second regulation $\frac{e^{-\frac{\tau}{\lambda}}}{\lambda}$ and a rational discount factor $\mathrm{e}^{-\frac{\tau}{\tau_0}}$ (see deliverable D4.1 for the definition of discount factors), the expectation becomes:

$$\mathrm{u_e(c)} = -\mathrm{J_0 e^{-(a\,c\,+\,b)}} - \int \mathrm{J_0' e^{-(a\,(\tilde{c}-c)+\,b)}}\,\frac{\mathrm{e}^{-\frac{\tau}{\lambda}}}{\lambda}\,\mathrm{e}^{-\frac{\tau}{\tau_0}}\mathrm{d\tau}$$

An optimum exists in general, and is situated in:

$$\mathrm{c}^* = \frac{\tilde{\mathrm{c}}}{2} + \frac{1}{2a}\log\frac{J_0}{J_0{}'} - \frac{1}{2a}\log B$$

where B is the discount factor $\mathrm{B} = \frac{\tau_0}{\tau_0+\lambda}$.

Note that this expression is mostly driven by the ratio J_0/J_0', i.e. the relative importance of the jumps in the first and the second iteration. Note also how, when the iterations are more spaced out (higher λ), c^* increases, i.e. the player tends to put more credits in the first iteration. Likewise, the stronger the discount factor (higher τ_0), the more credits the player will put in the first iteration. Finally, increasing the price of the parameter tends to decrease the amount of credits spent, as expected.

3.1.2 BE version 1 ('bounded' agent)

Using PT and HD introduces two modifications to the above equations. Indeed, instead of using the pure profit (or avoided loss) in the expectation, one can use instead the generalised utility function, or prospect function $u(C) = (-C)^{\alpha}$ (see D4.1 for more details[3] about the prospective function). The expectation is thus:

$$c_e(c) = J_0^{\alpha} e^{-(a\,c\,+\,b)} + \int {J_0'}^{\alpha} e^{-(a\,(\tilde{c}-c)+\,b)} \frac{e^{-\frac{\tau}{\lambda}}}{\lambda} e^{-\frac{\tau}{\tau_0}} d\tau$$

On top of that, we use HD, whereby the exponential discount used previously becomes $\frac{1}{1+k\tau}$. Using both these modifications leads to the following expression:

$$c^* = \frac{\tilde{c}}{2} + \frac{\alpha}{2a} \log \frac{J_0}{J_0{}'} - \frac{1}{2a} \log B'$$

with $B' = e^{-k\lambda} E_1(1/k\lambda)/(k\lambda)$, and E_1 the exponential integral function, computable numerically.

While most previous remarks apply to this expression, we can also notice that the risk aversion α modifies the amount of credits spent. Indeed, the higher the aversion (the smaller α is), the smaller the player will spend.

3.1.3 Application to full games and implementation.

Extending the previous expression to multi-iteration game is not easy. Indeed, while we have only one variable in the previous case (the amount of credits spent in the first iteration), a game with N iterations has in general N-1 variables. While it should be in principle feasible to infer the general expression by assuming stationarity, for this deliverable we keep the myopic assumption, i.e that agents play every time as if there were only another iteration coming after the current one.

The second extension that we need to do relates to multiple flight optimisation. First, we need to notice that the marginal gain in utility decreases with the amount of credits spent, i.e. $\frac{d^2 u_e}{d\Delta j^2} < 0$. This ensures that the airline, in general, does not put all its credits on the most important flight. Indeed, after having spent a certain amount of credits on the most important one (the one with the highest marginal return[4]), it will be better to invest credits in another flight. One could in principle compute the optimal allocation of credits among the different flights (where all final marginal gains are equal), but a closed form is hard to derive. As a consequence, we use a slightly easier procedure, which should yield similar results.

First, we compute the optimal credits spent, as if flights were independent. Then, we compute the total number of credits spent, by average the relevant parameters over all flights (in particular J_0).

[3] We just remind the reader that α in this expression, which smaller than 1, drives the so-called aversion of the decision maker to risk. The smaller the parameter, the more likely they will make a decision where uncertainty is small, to the detriment of the expected profit for instance.

[4] The "marginal return" should be understood in general as the number of euros saved by increasing the amount of credits spent by one unit. However, in the context of utility maximization or, like here, of prospect theory, the marginal return is the increase in the utility/prospective function.

Finally, we divide the credits among flights based on the relative amounts found in the first step. This procedure is equivalent to considering that the airline chooses first the amount of credits that it will spend based on the average importance of its flight, and then dispatches credits among flights based on their relative importance, taking into account non-linearities in the utility function and probability of delay at the same time.

3.1.4 BE version 2 ('bounded-simple')

For reasons explained in section 4.2, we expect the previous implementation of bounded agents to have some limitations, mainly because of the limitation of the underlying rational one. Thus, we decided to use BE, and more specifically PT, in a more direct way, bypassing the need to define a rational agent.

Indeed, the second version of the bounded agent, that we call bounded-simple, is defined with respect to the honest agent, in a similar way that was developed and reported in D4.2. In this case, we start from the simple realisation that the cost function itself can be considered as a kind of expected (anti-) profit itself. In other words, the application of the utility function could be to simply distort the cost function itself. In this case, we do not have a utility-maximisation process afterwards, assuming that the honest behaviour could be an acceptable approximation for the rational one in general. This allows us to avoid heavy approximations in terms of probability computation and long-term rewards. An important drawback is that there is no way to include HD in this framework.

The implementation makes use of the full prospect function in order to distort the cost function. If $C(\delta t)$ is the cost function, then the agent will first compute $u(C(\delta t) - C_{FPFS})$, where C_{FPFS} is the cost of the flight in the FPFS queue, taken as the point of reference, and where the utility is:

$$\begin{aligned} u(x) &= x^{\alpha} \; if \; x > 0 \\ &= \lambda'(-x)^{\alpha} \quad otherwise \end{aligned}$$

The function is then offset to be always positive (because of next step). Finally, the approximation process is applied to this new function, with the margin and jump communicated normally to the central optimiser.

3.2 Rationality and BE in GLOBAL

3.2.1 Rationality

The rationality for an agent playing a game with the GLOBAL algorithm is easy to deduce from the CM case. Indeed, the only difference between both games is the fact that the player pays for the parameters in the latter, and not in the former. Using equation (1), it is easy to see that increasing Δj will always increase the expected utility. As a consequence, the rational behaviour in this case is to set Δj to its maximum value (which is capped in the simulations).

3.2.2 BE version 1 ('bounded' agent)

Defining BE as a distortion of the rational case seems impossible in this case. Indeed, the optimal action being to maximise the jump parameter, there does not seem to be any room for behavioural biases, at least not from the PT and HD discount point of view. As a consequence, we only use the simple version of the bounded agent with GLOBAL, see next section.

3.2.3 BE version 2 ('bounded-simple' agent)

While we cannot rely on the rational case to define the BE one, we can use the honest one, which is well defined and not singular for GLOBAL. In fact, we use the exact same procedure as in the CM case, modifying the cost function with the prospect function, running the fitting procedure, and then communicating the parameters for the central optimiser. The only difference is the agent does not pay for the parameters like in the CM case.

4 Results

The aim of this section is to present what happens when we implement the agents described in sections 2.2 and 3 and let them play the games defined in section 2.1, measuring the indicators presented in 2.4, using the dataset described at the beginning of section 2. However, before presenting the results per se, we explore the agents and the simulation process, calibrating the model in the process. We also note that the main objective is to assess the efficiency of the new BEACON mechanisms, ISTOP and CM, the other mechanisms tested being here for comparison and benchmarking.

We start by having a look at the approximation process mentioned in section 2.1. The impact of this process (linked to how well the approximated cost functions represent the real costs of the airlines) has to be understood in particular with regards to the CM, which relies at its heart on such an approximation.

We then move on in section 4.2 to explore the notion of rationality and BE in the GLOBAL and CM mechanisms. More specifically, we want to explore the impact of the decision on agent on its final cost, and how the difficulty to capture this relationship impacts the definition of rationality.

Section 4.3 shows how the CM – the only mechanism with free parameters – was calibrated. This comprises also the calibration of the rational agent playing the CM, using results from section 4.2.

Once the model and the agents are calibrated, we can study how rationality impacts the efficiency of the mechanism, or more specifically how well the latter works when honest or rational agents play the corresponding games. This is what we do for GLOBAL and CM, the only mechanisms for which a rational player has been defined.

The next step is to study the impact of BE and how the corresponding agents fare in the corresponding games, which is done in section 4.5, in a similar way than what was done for rationality.

Finally, we perform a high-level comparison between all mechanisms in section 4.6. Using different kinds of agents defined for each mechanism, we show how efficient and how equitable the latter can be expected to be.

4.1 Impact of cost function approximation process

First, we are examining how the cost function approximation process impacts the mechanisms that use it. Indeed, as explained in section 2, for some of the mechanisms, we defined two distinct flavours, one where the central optimiser used approximated cost function, and one where they used real ones. The main reason for that is to see the impact of the approximation process on CM, which by design uses approximated functions.

Indeed, at the heart of the credit mechanism, there is the idea that if a flight puts more pressure on the system, the corresponding airline should pay for it somehow. This is required in order to avoid having airlines inflating their cost in order to have a better allocation for their flights. The way it was abstracted in BEACON originally was through the use of two paying parameters representing the features of the actual cost function. If increasing one of the parameters would likely lead to a better allocation for the flight (all other things equal), then the airline should pay for it. It was decided to use a virtual currency, that we call credits, to pay for this.

Hence, by design, the credit mechanism uses an approximation of the cost function to find the best allocation. This design was decided on in early experiments with fairly simple cost functions for the flights, and was found to be reasonable with two parameters, a ‘margin’ and a ‘jump’. In particular, the functions used for D4.2 were quite simple and easily approximated with such a process. The approximated cost function is then defined as being null if the slot of the flight implies a delay smaller than the margin, and equal to the jump value otherwise. Margins are thus defined in minutes and jumps in euros. See Table 4 for an illustration of this function ('jump2').

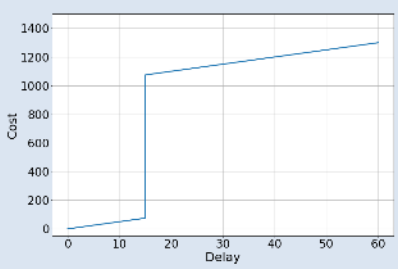
In this deliverable, we are more interested in looking at more realistic cost function, as produced by the simulator Mercury and its advanced cost structure capabilities. These cost functions are typically more complex, due to the interplay of passenger connections and aircraft turnaround, as shown in Figure 1.

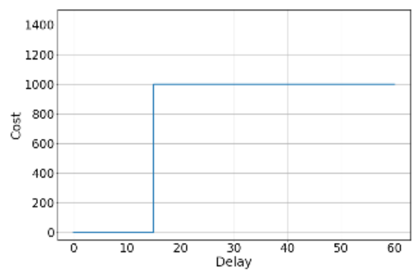
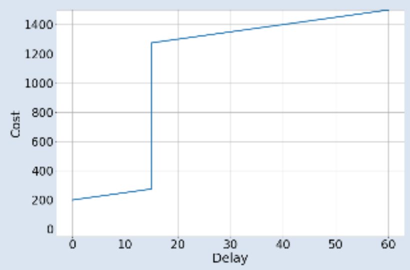
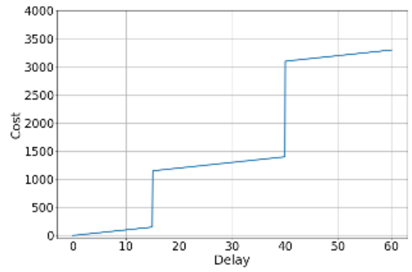
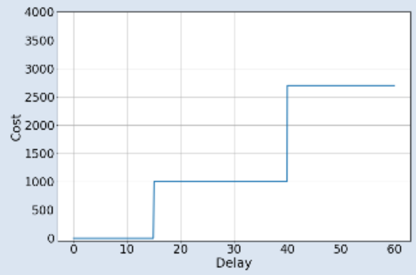
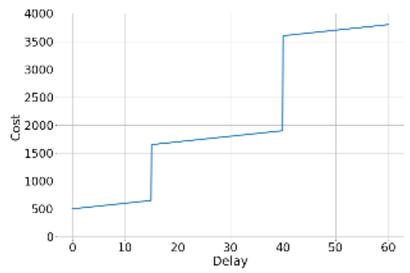
Figure 1: Examples of realistic cost functions taken from Mercury.

It is thus natural to look at how well the adopted step functions can approximate these cost functions, and, more importantly, how they impact the mechanisms. Indeed, the main point of the mechanism is to minimise declared costs. If the declared costs (based on step approximations) are far from the real cost functions, the mechanism will not work as intended. For instance, the ISTOP mechanism may suggest a swap between four aircraft (in two different airlines) that it thinks will decrease the cost for both airlines. However, it might not be true because of a poor approximation. Note that in the model, the suggestions from ISTOP are always accepted by the airlines, as opposed to how the mechanism is intended to work in reality – should it be implemented, where airlines can refuse the swap. Hence, in the first case, we shall see a drop in the efficiency in the mechanism (i.e. less cost saved), while in a the second case the suggestions will simply be discarded. However, even in the second case, beneficial swaps may still not be suggested because of the inferior approximation.

In order to explore this issue, we consider several approximations for the cost function (called “archetypes” in the following text), summarised in Table 4.

Table 4: Archetype functions for approximation.

