D2.1 Trade-off report on multi criteria decision making techniques

Deliverable 2.1

Pilot3

Grant:	863802
Call:	H2020-CS2-CFP09-2018-02
Topic:	JTI-CS2-2018-CfP09-SYS-01-12
Consortium coordinator:	University of Westminster
Dissemination level:	Public
Edition date:	22 September 2020
Edition:	01.01

Authoring & Approval

Authors of the document

Name/Beneficiary	Position/Title	Date
Adeline de Montlaur / Universitat Politècnica de Catalunya	Project member	22 September 2020
Jovana Kuljanin / Universitat Politècnica de Catalunya	Project member	22 September 2020
Xavier Prats / Universitat Politècnica de Catalunya	Project member	22 September 2020
Luis Delgado / University of Westminster	Project member	22 September 2020
Gérald Gurtner / University of Westminster	Project member	22 September 2020

Reviewers internal to the project

Name/Beneficiary	Position/Title	Date
Samuel Cristóbal / Innaxis	Project member	22 September 2020

Approved for submission to the CSJU

Beneficiary	Position	Date
University of Westminster	Project coordinator	22 September 2020

Document History

Edition	Date	Status	Author	Justification
01.00	29 May 2020	Release	Pilot3 Consortium	New document for review by Topic Manager
01.01	22 September 2020	Release	Pilot3 Consortium	New document reviewed by Topic Manager; for review by CSJU

Pilot3

A SOFTWARE ENGINE FOR MULTI-CRITERIA DECISION SUPPORT IN FLIGHT MANAGEMENT

This deliverable is part of a project that has received funding from the Clean Sky Joint Undertaking under grant agreement No 863802 under European Union's Horizon 2020 research and innovation programme.

Abstract

This deliverable describes the decision making approach that will be followed in Pilot3.

It presents a domain-driven analysis of the characteristics of Pilot3 objective function and optimisation framework. This has been done considering inputs from deliverable D1.1 - Technical Resources and Problem definition, from interaction with the Topic Manager, but most importantly from a dedicated Advisory Board workshop and follow-up consultation. The Advisory Board is formed by relevant stakeholders including airlines, flight operation experts, pilots, and other relevant ATM experts.

A review of the different multi-criteria decision making techniques available in the literature is presented. Considering the domain-driven characteristics of Pilot3 and inputs on how the tool could be used by airlines and crew. Then, the most suitable methods for multi-criteria optimisation are selected for each of the phases of the optimisation framework.

Table of Contents

List of figures

List of tables

Executive summary

Pilot3 is a CleanSky2 Innovative Action. The Topic Manager is Thales AVS France SAS. Pilot3 aims at providing a software engine model for supporting crew decisions.

The primary objective of Pilot3 is to develop a **software engine model** for supporting crew decisions for civil aircraft. This software will provide the crew with a **set of options** along with information to aid the crew to select the most suitable one considering the **multi-criteria business objectives** of the airline, including the **impact on the network** of flights of the airline of those decisions.

Pilot3 comprises five sub-systems:

- **Alternatives Generator**, which will compute the different alternatives to be considered by the pilot; fed by the two independent sub-systems:
    - **Performance Indicators Estimator**, which provides the Alternatives Generator with information on how to estimate the impact of each solution for the different performance indicators (PIs) needed to estimate the optimisation objectives which are relevant to the airline;
    - **Operational ATM Estimator**, which provides the Alternative Generator with information on how to estimate some operational aspects such as tactical route amendments, expected arrival procedure, holding time in terminal airspace, distance flown (or flight time spent) in terminal airspace due to arrival sequencing and merging operations, or taxi-in time;
- **Performance Assessment Module**, which, considering the expected results for each alternative on the different KPIs, is able to filter and rank the alternatives considering airlines and pilots preferences, and to show them to the pilot. This is part of the multi-criteria optimisation process; and
- **Human Machine Interface**, which will present these alternatives to the pilot and allow them to interact with the system.