Archetype function	Definition	Illustration
Jump	Defined by parameters: slope a, margin m, jump j: $c(x) = ax \quad if\ x < m$ $\quad\quad = ax + j \quad otherwise$	

Jump2	Defined by parameters: margin m, jump j: $c(x) = \begin{cases} 0 & \text{if } x < m \\ j & \text{otherwise} \end{cases}$	
Jump3	Defined by parameters: slope a, margin m, jump j, offset A: $c(x) = \begin{cases} A + ax & \text{if } x < m \\ A + ax + j & \text{otherwise} \end{cases}$	
Double jump	Defined by parameters: slope a, margin m_1, jump j_1, margin m_2, jump j_2: $c(x) = \begin{cases} ax & \text{if } x < m_1 \\ ax + j_1 & \text{if } m_1 < x < m_2 \\ ax + j_1 + j_2 & \text{if } x > m_2 \end{cases}$	
Double jump2	Defined by parameters: margin m_1, jump j_1, margin m_2, jump j_2: $c(x) = \begin{cases} 0 & \text{if } x < m_1 \\ j_1 & \text{if } m_1 < x < m_2 \\ j_1 + j_2 & \text{if } x > m_2 \end{cases}$	
Double jump3	Defined by parameters: slope a, margin m_1, jump j_1, margin m_2, jump j_2, offset A: $c(x) = \begin{cases} A + ax & \text{if } x < m_1 \\ A + ax + j_1 & \text{if } m_1 < x < m_2 \\ A + ax + j_1 + j_2 & \text{if } x > m_2 \end{cases}$	

In order to estimate how appropriate the archetypes are, we use the following procedure:

- we sample regulations coming from the dataset we built previously,
- we extract the cost functions of all airlines involved in the regulation,
- we fit each function using each of the previous archetypes, one after the other,
- we compute the coefficient of determination R2for each regression[5],
- we aggregate the R2 coefficients, defining:
  - the number of regressions with a positive R2,
  - the average R2 on these regressions.

The results are shown in Figure 2.

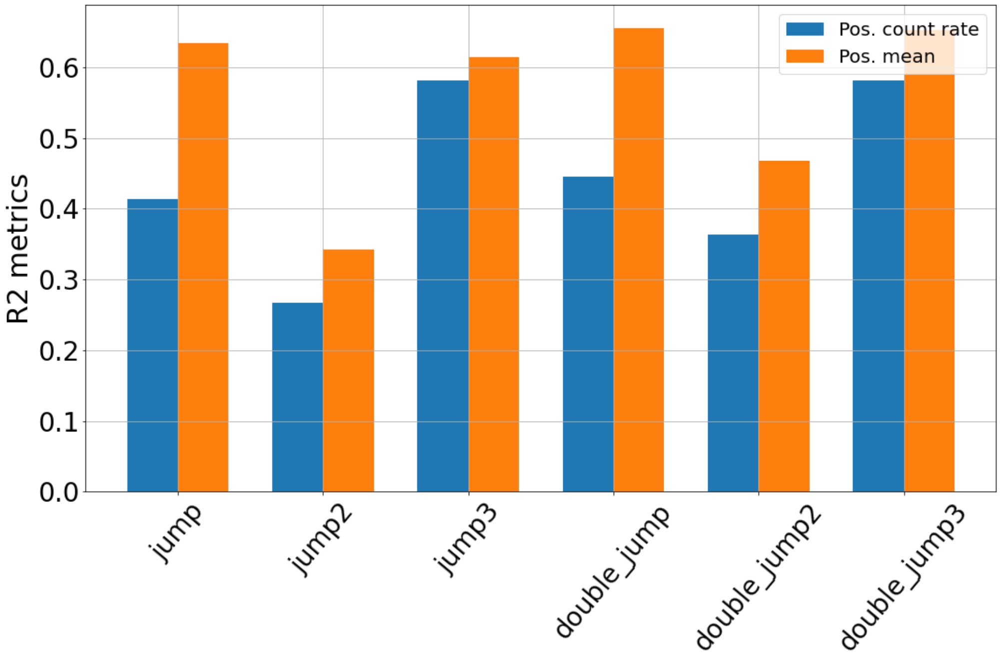


Figure 2: Goodness of fit when regressing cost functions with different archetype functions.

A first conclusion from this figure is that none of the proposed functional forms are excellent at approximating the real cost functions, unfortunately (as can be seen from the relatively low R2). It seems that these realistic cost functions are much harder to capture using a small number of parameters, at least using these archetype functions. On the contrary, during the first stage of the

[5] The R^2 coefficient is defined as: $R^2 = 1 - \sum_i (y_i - f_i)^2 / \sum_i (y_i - \bar{y})^2$, with $y_i$the experimental points, f_i the predicted points, and $\bar{y}$ the mean of the experimental points. It measures how well do the predicted points coming out of a model match the experimental ones.

project, we used cost functions that were a lot simpler and thus easier to approximate. Only using the "jump2" archetype yielded satisfactory results for these. A second conclusion is that including the slope in the functional form seems to significantly improve the regressions. Including an offset improves them even more. Finally, there is no notable benefit in using two steps instead of one.

A natural conclusion would then be to use jump or jump3 instead of the "jump2" one, as initially planned. However, this implies to add one or two prices for each new parameter, and the experiments to be performed in this deliverable (and the next one D5.2) would be fairly complex because of this. We thus choose to use a compromise: all agents use the "jump3" functional form, but the slope and the offset are always given "honestly". By 'honestly' we mean that the values given are the ones obtained out of the approximation process, i.e. the ones we are fitting the real cost function the best. We have also further considerations on this approximation issue in the conclusions.

The impact of the approximation on the mechanisms themselves are also crucial. In Figure 3, we show the efficiency of different mechanisms, in particular using true cost functions and approximated ones (all agents are honest in these simulations).

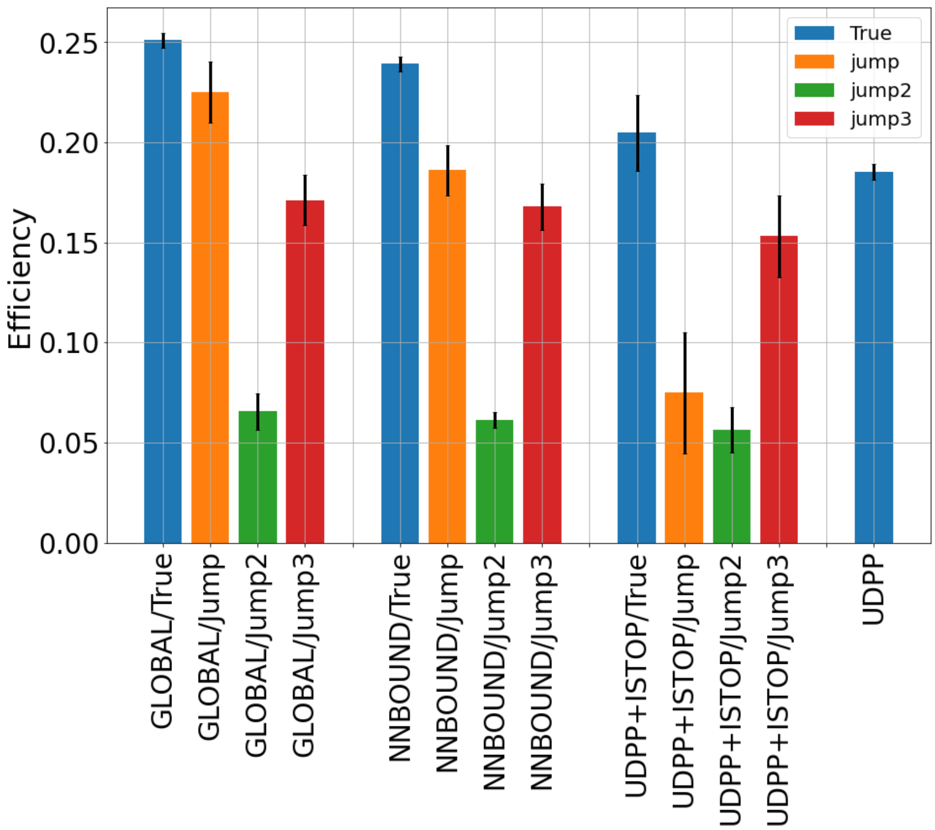


Figure 3: Efficiency of mechanisms with different approximations for the cost functions as well as with the true cost functions.

As clearly seen in the figure, the approximation has a big impact on the efficiency. Using jump2 drops the efficiency of GLOBAL and NNBOUND by a factor 5, and makes UDPP+ISTOP **less** efficient than UDPP alone, highlighting the *destructive power of erroneous information* on the system. 'Jump3' seems to be faring better, allowing to have a fairly smaller drop in efficiency. In particular, the approximation in

ISTOP does not completely destroy the UDPP gains, even though it is clearly a drawback to use in these conditions. The 'jump' approximation seems to provide the best results among all the functional forms except ISTOP, for a yet unknown reason. It is also striking to note that UDPP is almost as efficient as GLOBAL or NNBOUND with this approximation. This is an important point that we will discuss in the conclusions.

Finally, we can note that GLOBAL, even in its most perfect form, is barely better than NNBOUND. Indeed, we are expected the low-volume users to be impacted negatively in GLOBAL, because they cannot gain anything. Hence, the difference between GLOBAL and NNBOUND should come mainly from how the low-volume users are treated. This indicates probably that the global efficiency is not highly dependent on the low-volume users, as least in this case.

4.2 Impact of decisions in GLOBAL and CM

We are then interested at looking at how the agent decisions impact the mechanisms. By agent decisions, we specifically refer to the values the airlines transfer to the NM to rebuild the approximated cost function, e.g. the jump parameter.

4.2.1 One regulation, one specific flight

The first experiment we run is to see how the parameters given by the airlines actually influence their final allocation slot. It is natural to think for instance that, all other things being equal, increasing the jump value of a given flight will lead to a better slot for it. Indeed, the optimiser will realise that it is more fruitful to try to put this flight early in the queue in order to avoid the (declared) jump and its cost for the objective function. This is the reason why the airline has to pay for such an increase in the CM.

In order to explore this issue, we consider a setup where the main player (AFR) plays at random, while all the other agents are honest. We use the GLOBAL mechanism in order to see better the impact of different decisions. We then record how much delay each flight got in the final allocation (with respect to FPFS) against the magnitude of the change in the parameter.

We consider first a single instance of regulation, used in multiple simulations. Here, we ask the player only to modify at random one of the parameters, margin or jump, of their first flight. Figure 4 shows the cost as a function of the relative jump increase. The plot also includes an average made on 15 quantiles.

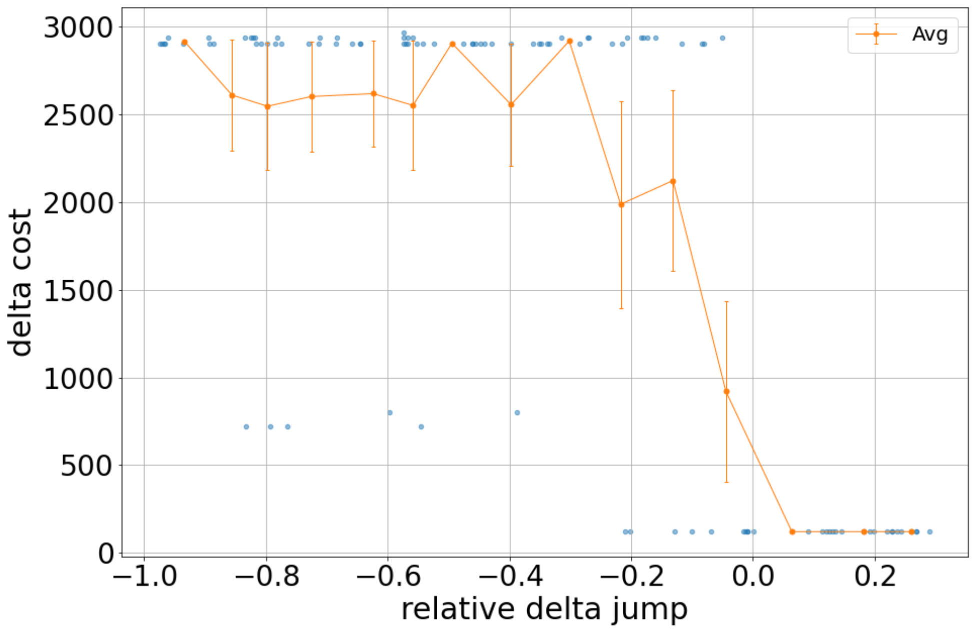


Figure 4: Evolution of drop of cost as a function of the difference between the jump parameter chosen by the agent and the default jump parameter. The orange line is an average on 15 quantiles, with corresponding standard errors.

As expected, increasing the size of the jump increases the likelihood a having smaller delays. One can note also the threshold effect, which should be expected as well: as soon as the cost is higher than some other flight competing for the same slots, this flight is prioritised higher and the cost drops.

Figure 5 shows a similar plot for the margin instead. The picture is more complex, with a non-monotonous behaviour. In fact, by setting a very low margin (say, 0), the flight becomes insensitive (as far as the **declared** cost is concerned) to the actual slot it is allocated to. Hence, the optimiser should put it in last position, and thus the cost is very high. When the margin increases, the optimiser sees an opportunity to put it early in the queue, where its declared cost is small. This happens roughly just before the honest margin is hit, as expected.

After that, the margin is bigger than the actual one, and thus the optimiser sometimes allocates it to a slot that is actually more expensive for the airline. Hence the real cost increases again with the declared margin.

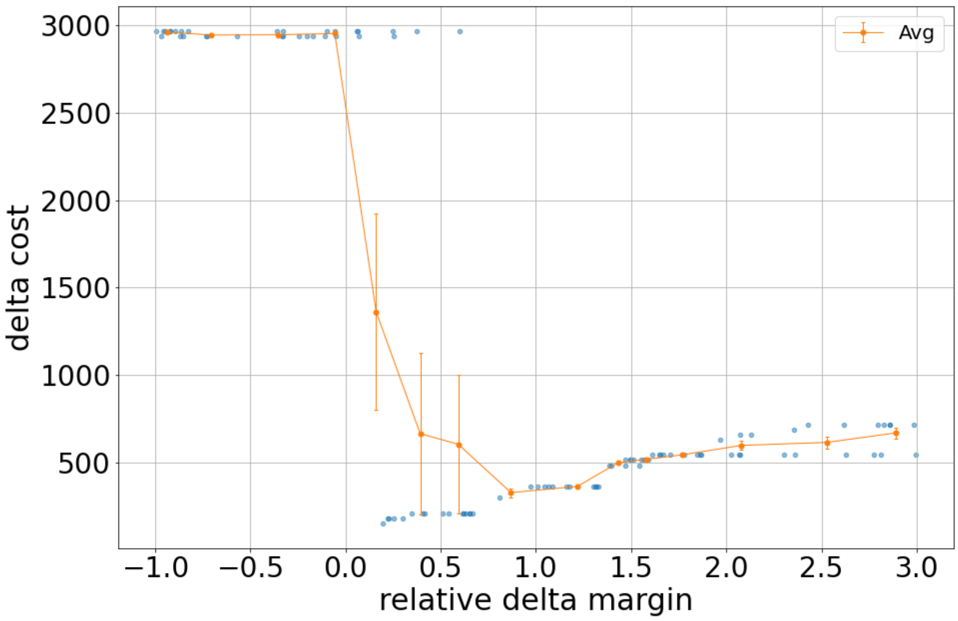


Figure 5: Evolution of drop of cost as a function of the difference between the margin parameter chosen by the agent and the default margin parameter. The orange line is an average on 15 quantiles, with corresponding standard errors.

The conclusion to draw from these figures is that for a given regulation and a given flight, it's quite easy to see the impact of changing the jump parameter on the system[6], even if some noise is present. Hence, a rational agent that would like to increase its profit (minimise its cost) should increase the jump parameter, for instance. We also note that the non-monotonous relationship for delay and margin complicates a lot the picture, since airlines would need to have more information on the flight in order to make the correct decision, as opposed to blindly decreasing the margin in the hope of having a smaller cost.

4.2.2 One regulation, one random flight

While the impact of a decision is pretty clear on a specific flight in a specific regulation, we are now interested in looking at the impact of a decision on a random flight, while still keeping the same regulation throughout each iteration of the game.

The results are presented in Figure 6 and Figure 7. On the contrary of the previous case, the effect of the decision is barely visible, both for the jump and the margin. The correlation coefficient is indeed

[6] The correlation coefficients are respectively -0.27 and -0.14 (both significant) for delay and cost.

smaller than the last case, with -0.11 for the jump parameter for instance, still statistically significant nonetheless. Note also how the shape for the margin seems more monotonous in this case.

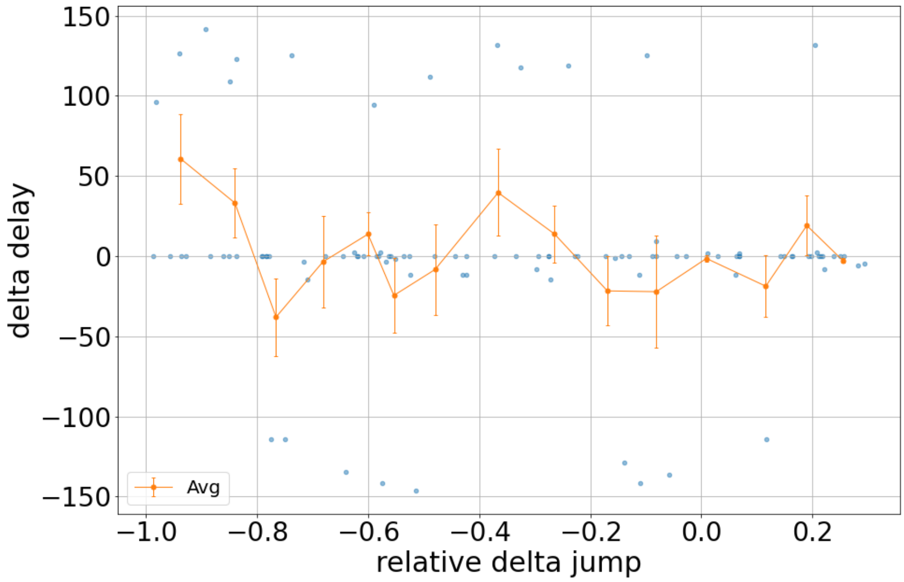


Figure 6: Evolution of delay with respect to FPFS as a function of the difference between the jump parameter chosen by the agent and the default jump parameter. The orange line is an average on 15 quantiles, with corresponding standard errors.

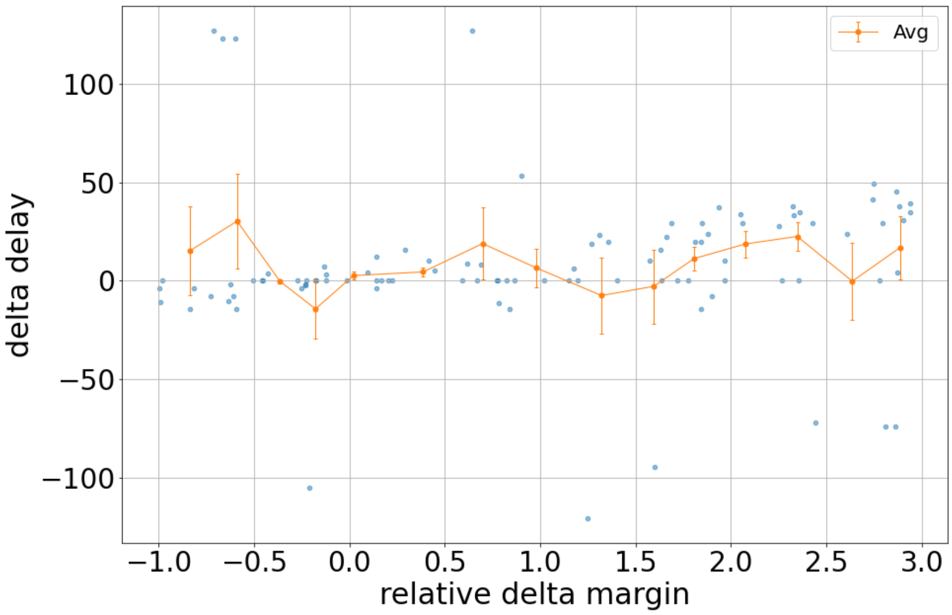


Figure 7: Evolution of delay with respect to FPFS as a function of the difference between the margin parameter chosen by the agent and the default margin parameter. The orange line is an average on 15 quantiles, with corresponding standard errors.

4.2.3 Random regulation, random flight

Finally, we are looking at the impact of a decision on a random flight in a random regulation, thanks to Figure 8, where we randomised the regulation, the flight on which a random decision was applied, and the decision itself. In this case, one can barely see a trend. The correlation coefficient in -0.038 and, while still statistically significant, shows almost no link between the decision and the final delay assigned to the flight.

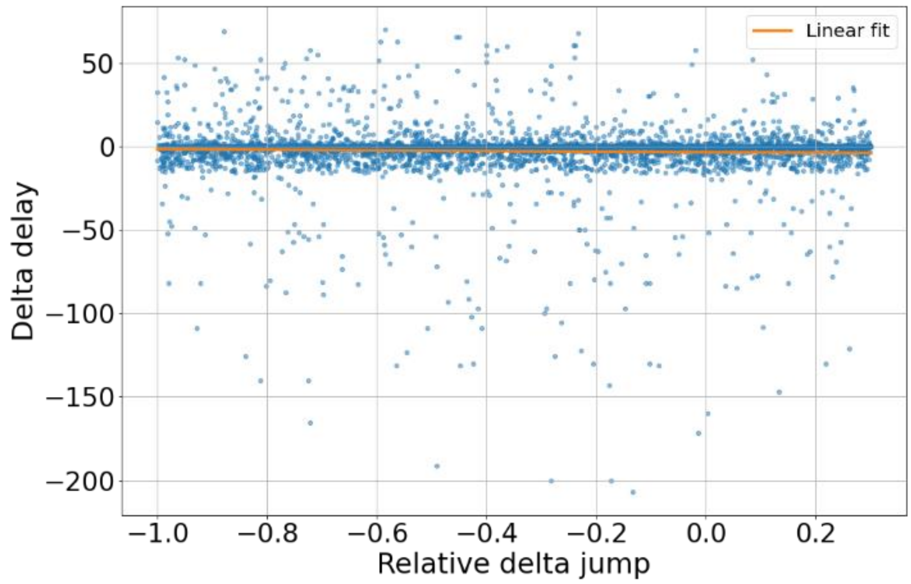


Figure 8: Evolution of delay with respect to FPFS as a function of the difference between the jump parameter chosen by the agent and the default jump parameter. The orange line is an average on 15 quantiles, with corresponding standard errors.

This simple fact is an important one and has several consequences:

- Theoretically, the definition of a rational behaviour assumes that the agent is able to know which decision it should take, i.e. it knows the relationship between its decision and its consequences. The previous figure shows that this mapping is particularly noisy at an aggregated level, and thus a rational agent would struggle to use this mapping to make significant gains.
- The difficulty to define a rational agent has also a direct consequence on the possibility to define a bounded agent, since the latter is a distorted version of the former (at least when using HD and PT). Hence, in section 2 and 3 we introduced a simpler version of the bounded agent, not based on the rational agent.
- The difficulty of defining a rational agent is directly linked to the level of information given to the airline. Indeed, one can introduce more information (features in ML language) that would help make the mapping between the delay and the decision, for instance the absolute position of the flight in the FPFS queue. Other information should however be kept hidden from the player, for instance the cost functions of the other airlines. The natural framework to compute a better rational agent is reinforcement learning, a powerful tool to harness decision-making processes in iterated games, especially in stochastic environments. We go back to this in the conclusions.
- Practically, it also means that real players would struggle to find a good strategy to make gains beyond UDPP. We will discuss more in detail the consequences of this point in conclusions.

4.3 Calibration of CM

In this section, we are interested in looking at how the credit mechanism works and how its parameters should be set. For simplicity, we explore a particular version of the game where only the jump parameter is "played", i.e. that the agents are honest in all cases on all parameters, which are free, except for the jump parameter.

Given this simplified game, the parameters that are interesting to explore are the following:

- the default jump, i.e the "free" jump,
- the price of the jump,
- the initial number of credits,

On top of that, we are going to explore a simple parameter, called "reinjection", linked to 'quasi-honest' agents, and we show how we calibrated the rational agent for CM.

An important difference between the CM and the other mechanisms is the fact that there is a 'memory' of the agent from one iteration to the other because of the fact that credits are carried out. Hence, when we perform 100 iterations for instance, these are not 100 independent ones, since at the beginning of each iteration, the agents have the amount of credits they used to have at the end of the previous. This is in contrast with all other mechanisms, where agents have a blank slate at the beginning of each iteration.

A consequence for CM is that one has to study the stationarity of the game, i.e. state variables (here, credits) depends on the iteration index. For this game, we have encountered mostly three cases:

- Absorptive: the number of credits goes to 0 after some iterations. It can go up again later due to some small regulations for instance, but always goes back to a very small value.
- Stable: in this case, the number of credits is fairly stable throughout iterations.
- Explosive: in this case agents have more and more credits, going to infinity when the number of iterations is big.

Note that we did not really explore other cases (notably periodic, chaotic), even though a more thorough study should be performed. Also, note that some transient regimes sometimes exist in the first few iterations, before the stabilisation happens.

4.3.1 Default jump parameter

The default jump parameter is crucial because it conditions the inflation or deflation of credits, at least with honest agents. Indeed, with the honest agents, the jump and margin parameters communicated to the central optimiser are the ones coming from the regression. If the default parameter is too high, airlines will thus make credits automatically, when the regression gives a parameter higher than the default one (leading to explosive regimes). If it is too low, the airlines will not have enough credits to pay for the ideal parameter from the regression (leading to absorptive regimes). We are thus aiming at a middle ground, where airlines have quite a stable amount of credits (stable regime). Figure 9 shows the number of credits of the main player and the other airlines (all honest), with a generous default parameter (3000 euros). In this case, the main player sees its number of credits increase indefinitely, because the average honest jump is obviously smaller than the default one.

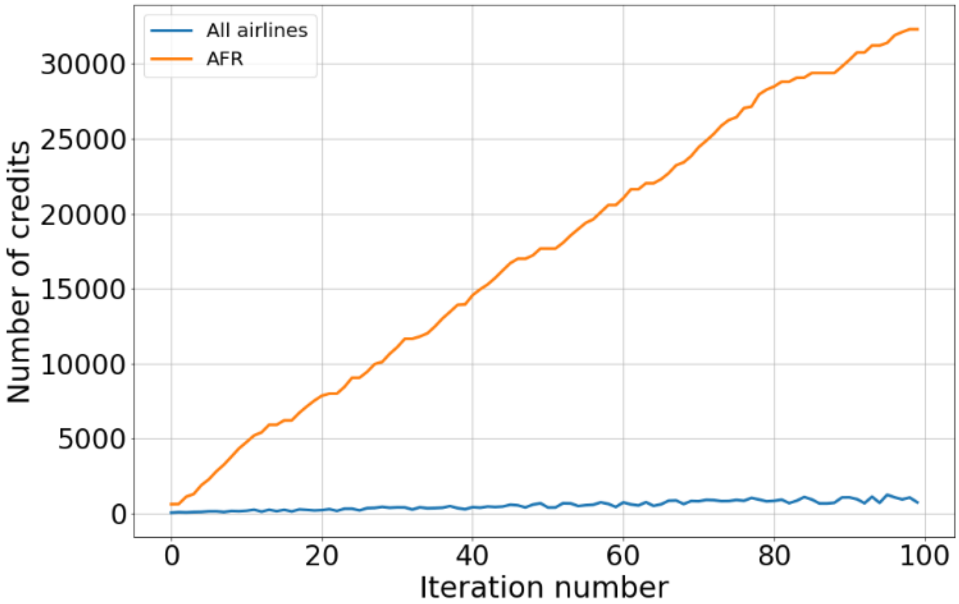


Figure 9: Evolution of credits throughout iterations, average for all airlines (blue) and for AFR (orange).

Measuring the credits of AFR on the last 50 iterations while sweeping the default jump parameter allows us to have a more systematic picture of the effect of the parameter, as shown in Figure 9. In this figure, we also show the slope of a regression performed on the last 50 iterations, in order to see the stationarity of the credits, as well as the standard deviation on the last 50 iterations, in order to see the variance of the process[7].

[7] Some plots illustrating the transition between the different regimes can also be found in Appendix A.