Pilot3 has four different execution phases:

1. **Configuration phase**: This phase is performed by the airline prior to the flight to set the different parameters and preferences on the usage of Pilot3 tool.
2. **Generation phase:** A first automatic generation and selection of candidate solutions will be produced by the Alternatives Generator considering: the airline’s objectives, operational constraints, environment data, and information from the Indicator Estimator and the Operational ATM Estimator.
3. **Ranking phase:** The alternatives' ranking phase is the first part of the Performance Assessment Module and it consists on ranking the alternatives provided by the Alternatives Generator. This post-processing of the trajectories generated by the Alternatives Generator is performed in order to pre-compute how the alternatives will be presented to the pilot. This could be considered as a discrete multi-objective optimisation.

4. **Selection phase:** This is the final step of the Performance Assessment Module and it considers the interaction with the pilot via the Human-Machine Interface (HMI) to capture pilot operational related aspects.

This deliverable presents the definition of the multi-criteria methods that will be used on the different optimisation phases. This selection has been done considering inputs from:

- **Pilot3 Deliverable D1.1** - Technical Resources and Problem definition (Pilot3 Consortium, 2020),
- **Topic Manager**,
- **Advisory Board workshop**, and
- **Advisory Board consultation**.

First, an identification of the objectives that are relevant to the airlines' operations is conducted:

1. a set of performance indicators (PI) were identified (e.g., number of passengers missing connections, minutes of delay at arrival);
2. PIs were aggregated into KPIs (e.g., cost of fuel, cost of IROPs); and from these,
3. two objectives that should be considered when optimising the trajectories were derived: **cost** and **on-time performance (OTP)**.

These two objectives that are relevant for airlines have the following characteristics:

1. **Cost** is a complex objective build from the aggregation of 3 key performance indicators (KPIs):
    a. cost of fuel,
    b. cost of IROPS, including hard and soft passenger costs (considering connecting and non-connecting passengers), and
    c. other costs, which account for extra crew and maintenance costs, but most importantly for reactionary costs.
2. **On-Time Performance (OTP)** is considered as a binary variable of achieving on-time performance (i.e., arrival delay ≤ 15 minutes or not)

Pilot3 supports the crew by providing solutions which would produce the best outcome in average, therefore, the optimiser will be **risk-neutral**.

After the literature review on multi-criteria optimisation techniques, ten different criteria grouped in five categories have been used to select the most adequate technique for each of the Pilot3 phases:

a. **Data (input) required by the method**
    1. the input needed for the method to function should be available
    2. responsibility sharing on user (dispatcher, pilot) providing the input required

b. **Objectives considered**
    3. ability to deal with high/low number of objectives
    4. consideration of variability/uncertainty

EUROPEAN UNION

c. **Human-machine interface considerations**

5. easiness of providing the input required
6. easiness of providing the output required

d. **Other non functional considerations**

7. computational cost of the method
8. easiness to implement the method
9. the method should provide a necessary and sufficient condition method for Pareto optimality

e. **Other functional considerations**

10. other general preferences expressed by stakeholders

The details on the optimisation technique that will be used to optimise the trajectories considering the multi-criteria framework will be defined in WP4. Further factors will then be considered, such as the possibility of exploring the space of search to provide more than one alternative which can be considered optimal, or the mathematical modelling of the flight trajectory with uncertainty.

First, the characteristics of the problem have been used to filter the number of potential methods across all the optimisation phases. Then, remaining candidate methods were further analysed considering the particularities of each of the execution phases:

- **Generation phase**: The consultation with the Advisory Board established that the optimisation should focus on minimising cost, and on producing the trade-off (in cost) required to achieve the OTP, if possible. This has led to the selection of the **Lexicographic ordering** multi-criteria optimisation method to capture this trade-off. Trajectories will be generated considering cost as first objective and OTP as second. If OTP is not achieved, then the trajectory generator will be re-executed forcing OTP (if possible), by adding it as a constraint, and then minimising the cost as a second objective. The generation phase will not select a solution but rather generate different trajectories in the Pareto front so that the decision maker can explore the alternatives after their ranking.

- **Ranking phase**: The relative importance of the KPIs which form the cost objective (cost of fuel, cost of IROPs and other costs), as provided during the Configuration phase, will be used to rank and filter the solutions produced by the Generation phase. This will be a discrete optimisation for which the most suitable method is the **Compromise Ranking Method**, also known as the **VIKOR** method, improved by introducing the **Analytical Hierarchy Process** for assigning the weights of relative importance of attributes.