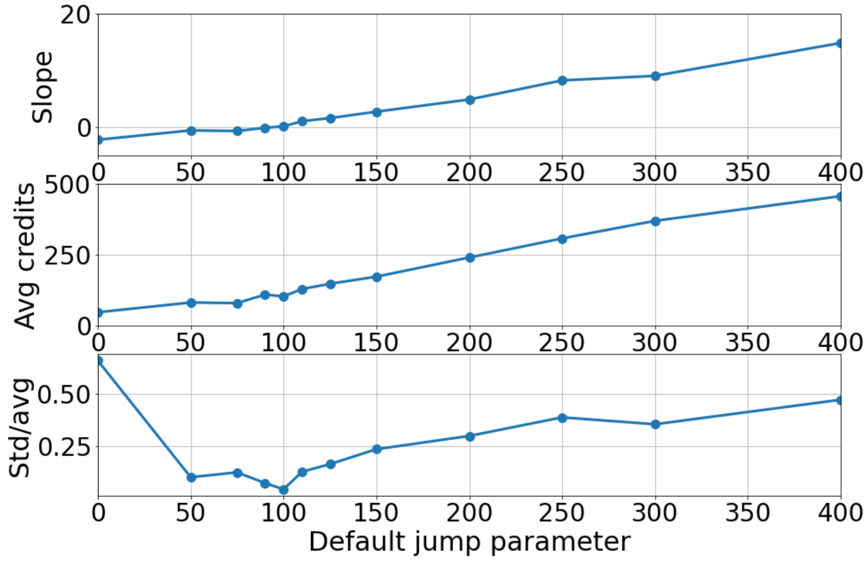


Figure 10: Top: slope obtained out of a regression on the credits in the last 50 iterations of the game. Middle: average credits of AFR during the last 50 iterations. Bottom: Standard deviation over average on last 50 iterations.

The conclusion from Figure 10 seems to be that 90 euros as a default parameter seems to be a good choice, allowing to have a stable process without compressing credit fluctuations too much. An example of the evolution of the number of credits with this parameter is displayed in Figure 11.

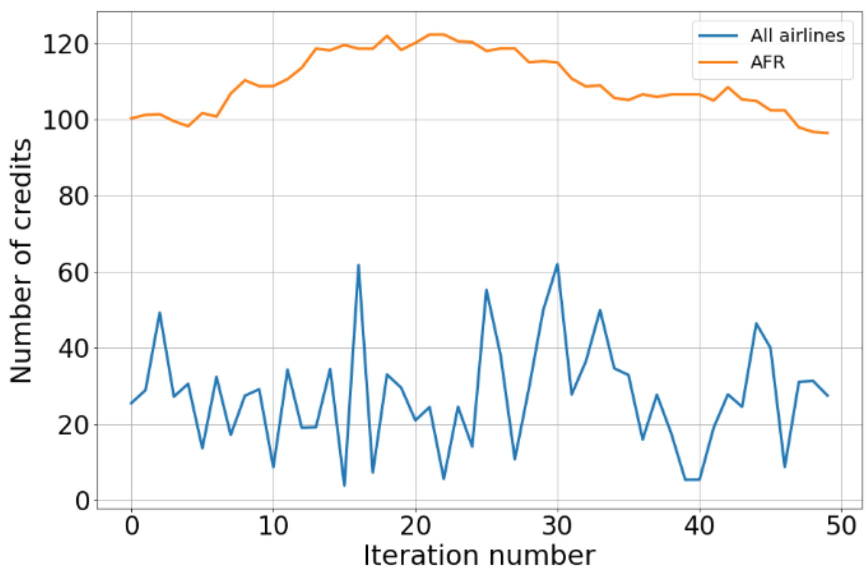


Figure 11: evolution of the number of credits for all airlines (blue) and AFR (orange) on 50 iterations for a default jump parameter equal to 90 euros.

4.3.2 Price for jump parameter

The price that the airline has to pay for deviating from the default parameter is in general a crucial parameter, since it modifies the value of credits in real money. However, when only one parameter has to be paid, the price is actually only a rescaling factor, at least after the transient regime due to the initial credits has passed. Figure 12 illustrates this fact, with three different runs displaying similar behaviour, up to a scaling factor.

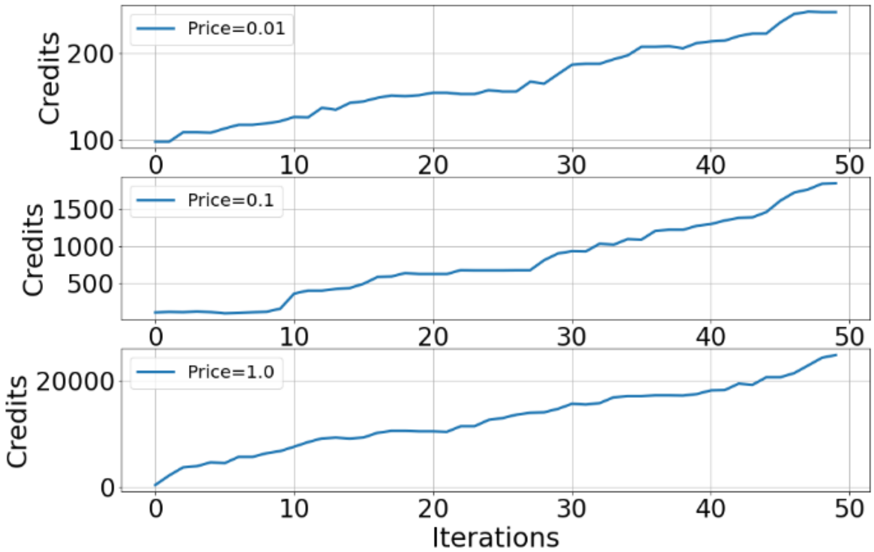


Figure 12: Evolution of credits for AFR for three values of the jump price: 0.01 (top), 0.1 (middle) and 1. (bottom).

Hence, we will fix the price to 0.01 in the following, similarly to what was done with the human in the loop simulations.

4.3.3 Initial number of credits

The initial number of credits should come into play either only in the transient regime of the evolution of credits during the simulation, if the credits are in an unstable regime, or simply offset the level of credits in the stationary regime. In the latter, that we selected previously thanks to the default jump parameter, the agents tend to have a stable number of credits, spending roughly as much as they gain. In Figure 13 we show what happens when we endow 0, 200, and 500 credits to all agents before the first iteration.

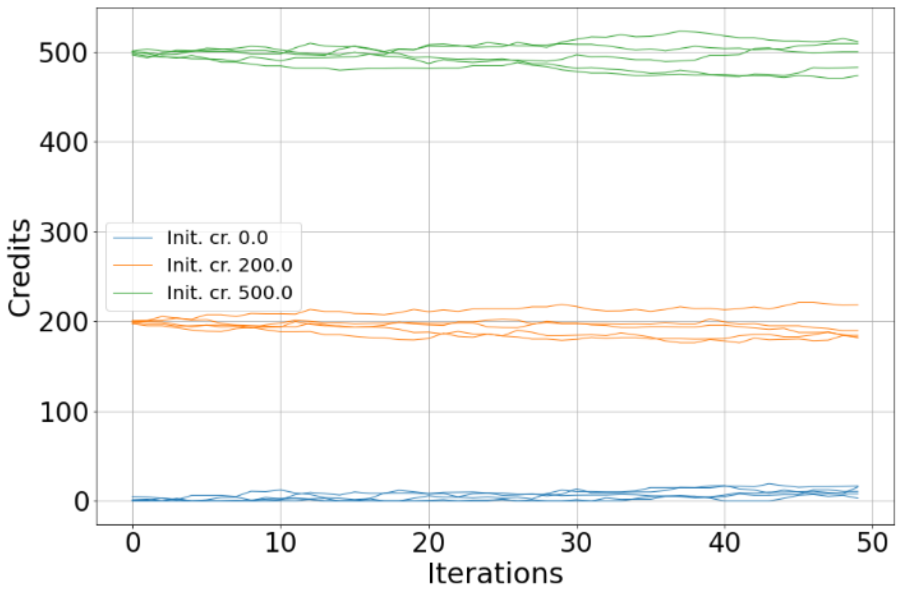


Figure 13: Evolution of credits in different runs for different initial credit endowments (0, 200, and 500).

The initial number of credits being largely irrelevant in the stable regime, as seen thanks to the previous figure, we choose to set to 0 and let airlines make gains in initial regulations before spending them in subsequent ones. This also avoids raising the question of equity in the number of credits among airlines. Indeed, it could be argued that bigger airlines would need a higher number of initial credits to function properly.

4.3.4 Reinjection parameter for honest agent

Another parameter that was introduced was the 'reinjection', whereby an honest (in practice quasi-honest) agent may spend more credits than what would be done with the purely honest rule. For instance, if the parameter is 0.5, it means that, on top of the credits spent for its preferred parameters (the ones coming from the regression in the approximation process), an agent will spend 50% of its remaining credits. This parameter was introduced as a mean to have a more stable regime in credits. Its impact can be seen in Figure 14, where we display the same kind of metrics as in Figure 10, with a reinjection parameter of 0.5.

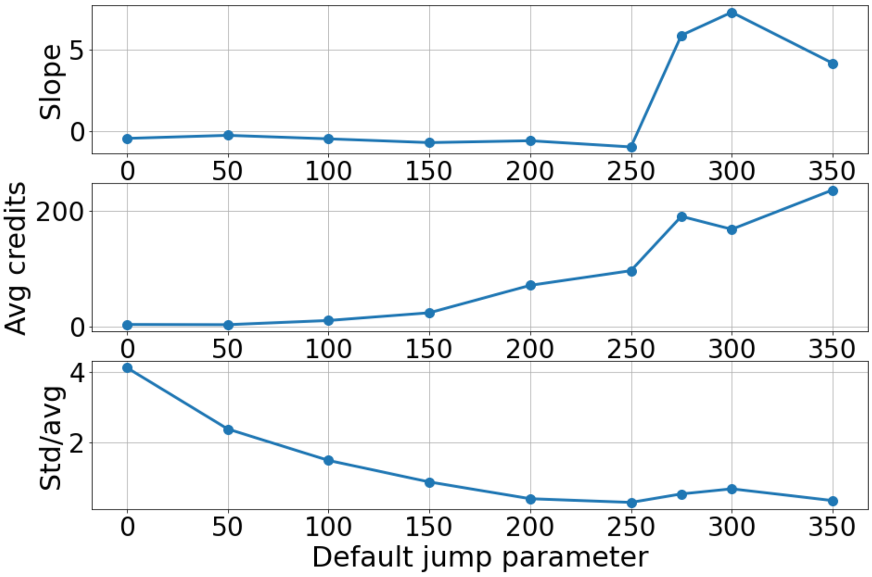


Figure 14: Top: slope obtained out of a regression on the credits in the last 50 iterations of the game. Middle: average credits of AFR during the last 50 iterations. Bottom: Standard deviation over average on last 50 iterations.

It looks, however, that this parameter makes it more difficult to have a stable regime, as can be seen with the fact the fitted slope goes from negative to positive in a very short range of the default jump parameter. Even if this parameter may be of interest in designing a bounded agent, we chose to keep the value of this parameter 0 in the following.

4.3.5 Calibration of rational agent

As explained in section 3, the rational agent for CM needs to be calibrated. In particular, the relationship between average gain in cost and jump parameter modification has to be performed (done thanks to Figure 8). Another important calibration is the shape of the delay probability distribution, modelled as an exponential in section 3. Figure 15 shows the empirical distribution obtained with the experiments from section 4.2.3, with an exponential function regressed on the data. The agreement of the fitted function $f(x) = Ae^{-x/\lambda}$ with data is pretty good, with optimal scaling (A) and time (λ) parameters equal to 0.146 and 4.73, respectively.

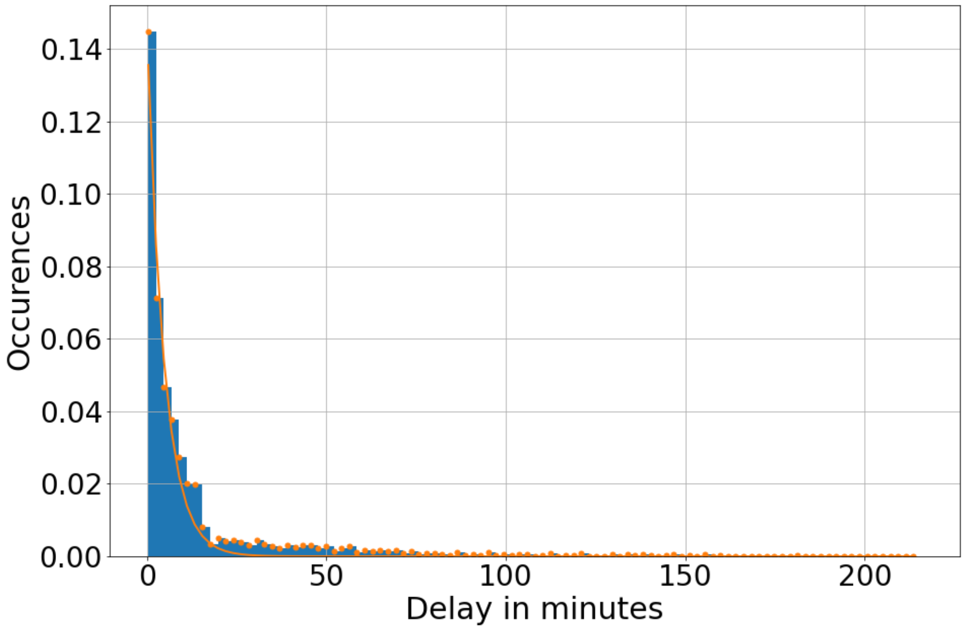


Figure 15: Distribution of delay.

4.4 Impact of rationality in GLOBAL and CM

We are interested in this section in the impact of having a more rational agent, as opposed to an honest agent that communicates its true cost, in particular for the credit mechanism.

As seen previously, defining a rational agent is a hard task for the CM, because of the difficulty of the mapping between decisions and expected results of these decisions. This is partly due to the nature of the game, in particular its stochasticity, and even more prominently due to the approximation process done before the game, which adds a level of noise difficult to overcome for a rational agent. Hence, while we are expecting a rational agent to be better than an honest one, it will not always be the case in the following. We are still interested in seeing 1) how multi-sided games – where all agents are rational – compare with single-sided ones – where only one player is rational – and 2) see the expected drop of efficiency – if any – in the mechanism when introducing bounded agents.

In this section, we are interested mainly in the efficiency of the mechanism, but keeping the metrics for AFR and the other players distinct in order to see the differential impact of the mechanism on these companies.