- **Selection phase**: The Selection phase will rely on the interaction with the pilot via the **Human-Machine Interface**. The pilot will receive information on the alternatives and will be able to explore and compare the ranked trajectories. Some tactical operational indicators will be computed to facilitate this process (e.g., number of flight level changes required). The crew will be able to dismiss trajectories, add constraints, and re-execute the optimisation to re-evaluate the alternatives. Note that the ranking of alternatives could include the previously generated (and not yet dismissed) trajectories. The specific requirements and design of the interface will be performed as part of WP4 activities.

1 Introduction

1.1 Pilot3: multi-criteria decision making tool for support in flight management

During flight operations, when a disruption (or an update on information affecting the trajectory prediction, such as an update on the weather forecast) arises, the crew might consider to modify the planned trajectory. The pilot should consider the airline's targets and policies, and using a trajectory optimisation or prediction system, estimate different trajectory options. Different alternatives will have a different impact on the expected duration of the flight (time) and/or on the amount of fuel consumed (kg), which are typically the main outcomes of these trajectory optimisation/prediction systems.

Airlines do not necessarily focus on these two indicators (fuel and time) but on other high level objectives, such as cost of the operations (e.g., irregular operations costs due to passenger management (IROPS), reactionary costs), or on-time performance (see Section 1.4 for more information on the different objectives considered on the optimisation). In some cases, a pre-defined cost index[1] could be used to translate the variation in time and fuel into equivalent fuel usage. However, note that the cost index can be considered as a proxy to the real indicators that are relevant to the airline, as previously indicated.

The crew assesses the outcome of these optimisations along other available data (such as the list of connecting passengers who are on board, or previous experience on expected delay at arrival for that particular route) and has to estimate the existing trade-offs to decide if it is worth it to recover a given amount of time using a certain amount of extra fuel. During this analysis and selection exercise, the crew might discard options, which they do not accept as valid (e.g., changing cruise altitude to a level where the pilot knows turbulence is experienced), and mentally ranks the different possibilities to select the one that is considered best. Thus, the pilot is doing a manual iterative analysis of alternatives within a multi-criteria optimisation.

Note that different airlines might have different policies in place. Yet, one common approach to managing larger disruptions is to estimate the alternatives from the ground (e.g., monitoring the operation of flights by dispatchers) and indicate to the pilot how they should operate (e.g., which cost index to select). However, in some instances, for example, small variations (e.g., weather update), or when considering tactical operational issues (e.g., where to perform the top of descent), pilots still

[1] Current Flight Management Systems (FMS) (re)compute flight trajectories by minimising a compound objective function that considers both fuel and time costs. The Cost Index represents the ratio between time and fuel costs and is manually introduced by the pilot into the FMS (Airbus, 1998).

maintain some autonomy. Moreover, pilots might still make decisions based on their own interpretation of priorities, which might vary from flight to flight. And even if the decision is performed on-ground the same principles of multi-criteria considerations apply.

Pilot3 will develop an optimisation engine within a multi-criteria optimisation framework to assist the crew on this process. **Figure 1** presents the high-level architecture of the suggested solution.