4.4.1 In single-sided games

First, we are interested in single-sided games, where we isolate a single player, which will be either honest or rational (and later bounded). All other agents act as honest agents, with the calibration explained in section 4.3. The agent we choose is AFR, the main company in general in the set of regulations we are using. We are using them as agents to isolate a higher impact of the rational behaviour.

Figure 16 shows the efficiency for all agents, AFR, and the other agents, where using a honest or a rational agent for AFR with the GLOBAL mechanism. A first thing to note is how AFR has naturally a much higher efficiency than small companies, even when companies are honest. This is expected, since AFR has a lot more flights than its competitors in the regulation set used, thus potentially having the highest potential gains in terms of costs avoided.

The results are very clear: while the efficiency of AFR is increased, the efficiency of other agents drops to 0. This is completely to be expected. The 'rational' agent in this case will maximise its jump parameter, in order to make its flight seems more important than they are in reality (and more important than its competitors). As a result, the optimiser prioritises AFR flights, to the detriment of all other flights.

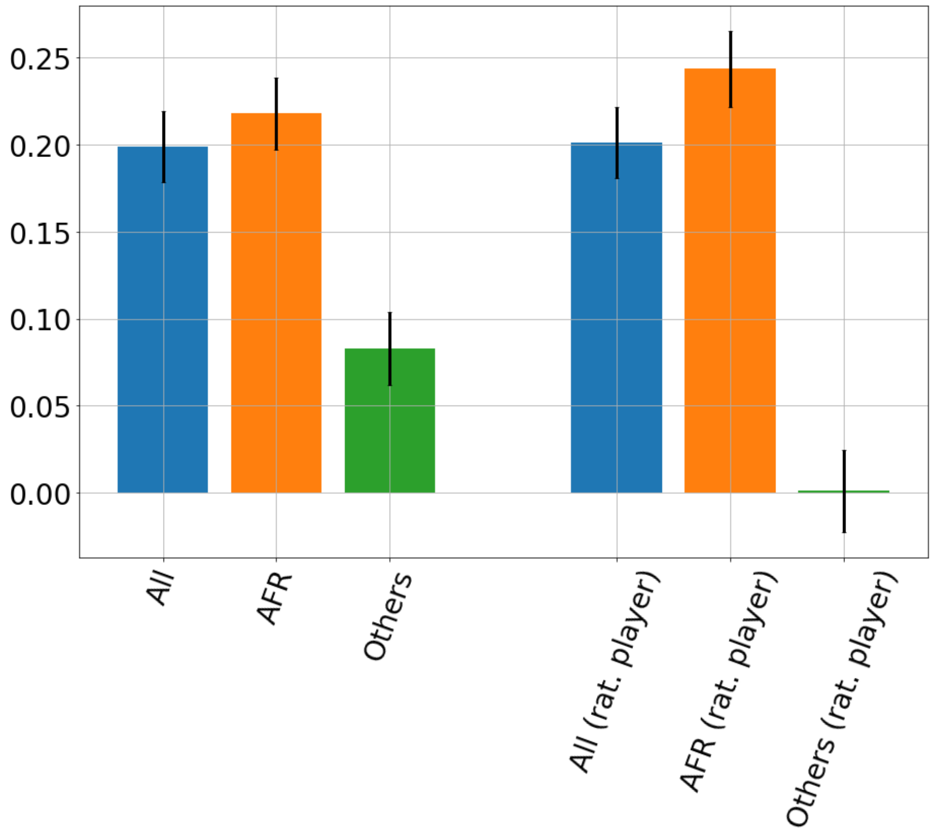


Figure 16: efficiency of GLOBAL mechanism for all airlines (blue), AFR (orange), and other airlines (green), for an honest AFR (left) and a rational AFR (right).

We now show the same experiment, but with the CM mechanism this time, in Figure 17. The first point to notice is that the efficiency of AFR drops, which is not what you would expect, given than a rational agent would be better performing than an honest agent in a single-sided game. The impact on the other companies seems also very mild, and in particular it is important to note this efficiency is far from dropping to 0, on the contrary of the previous mechanism.

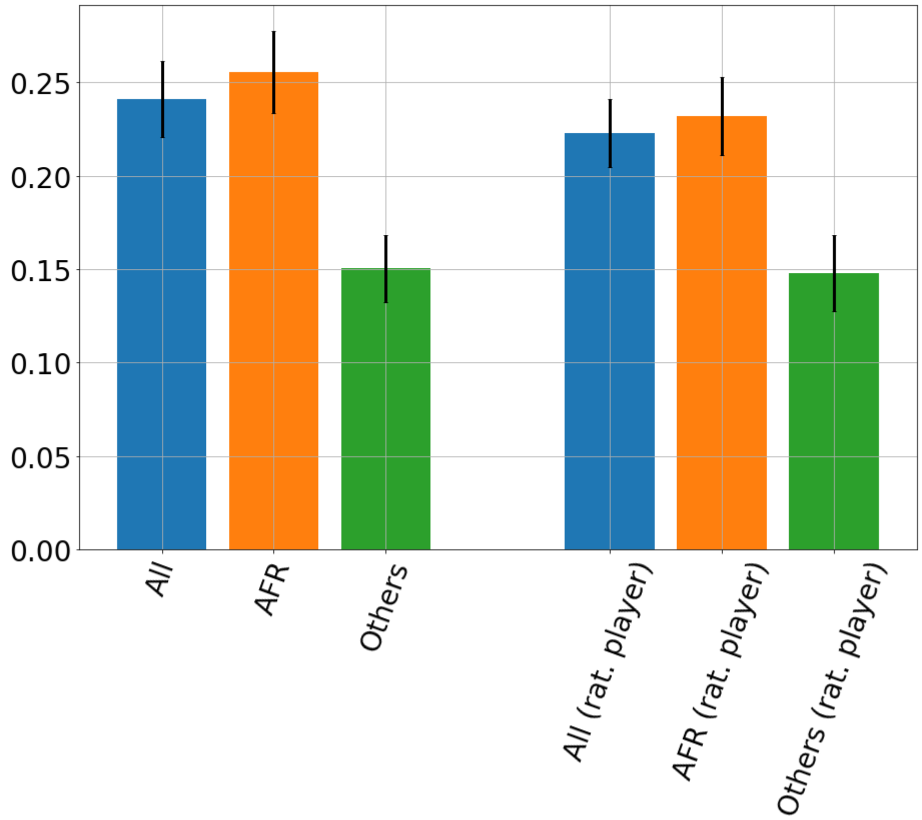


Figure 17: efficiency of CM for all airlines (blue), AFR (orange), and other airlines (green), for an honest AFR and a rational one (right).

These two figures illustrate the problem with **gaming**. Indeed, given a game with fixed rules, agents may (and will) try to take advantage in order to have the best outcome for them (here, the smallest cost possible). Gaming can destroy an otherwise very good mechanism, and is the main reason why one cannot just ask for airlines' costs and solve the regulations. Even though the results are limited by the quality of the rationality of the agents introduced here, it seems that introducing credits counter-balance to some extent the gaming behaviour, with honest company not losing everything from the gaming behaviour of other players (and here, even worse, very big players, which naturally have higher efficiencies).

The next question is to know whether injecting rationality for all agents tends to rebalance the situation or will make it even worse.

4.4.2 In multi-sided games

Indeed, in multi-sided games, all agents are considered players, and here, for simplification, are considered to follow the same strategy. 'Rationality' here is thus to be considered as 'single-sided rational', i.e. that we are using agents faring better in the single-sided game (at least in theory…) that will now play the multi-sided one with the same strategy, and examine if the total efficiency drops or not.

First, we show in Figure 18 the results when having all honest agents, one rational player (AFR, as in previous experiment), and all rational agents, in the case of the GLOBAL mechanism. The results are very clear: introducing rationality does not improve much the fate of other companies. While we were expecting a fall of the efficiency for AFR, it seems that using only rational agents has the same effect as having the major player being rational. There might be a subtle statistical effect due to the size of the company here that would require more study to uncover.

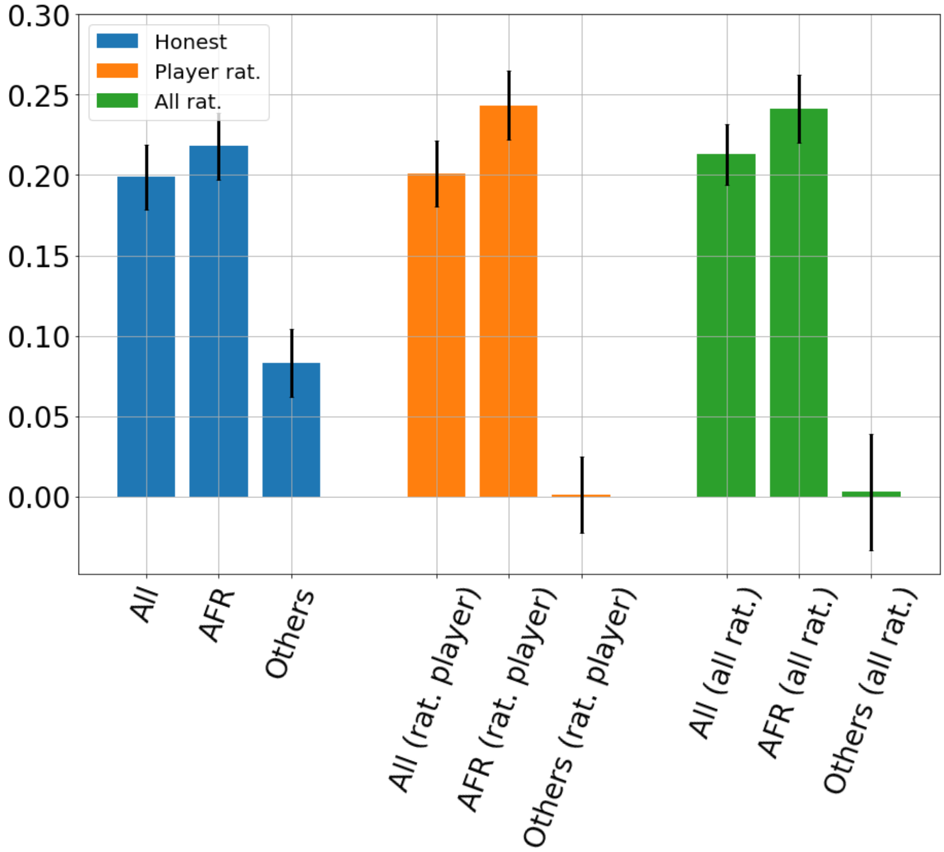


Figure 18: efficiency of GLOBAL for all airlines, AFR, and other airlines, with all honest players (left), AFR being rational (middle) and all players being rational (right).

Figure 19 shows the same results for the CM. In this case, there is a visible drop of efficiency when all agents are rational, but mainly the other companies and not AFR. While it would have been expected to have a drop for AFR too, these results are more in line with what is expected in case of gaming from all players. Note, however, that the efficiency the other companies is clearly different from 0, contrary to the case of the GLOBAL algorithm.

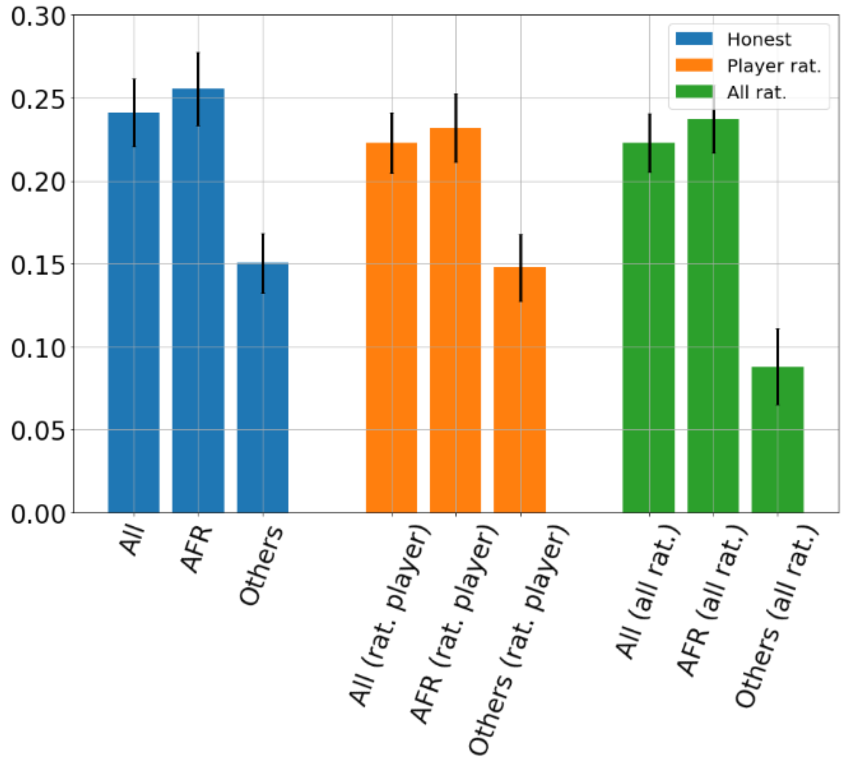


Figure 19: efficiency of CM for all airlines, AFR, and other airlines, with all honest players (left), AFR being rational (middle) and all players being rational (right).

Overall, the conclusion from these two figures is that the gaming effect from the main player is more important than the one from the other companies. In other words, it is crucial to take into account the gaming effect for the main player but not so much for the others. On top of that, it is clear that the introduction of credits in the mechanism has a net positive effect for small player when taking gaming effects into account, which is exactly why credits were introduced in the first place.

4.5 Impact of behavioural biases in GLOBAL and CM

In this section, we are interested in behavioural biases and their impact on the mechanisms. As noted previously, defining bounded agents with PT and HD is essentially distorting the decisions of a rational agent. Thus, the quality of the bounded agents is only as good as the quality of their rational counter-parts, and we already highlighted some of the issues we encountered when defining the latter. However, it is interesting to note that one can still have an idea of the magnitude of the drop of efficiency – again, if any – when introducing these distortions. On top of that, we introduced the idea of a simpler kind of bounded agents in section 2 and 3, which is not defined with respect to the rational agent, but rather to the honest agent, considering that the latter may be a good approximation for real agents in some cases (for instance, when gaming is a difficult task). We also explore the results obtained with these strategies in this section.

4.5.1 Bounded agents

We start with the bounded agent type, defined with respect to the rational one. As explained in sections 2 and 3, this agent type introduces two independent distortions to the decision of the rational agent type: the presence of prospect function replacing the profit (PT), and the modification of the discount factor on future gains (HD). We first have a quick look at the isolated effects of these two ingredients.

We also remind the reader that this agent type is only defined in the credits mechanism, because the rational agent of the GLOBAL mechanism maxes out its jump parameter (as shown in section 3), thus allowing no further meaningful distortion.

4.5.1.1 Effect of PT and HD

Like previously, we are interested in the effect on AFR and the other companies, although only AFR is bounded in this case. Figure 20 shows the efficiency of each of them when introducing PT, HD, and both combined, comparing with the rational case.

First, it looks like the introduction of PT alone has almost no effect on the system, with possibly only a small drop in the efficiency of non-AFR companies. The introduction of HD is more drastic, with a quite a clear drop for the non-AFR companies and a small one for AFR. Interestingly, it seems that the effect of HD and PT compounds, since their combined effects is clearly higher than the sum of their isolated ones. Hence, as an approximation, one can consider that the behavioural biases reduce the efficiency of AFR by a few percentage points, and by half for the other ones.

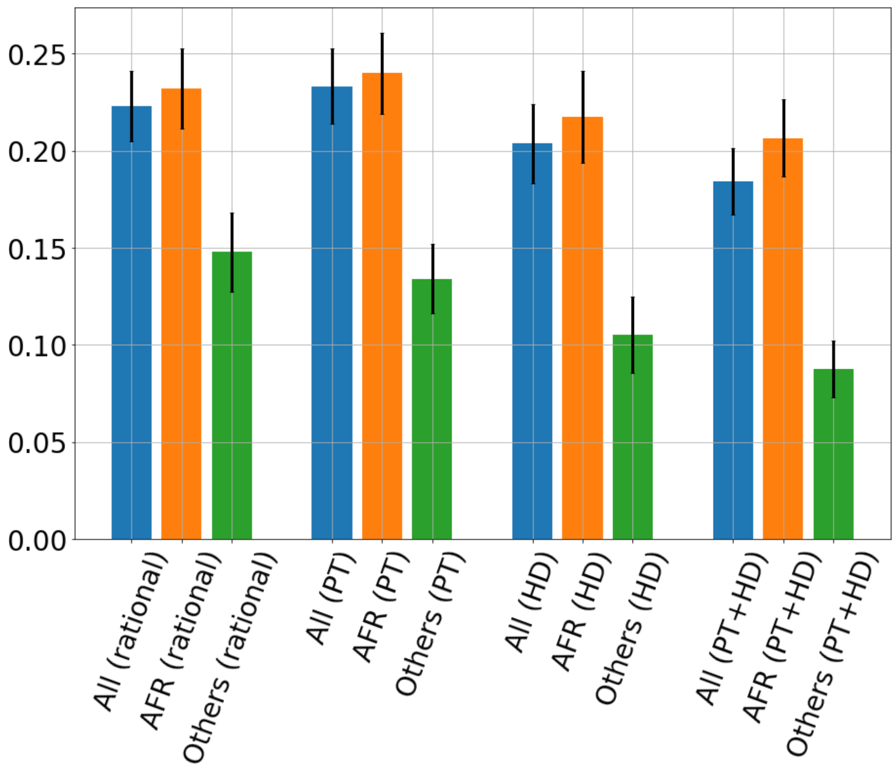


Figure 20: efficiency of CM for all agents (blue), AFR (orange), and other agents (green) when the simulations feature rational agents, agents with PT, agents with HD, and agents with both.

4.5.1.2 One-sided and multi-sided game

We then have a look at the single and multi-sided games when introducing behavioural biases, as shown respectively in Figure 21, comparing again the bounded case (now with PT and HD at the same

EUROPEAN PARTNERSHIP

Co-funded by the European Union

time) with the rational one. While there is a clear drop in the single-sided game, as already noted, the multi-sided game seems to be fairly insensitive to the introduction of the biases. It is very probable that the distortions introduced cancel each other in this case. While keeping in mind the caveats highlighted previously in this deliverable (section 3 for the rational part and section 4.1 for the approximation), it would mean that considering the agents as all rational could be a very good approximation to the real case, even when agents tend to be prone to behavioural biases.

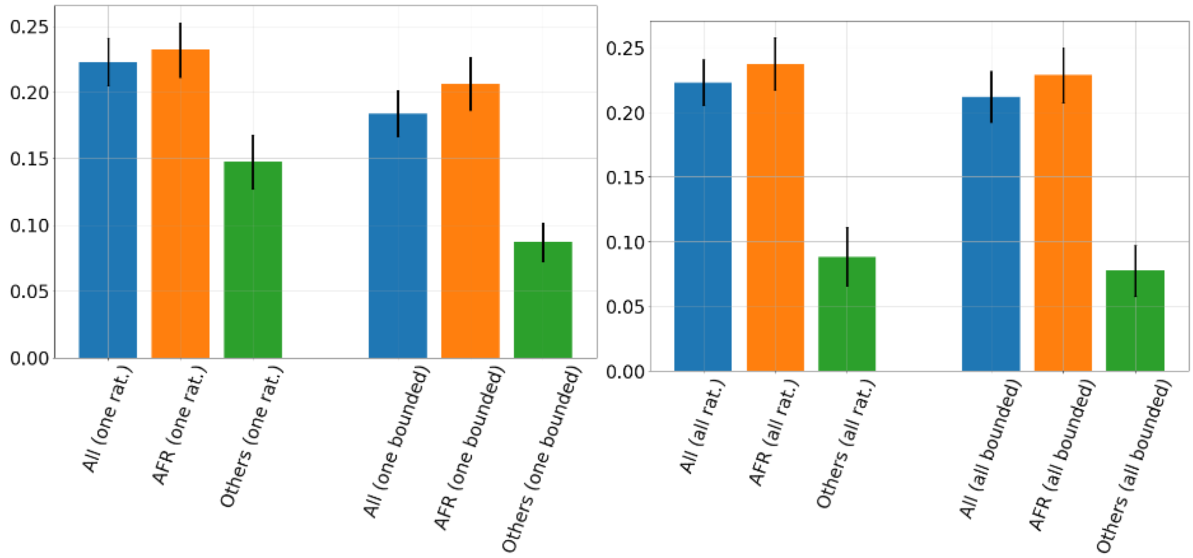


Figure 21: efficiency of CM for all agents (blue), AFR (orange), and other agents (green) when the simulations feature AFR as rational vs AFR as bounded (left) and all agents rational vs all agents bounded (right).

4.5.2 Bounded simple

We are now interested in seeing how the 'bounded-simple' agents, defined with a distortion of their real cost but otherwise honest, behave. Because of the simplicity of its definition, we can actually use the GLOBAL algorithm in this case, and we show the corresponding results in Figure 22. Note that this time, we compare the results to the honest case, since the bounded-simple agent is defined with respect to the honest one.

First, it is interesting to see that the distortion of AFR only seems to have a positive effect on the other companies, which are probably more prone to get a good place in the queue when AFR is making mistake. However, it is striking to see that the efficiency of AFR is exactly the same. It would tend to show that the mistakes do not drastically modify the queue in its disfavour, but only allows the other companies to gain advantages on some minor slots.

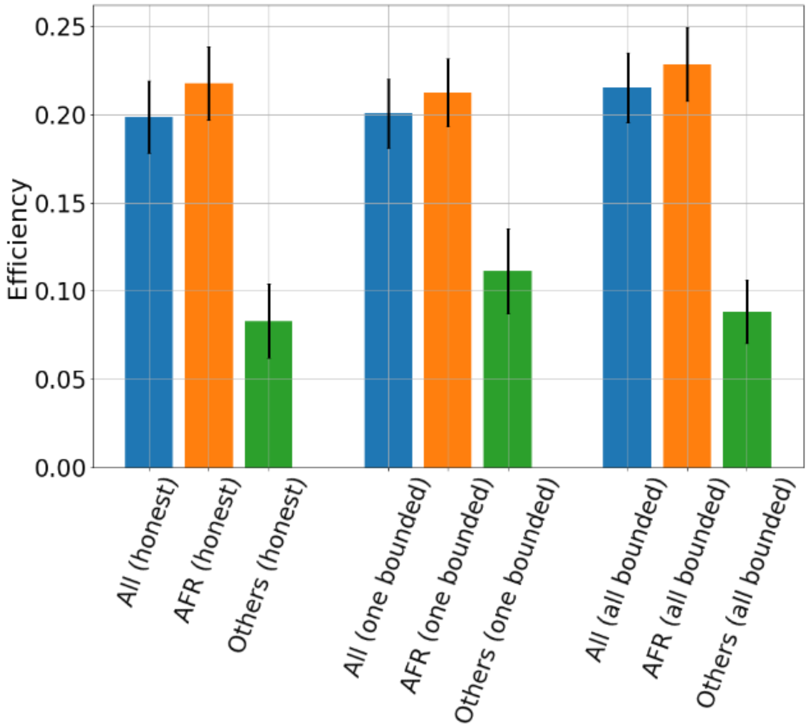


Figure 22: efficiency of GLOBAL for all agents (blue), AFR (orange), and other agents (green) when the simulations feature honest agents, one bounded-simple agent (AFR), and all bounded-simple agents.

The introduction of bounded-simple strategies for everyone seems, on the contrary, to have a positive effect on AFR, and a negative effect on the other companies. In fact, AFR looks like it has a higher efficiency as in the fully honest case, which would tend to show that the errors of your competitors are more important for your efficiency than avoiding your own mistakes, at least if you are a major player. This might be because the competition is fierce for the very first slots. When other small companies make mistake, they might free these slots completely involuntary, while the major player always has a flight ready to take these slots, even if it makes other mistakes otherwise.

Introducing bounded-simple agents seems to have a completely different impact in CM than bounded agents, however. As shown in Figure 23, the introduction of behavioural biases for AFR alone seems to destroy completely its efficiency, and gives a clear advantage to the other companies (in fact, this is the highest efficiency for these companies in any settings, almost 20%). Just like for GLOBAL algorithm though, introducing bounded strategies for them seems to benefit AFR greatly, with AFR recovering the efficiency from the honest case. This would tend to show, again, that the mistakes of your competitors are crucial when you are a major player.

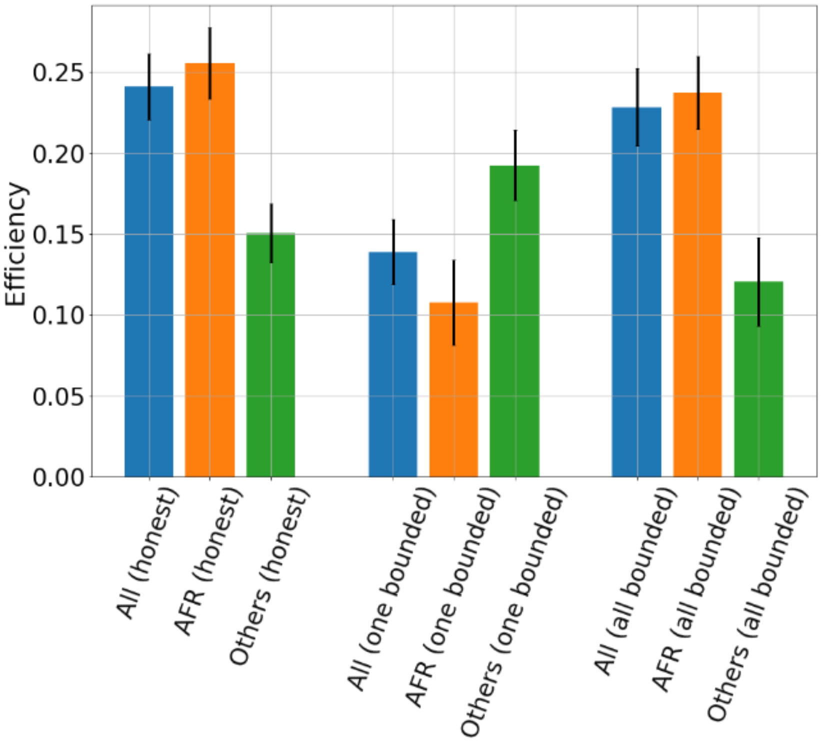


Figure 23: efficiency of CM for all agents (blue), AFR (orange), and other agents (green) when the simulations feature honest agents, one bounded-simple agent (AFR), and all bounded-simple agents.

4.6 Comparison of all mechanisms

Before moving on to more general conclusions, we try to give an overall picture of the mechanisms and agents tested in this deliverable, which could guide the future decision of a policy maker when introducing these new schemes.

In order to do this, we use the three indicators that were highlighted in section 2: efficiency, absolute equity (EQ1) and relative equity (EQ2), all aggregated for all airlines in the regulations. We focus on the case where all agents are of the same type, and we consider all the possible combinations of agents and mechanisms tested for the purpose of this deliverable. Figure 24 shows the results for efficiency, and Figure 25 for equity.