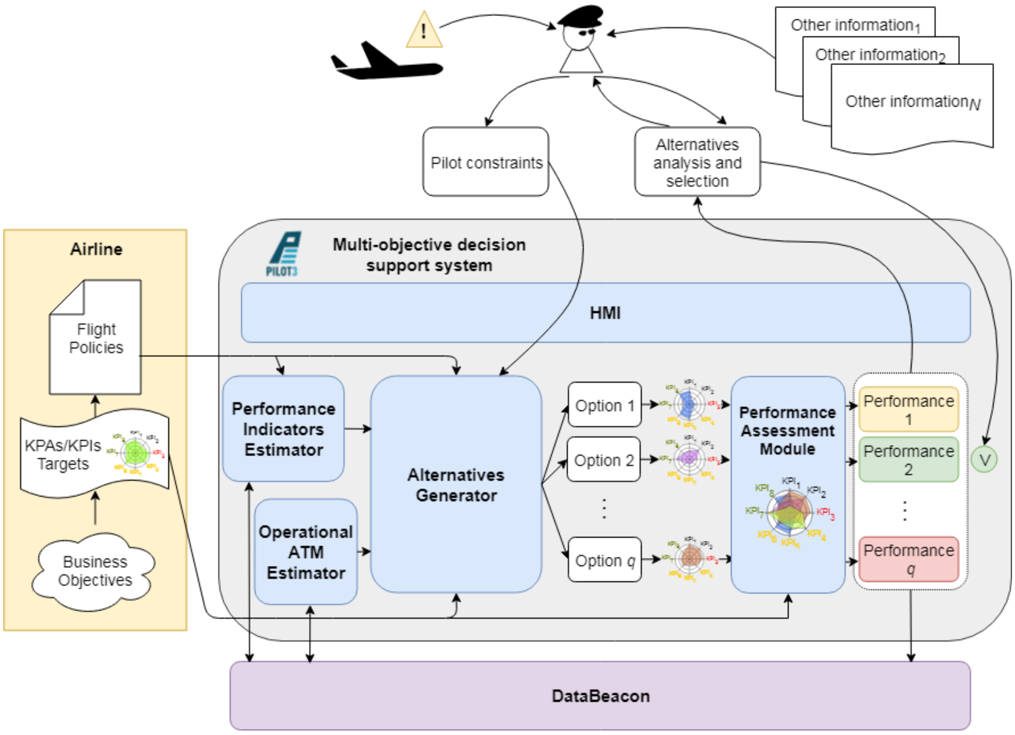

Figure 1. High-level Pilot3 architecture

Pilot3 comprises five sub-systems, namely the:

- **Alternatives Generator**, which will compute the different alternatives to be considered by the pilot; fed by the two independent sub-systems:
    - **Performance Indicators Estimator**, which provides the Alternatives Generator with information on how to estimate the impact of each solution for the different performance indicators (PIs) needed to estimate the optimisation objectives which are relevant to the airline (see Section 1.4);
    - **Operational ATM Estimator**, which provides the Alternative Generator with information on how to estimate some operational aspects such as tactical route amendments, expected arrival procedure, holding time in terminal airspace, distance flown (or flight time spent) in terminal airspace due to arrival sequencing and merging operations, or taxi-in time;
- **Performance Assessment Module**, which, considering the expected results for each alternative on the different KPIs, is able to filter and rank the alternatives considering airlines

and pilots preferences, and to show them to the pilot. This is part of the multi-criteria optimisation process; and

- **Human Machine Interface**, which will present these alternatives to the pilot and allow them to interact with the system.

For more information on the different modules of Pilot3 the reader is referred to D1.1 - Technical Resources and Problem definition (Pilot3 Consortium, 2020).

1.2 Multi-criteria decision making methods families

In a multi-objective optimisation problem, a set of optimal solutions that are equally acceptable from a mathematical point of view (the Pareto optimal solutions) can be reached. Mathematically speaking, the problem is solved when the Pareto optimal set is found. In order to finally select one final solution (or a subset of solutions), this set must be ranked according to some preferences set by the decision maker(s) (DM).

A typical technique to select the *preferred* Pareto solution, is to assign to each individual criterion a given weight (or scalarisation constant), which reflects their priority or relative importance. Then, a linearly weighted sum of the individual optimisation objectives is typically done, yielding to a single compound optimisation criterion, which can be solved with standard (single-objective) optimisation techniques. As mentioned before, the optimisation done in current FMS (and in general by most flight planning or dispatching tools) uses as objective function, a linear weighted sum expressing the relative importance of fuel and time costs, given by a weighting parameter: the Cost Index. As presented below, this corresponds to an *a priori* multi-objective optimisation method.

Although the weighting technique is widely used in many applications (for its apparent simplicity), it presents several important drawbacks. The first one is that choosing the exact values for the different weights (if done beforehand) is not a straightforward task, since it is based either on a somewhat vague intuition of the user about the relative importance of different criteria, or on trial and error experimentation with different weighting values. Another problem is that once they have been established, the optimisation algorithm will find the best solution for that particular setting of weights, missing the opportunity to find other solutions that may represent a considered better trade-off between different criteria.