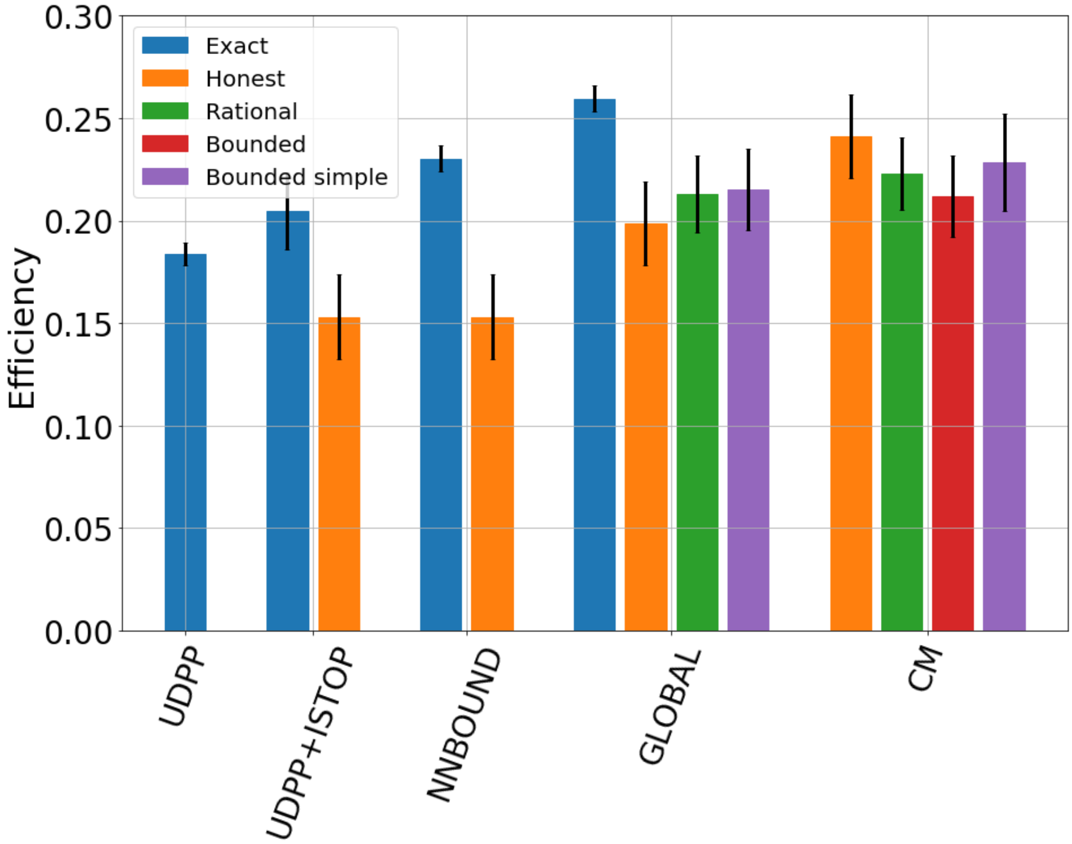


Figure 24: Efficiency for different mechanisms, with true costs (blue), honest agents with approximated functions (orange), rational (green), bounded (red), and bounded-simple agents (violet).

First, as noted before, the "naked" UDPP is already fairly efficiency, with around 18% of cost saved with respect to FPFS in average. ISTOP allows to gain a couple of more percentage points, although using the approximation, even the best one, puts it under UDPP. Using NNBOUND and GLOBAL further improves the situation, culminating to a 26% of decrease in cost in average for the theoretical maximum (for our regulation dataset). Even with the approximation, being honest is marginally better in the GLOBAL algorithm than using UDPP. Interestingly, introducing credits improves the situation quite a lot, with around 24% of savings for honest agents, higher than the UDPP+ISTOP mechanism, even with exact cost functions.

Introducing rational agents only improves marginally the situation for the GLOBAL algorithm, and decreases it for the CM one, probably due to the low quality of the rational scheme used here. Having behavioural biases only drops the efficiency by another percent in the CM case, or not at all if considering the bounded-simple strategy.

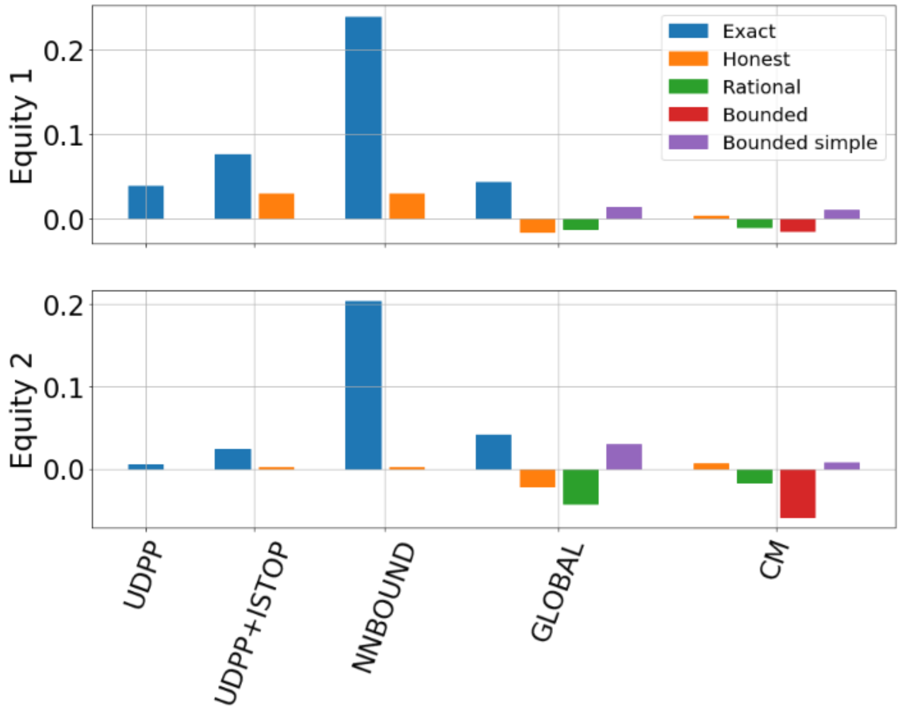


Figure 25: Equity indicators for different mechanisms, with true costs (blue), honest agents with approximated functions (orange), rational (green), bounded (red), and bounded-simple agents (violet).

The picture for equity is also rich and complex. While NNBOUND seems to be the champion of equity among all mechanisms, the CM mechanism fares very badly. Even UDPP, which is supposed to be highly inequitable due to the inherent possibilities to rearrange flights for major players, fares better. Rationality seems to have a very bad impact on equity, which might have been expected. Indeed, it is fairly clear that having more flights allows you to rearrange your flights more aggressively when maximising your profit, even though it is unclear if the agents defined here are actually able to do that. Interestingly, the introduction of behavioural biases may have a positive or negative impact on equity indicators, depending on the flavour (bounded or bounded-simple).

5 Conclusions

In this deliverable, we have explored the impact of different mechanisms for regulation solving, using different theoretical frameworks in order to paint a complete picture of the situation. We have introduced five main mechanisms, which were defined in previous deliverables and explored already in D4.2:

- UDPP, a local intra-airline optimisation,
- UDPP+ISTOP, a simple slot swap framework built on top of UDPP,
- NNBOUND, where we try to find the best outcome cost-wise in total, while making sure that no one loses,
- GLOBAL, where we are just interested in the best outcome for all airlines in total,
- CM, where we introduced credits in order to mitigate the effects of gaming in the previous mechanism.

We have introduced several agent types that mimic some of the behaviours that true airlines may have, or that were simply introduced for comparison purposes:

- the random agent, which is performing random decisions,
- the honest agent, which communicates its true costs to the central planner, to the best of its abilities,
- the rational agent, which is trying to maximise its profit (minimising its costs),
- the bounded agent, and its cousin the bounded-simple agent, which take into account behavioural biases, to which all humans are prone.

Combining different mechanisms and the agent types, the study aimed to find out what the theoretical effects of the introduction of these mechanisms are.

In order to do this, we built a dataset of regulations thanks to the Mercury simulator. Using many days of simulations, we built a dataset of regulations happening at CDG, including capacity information and cost structure for every airline involved in these regulations. This allowed us to run small scale simulations efficiently, building different agent types and letting them play the different mechanisms described above.

A first crucial conclusion reached with the results of section 4.1 is that having a good idea of the shape of the cost functions is very important to have an accurate picture of the effect of a mechanism. Indeed, using realistic cost functions and exploring several approximation schemes, it became clear that communicating a badly approximated cost function to the central planner will destroy a good part of the efficiency of said mechanisms. We saw for instance that in an ideal world, even if airlines knew their true costs, if they were forced to communicate their costs with the kind approximation tested here to the central optimiser in order for it to perform the best optimisation possible (with the GLOBAL algorithm), the efficiency could drop from 25% of savings to only 6%. Improving the approximation archetype function improves the situation, but complicates the mechanism, since more parameters need to be communicated to the central optimiser[8]. Quite crucially, it is hard to beat the bare UDPP mechanism with an approximated function, which defeats the whole purpose of introducing an extra layer on top of it. Thus, our current recommendation is to study the approximation problem in depth before designing any new mechanism. This is true for the CM for instance, which relies at its heart on an approximation to the cost function.

A second conclusion, more theoretical but with a very practical consequence, it that the CM game that we have defined is "hard" to play. By hard, we mean that finding a good strategy to find a better outcome than just the honest communication of the true cost to the central planner is far from obvious. While the best strategy is obvious for the GLOBAL game (at least in single-sided games: the airline just inflates their costs), the introduction of credits makes it very difficult to find a good strategy again. This is actually very good news, and is the reason why credits were introduced in the first place. Indeed, if finding an optimal strategy is very time-consuming, companies will naturally be tempted to play honestly. This is better for the mechanism, as can be seen clearly with the much higher efficiency reached by small companies in the CM game with respect to the GLOBAL one. Note, however, that this is not an absolute conclusion, since we have only scratched the surface of the possible strategies in multi-sided games. Going further, however, requires more advanced tools, like reinforcement learning, as discussed further below.

A third conclusion concerns the behavioural aspect. Indeed, the introduction of biases seems to have a small (in general) but significant impact. Our simulations show that efficiency can change by a few percent points, even more in some cases. Our investigation seems to indicate that making mistake is less important when you are major airline and that your other competitors make some too. However, being the one to make mistakes seems to be penalising. These results would tend to show that including behavioural economics improves the quality of the prediction when designing such mechanisms, and that relying only on rational players may be dangerous from the indicator computation point of view.

Overall, the CM mechanism seems to be faring well in terms of total saved costs. Its efficiency, even when introducing gaming and biases, seems to be slightly higher than UDPP, or even UDPP+ISTOP with real cost. While not as high as the theoretical maximum (GLOBAL with true costs), it seems better than the GLOBAL algorithm with approximated cost function. Its main drawback seems to be the equity. While GLOBAL has an equity comparable to UDPP in general, at least with true costs, CM is very inegalitarian. This is bad news, in particular with because credits were introduced in order to give more room to small airlines to make gains. It is still not clear to us what should be done exactly for the low-volume users but we note the following points:

- First, a better calibration of the CM might be needed. It is possible that high-volume users make scale returns on the credits, for instance with lots of unimportant flights fuelling the

[8] It is true in general for any mechanism if any kind of approximated cost is communicated to the NM. However, it is particularly crucial for CM, because each additional parameter creates new calibration issues (because it adds a new price and a new default parameter).

credit spendings on a few important ones. To counterbalance this effect, it may be worth trying to give different initial credit endowment to airlines for instance, based on the number of flights they have in regulations in average.

- The CM mechanism is currently based on a global cost reduction optimisation, i.e. that the mechanism solves for minimum total cost across airlines. Another possibility is to include an equity indicator in the objective function itself. This would naturally drive the low-volume users to be less disadvantaged. In fact, it would be an easier way to create a balance between economic efficiency and equity (there is always a trade-off between these two), instead of designing new mechanisms and hoping they have better characteristics in equity while probably having lower economic efficiencies.
- Before going further, we suggest however to study the exact reason why the low volume users are apparently disadvantaged. For instance, it may be that they have lower costs in average and thus cannot gain as much as the others from any mechanisms. However, it is still striking in this case that NNBOUND can reach such high degree of equality, while not sacrificing its economic efficiency too much. Hence, another possibility is to design a mechanism inspired by NNBOUND, maybe merging it with CM or ISTOP.

This leads us to calibration problems. Indeed, the CM is more complex to calibrate than UDPP or UDPP+ISTOP, which have no tunable parameters once the basic rules are decided. On the contrary, the CM has as twice as many parameters as the number of parameters used in the approximation of the cost function (one price parameter for each, and one default parameter), plus the initial endowment of credits to each airline. In this deliverable we have simplified the problem, using some honest behaviour on most parameters except the one of importance (the jump), but a more general calibration should be performed. In particular, giving a different initial endowment of credits to different airlines based on the number of flights they have, may help counter-balancing some of the equity effects we have seen in the mechanism. Note also that having a more complex game has also a very good side effect on gaming issues, since winning strategies are even harder to come by and thus airlines will be tempted to go for honest communication of costs, even relying on automated "honest" tools. Further considerations on this subject will be made in the concept assessment deliverable D6.2, in particular on the subtle interplay between complexity, automation, and gaming.

There are two main limitations to the present results. First, we have performed simulations only on a specific airport, CDG, for simplicity. It is not obvious whether the present results generalise easily to other airports, even if major hubs should behave fairly similarly. However, it is important to explore the impact of these mechanisms and agent types on the resolution of other airports in Europe, and this will be the task of D5.2. In that deliverable, we will use a more general dataset of regulations, computed all over Europe, in order to draw more general conclusions. We will also present a more detailed study on the effect of the size of regulations and the size of airlines on the efficiency and equity of the mechanisms.

Second, as noted previously, we have only scratched the surface of optimal strategies. The 'rational' agents presented in section 2 and 3 are very simple and could be improved in many ways. A natural candidate is reinforcement learning, an optimisation framework which allows to find optimal strategies in complex iterated games. In particular, the RL framework aims at finding the best approximation for the mapping between an action (a decision) and its consequences (short and long term) given the state of the agent and environment (the number of credits for example, but other information like the total number of slots can be included). This is typically done by using advanced machine learning models to approximate this mapping, like neural networks. An important aspect is to find the right level of information given to the agent and how to use them efficiently to predict the best action in a given situation. While the BEACON has already explored some of these techniques during the preparation of these results, we keep these research questions for another time. Thus, an example of practical steps that one could take to improve the current model are the following:

- Frame the current game as a single-sided iterated game with continuous or discrete decision action space, with the reward being the cost saved by the player.
- Find the relevant features to be used by the player to make the mapping between the potential decisions and the expected gain.
- Train different models of agent, starting with simple artificial neural networks.
- Study the characteristics of the optimal strategies for the different mechanisms and different strategies followed by non-players.
- Test optimal strategies without training, as in multi-sided games from this deliverable. Study the matrix or function of reward depending on the strategies followed. Find numerically the Nash equilibrium(a), if any[9]. Study if it (they) are Pareto efficient.
- Next, frame the game as multi-agent iterated game, with the reward of each agent being their cost saved. Allow for a degree of collaborative reward for agents to test gaming effects.
- Train the model. Study its stability, its fixed points if any, in particular in regard to the strategies found for the single-sided game.
- Conclude on the potential strategies likely to appear in reality and compute indicators accordingly.

Finally, we will explore in D6.2 the consequences of the results contained in this deliverable on the potential implementation of the mechanisms. We will also highlight the next steps suggested for this line of research beyond BEACON.

[9] There is at least one if the decision space is discrete, according to the Nash theorem.

6 References

[1] Grant Agreement No 893100 BEACON Annex 1 Description of the Action

[2] BEACON Consortium Agreement

[3] BEACON D1.1 Project Management Plan, July 2020

[4] BEACON D2.1 Data Management Plan, December 2020

[5] BEACON D2.2 Database structure and data elaboration, January 2022

[6] BEACON D3.1 High-level modelling requirements, December 2020

[7] BEACON D3.2 Industry briefing on updates to the European cost of delay, September 2021

[8] BEACON D4.2 Final model results, July 2022.

EUROPEAN PARTNERSHIP

Co-funded by the European Union

7 Acronyms

Acronym	Meaning
AFR	Air France
BE	Behavioural Economics
CDG	Charles de Gaulle airport
CM	Credit Mechanism
FPFS	First Plan First Served
HD	Hyperbolic Discount
ISTOP	Inter-airline Slot Trading Offer Provider
NNBOUND	Non-negative bounded optimisation
PT	Prospect Theory
UDPP	User-Driven Prioritisation Process

EUROPEAN PARTNERSHIP

Co-funded by the European Union

Appendix A Regimes in credit evolution during CM

Figure 26 shows the transition of the evolutions of the number of credits (in particular for the main player) when the default jump parameter increases (resp. 0, 50, 100, 150, 200 euros). The regime goes from an absorptive regime (with credits going to 0 after a few iterations) to an explosive one (with credits going to infinity).

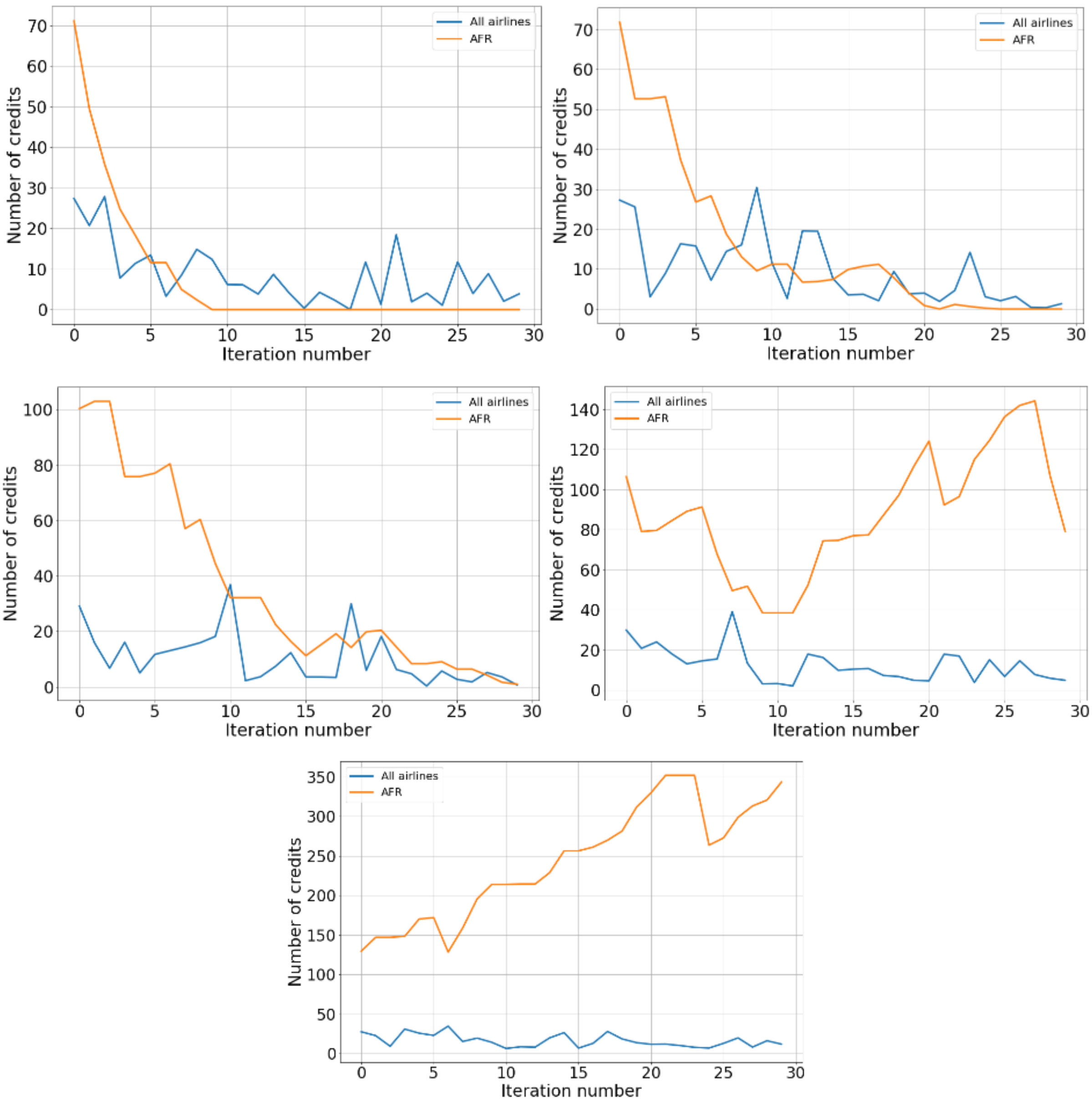


Figure 26: Evolution of the number of credits for AFR and other companies for different values of the default jump parameter (see text for values).

Appendix B Auction mechanism and rationality

We present in this annex the theoretical framework that would have been used to implement rational and bounded agents. The framework is similar to the one presented for the CM in section 3 and has the same limitations, notably the model mapping actions to probabilities of outcome.

1 Notations

Vectors have an underbar, like that $\underline{u}$.

2 Auctions

2.1 Complete problem without BE

Consider a flight in auction with N slots. The real cost of delay of the flight departing at each slot is captured by the vector $\underline{v}$. Then the profit (avoided loss) of the airline for this flight is:

$$r = (-\underline{v} - \underline{g}(\underline{b})) \,.\, \underline{p}(\underline{b}), \tag{1}$$

where:

- $\underline{b}$ is the vector of bids for each slot
- $\underline{p}$ is the vector of probability that the flight get assigned to each slot after the auction resolves.
- $\underline{g}$ is a function where $\underline{g}(\underline{x}) = \{g(x_i)\}_i$, g being a scalar function.

The expression (1) represents the expected profit (avoided loss) in terms of cost of delay, taking into account the value of credits for future regulations. Indeed, function g represents the value of a single credit, in terms of the cost that it could save in the future.

The complex thing is of course to compute $\underline{p}(\underline{b})$, i.e. the probabilities of getting each slot as a function of the bids. $\underline{g}(\underline{b})$ needs also to be estimated, but it is less complex, as we need to estimate only the function g.

In any case, the problem to solve is then:

$$\max_{\underline{b}} r$$

2.1.1 Credit value estimation

As a first approximation, g could be considered as linear, i.e. $g(b) = \alpha b$. α is then just an exchange rate between euros and credits, which may be computed over a high numbers of simulations (solving a regulation without g, computing how much the airline spent and how much it saved them).

2.1.2 Probability estimation

One could assume a functional form for the probabilities, i.e.:

$$p_i(\underline{b}) = \frac{e^{b_i/\lambda}}{\sum_j e^{b_j/\lambda}} \tag{2}$$

1

This form for instance has one parameter to estimate, to be done similarly to $\underline{g}$. However, it does not take into account that first slots are a lot harder to get that late ones (they are naturally more valuable). Thus, a better model may be:

$$p_i(\underline{b}) = \frac{e^{b_i/\lambda_i}}{\sum_j e^{b_j/\lambda_j}}, \tag{3}$$

which has now N parameters, one par slot. However we can probably assume a functional form for the λ_i too, since $\lambda_{i-1} > \lambda_i$ – again, early slots are harder to get. We could thus assume for instance:

$$\lambda_i = a/(i+b),$$

which reduces the parameters to estimate to two (a and b).

2.2 BE

BE could be injected like follows.

2.2.1 Utility function

First, we can replace the actual loss avoidance by the utility function $\underline{u}$, applied on each component independently:

$$r' = \underline{u}(-\underline{v} - \underline{g}(\underline{b})) \,.\, \underline{p}(\underline{b}) \tag{4}$$

Like for $\underline{g}$, $\underline{u}$ is a scalar function applied component per component, i.e. $\underline{u}(\underline{x}) = \{u(x_i)\}_i$, The reference point could be the cost of delay of the flight in the FPFS allocation. The Cumulative Prospect Theory (CPT) function ν is given by:

$$\nu(x) = \begin{cases} x^\alpha, & \text{if } x \geq 0 \\ -\lambda(-x)^\beta, & \text{otherwise} \end{cases} \tag{5}$$

2.2.2 Weighting function

Then one can inject weighting function from PT to modify the probabilities:

$$r'' = (-\underline{v} - \underline{g}(\underline{b})) \,.\, \underline{\omega}(\underline{p}(\underline{b})), \tag{6}$$

where $\underline{\omega}$ are the weighting functions, actually applying a unique function ω to each component, like $\underline{u}$ and $\underline{g}$. This weighting function in the context of CPT, usually takes the form of:

$$\omega \pm (p) = \frac{p^\gamma}{(p^\gamma + (1-p)^\gamma)^{1/\gamma}} \tag{7}$$

with $\gamma = \gamma\pm \in (0, 1]$.

2

2.2.3 HD

Finally, one can modify the value of a credit with respect to the baseline. For instance, if a regulation hits in average every T, if we consider only the next iteration, then the value of C euros is:

$$V = \frac{C}{1 + kT}.$$

Thus, function $\underline{g}$ can be replaced by:

$$\underline{g}'(\underline{b}) = \frac{\alpha \underline{b}}{1 + kT} = \beta \underline{b} \tag{8}$$

Thus, putting together equations (1), (4), and (8), the final equation is:

$$r^* = \underline{u}(-\underline{v} - \beta \underline{b}) \,.\, \underline{\omega}(\underline{p}(\underline{b})) \tag{9}$$

2.3 Several flights

For several flights (say M) in the same airline, we can use tensors, for instance v is now the tensor that we note v_{ij}, where i denotes the slot and j the flight. Using Einstein notation for tensors, the fully rational case becomes:

$$r = (-v_{ij} - g(b_{ij}))p^{ij}(b).$$

In this case, the probabilities depend on the entire tensor b. The BE case is thus:

$$r^* = u(-v_{ij} - \beta b_{ij})\omega(p^{ij}(b)). \tag{10}$$

The probability estimation may be done by using a similar functional approximation than above:

$$p_{ij}(b) = \frac{e^{b_{ij}/\lambda_i}}{\sum_l e^{b_{kl}/\lambda^k}}, \tag{11}$$

possibly with the same kind of form for λ:

$$\lambda_i = a/(i + b),$$

Note that the problem is harder to solve, since the space of search is $N \times M$, but the number of parameters to estimate is the same (in particular a, b, β).

Note also that the above is valid when all flights have access to all slots, which is not the case in general.

Constraints:

- zero-sum bid for each flight:

$$\forall j, \sum_i b_{ij} = 0$$

- Sum of max bids has to be smaller than the available credits:

$$\sum_j \max_i b_{ij} \leq C$$

3

3 Probability estimation

Using the simple case of 2 flights and 2 slots, the probability that the first flight gets the first slot is :

$$p_0(b_0) = P(b_0 > b'_0) = F_0(b_0)$$

with b the bid vector of the first flight, b' the bid of the second one, and F_0 the cumulative distribution of the bids in the first slot.

As an example, take the second flight has a random player, using $B' \times [1, -1]$ as bid, with B' a random uniform variable between 0 and L. In this case:

$$p_0 = \frac{\bar{b_0}}{L}$$

where $\bar{b_0} = \max(0, \min(b_0, L))$. In other words, p_0 is a piece-wise liner function.

For three flights, the probability becomes:

$$\begin{aligned} p_0 &= P((b_0 > b'_0) \cap (b_0 > b''_0)) & (12)\\ &= P(b_0 > b'_0)P(b_0 > b''_0) & (13)\\ &= F_0(b_0)^2 & (14) \end{aligned}$$

Where F_0 is the cumulative function for the random bids, like previously. More generally, if we have n flights and n slots, $p_0(b_0) = F_0(b_0)^{n-1}$. Note that this probability does not depend on the bids on other slots. This is because the slots are resolved one by one starting from the first one.

For the next slot, the probability of allocation is (for instance for three flights):

$$p_1 = P(\text{not allocated before}) \times P(b_1 > b'_1 | b''_0 > b'_0).$$

The probability on the right takes into account that when the flight has not been allocated in the first slot, the bids on the second slots are the ones that remain, and thus have a different probability distribution. For instance, imagine that the probability of having $b1'$ in the second slot is correlated positively to the probability of having b'_0 in the first slot. Then, when the first slot is allocated to another flight, it means that the bid b'_0 was not high enough, and thus the bid b'_1 is in average smaller. In general, this correlation exists, and depends highly on the type of strategies used by the other airlines. Here, we will assume for simplicity that $P(b_1 > b'_1 | b''_0 > b'_0) = P(b_1 > b'_1)$, i.e. there is no correlation.

In this case, for three flights, the probability becomes:

$$p_1 = (1 - F_0(b_0)^2)F_1(b_1).$$

It is then easy to show that in general the probability p_k of getting the slot k is thus:

$$p_k = F_k(b_k)^{n-k-1} \prod_{i=0}^{k-1} \left(1 - F_i(b_i)^{n-i-1}\right),$$

4

where F_j is the cumulative probability distribution of a flight bidding on slot j.

Note that these probabilities have the following properties:

$$\forall k, \frac{\partial p_k}{\partial b_k} \geq 0,$$

$$\forall k > j, \frac{\partial p_k}{\partial b_j} \leq 0,$$

$$\forall k < j, \frac{\partial p_k}{\partial b_j} = 0.$$

In other words, the probability to be allocated to a slot k is independent of the bids on the subsequent slots, is increasing with the bid on the slot k, and decreasing with the bids on the previous slots.