In this context, it is usual to perform *a posteriori* sensitivity studies, but altering the weighting vectors linearly does not ensure that the values of the objective functions also change linearly, turning these sensitivity studies not obvious to conduct. Furthermore, this method has the limitation that it cannot find solutions in a non-convex region of the Pareto front, which can happen when involving non-linear constraints or objective functions (Miettinen, 1999). More difficulties appear when the objective functions involve summations/subtractions of terms representing different magnitudes (such as noise annoyance, emissions, fuel consumption, flight time, reactionary delay, or missed passenger connections), often with very different scales in their units of measurement (non-commensurable functions). It is true that this can be partially dealt with by normalising the different criteria, so that they all refer to the same scale, but this approach suffers from a subtle problem rarely discussed: in general there are several different ways of normalising (see for instance Marler & Arora, 2005), the decision about which normalisation procedure should be applied tends to be ad-hoc, and different normalisation techniques may lead to significantly different results.

Trying to overcome (some of) these issues, a plethora of multi-objective optimisation methods have been proposed in the last half-century. There are different ways to classify multi-objective optimisation methods, according to different considerations. Here, we adopt the classification presented by (Miettinen, 1999), which is largely accepted in the literature. The classes are:

1. Methods where *a posteriori* articulation of preference information is used (*a posteriori methods*)
2. Methods where no articulation of preference information is used (*no-preference methods*)
3. Methods where *a priori* articulation of preference information is used (*a priori methods*)
4. Methods where progressive articulation of preference information is used (*interactive methods*)

It is worth noting that no classification can be complete and absolute, since overlapping of methods is possible, some methods might be considered to belong to more than one class and some methods are in fact combination of other methods. See for instance (Marler & Arora, 2004), who considers the *interactive* and *a posteriori* methods into a single category.

1.3 Uncertainty in flight management

There are many different sources of uncertainty when considering the optimisation of flight trajectories. This means that the planned and executed trajectories might differ, leading to dissimilar results on the objectives considered by the airline (e.g., cost). Sources of uncertainty include weather, but in the tactical phase notably ATM aspects such as management of flows at TMA (e.g., holding times, path stretching), or taxi-in times (as most of the indicators that are relevant to airlines relate to the arrival time at the gate).

Pilot3 tries to consider some of these uncertainties, weather uncertainty could be considered from forecasts by the Alternatives Generator, and the Operational ATM Estimator module which will produce estimations on operational factors that will affect the trajectory (e.g., expected holding at arrival). However, even these estimations will have some degree of uncertainty (e.g., variance, probabilities or distributions attached to them). In Pilot3, we consider that the system should focus on generating the trajectories that minimise (or maximise) the **expected value in the objectives** identified by the airline. Even with this consideration, including uncertainty might lead to several trajectories being statistically equivalent increasing the alternatives generated by the Alternative Generator.

Section 2.2 presents a review of uncertainty, risk and robustness considerations for optimisation. Note that in this deliverable, uncertainty is only considered with respect to its impact on the multi-criteria optimisation method and on potential considerations for the HMI. How uncertainty will be modelled and captured in the trajectory generator will be considered in detail as part of the modelling activities of WP4.

1.4 Optimisation objectives

The key performance indicators (KPIs) and objectives to be considered in Pilot3 have been selected after consultation with the Advisory Board: first the Advisory Board meeting (attended by relevant stakeholders (airlines, EUROCONTROL) and experts) and then a follow-up survey. After these processes:

4. a set of performance indicators (PI) were identified (e.g., number of passengers missing connections, minutes of delay at arrival);
5. PIs were aggregated into KPIs (e.g., cost of fuel, cost of IROPs); and from these,
6. two objectives that should be considered when optimising the trajectories were derived: **cost** and **on-time performance (OTP)**.

Details on these objectives definition aspects, and how these are considered in the selection of the methods for the multi-criteria optimisation, are presented in Section 3.3.

1.5 Pilot3 configuration and execution logic

This section introduces the configuration and execution logic (including the different optimisation phases) of Pilot3. Prior the flight, Pilot3 engine will be configured (by the airline engineers/dispatchers); then tactically, when analysing and selecting alternatives, the pilot will perform a multi-criteria trade-off analysis. This will be driven by different factors: the airline's business objectives and flight policies, and operational aspects considered by the pilot. The process can be seen as an exploration of alternatives consisting on an optimisation framework that requires to generate trajectories considering objectives and constraints, filtering the solutions and adding/modifying them with operational constraints. **Figure 2** illustrates this logic, which is explained in detail next.

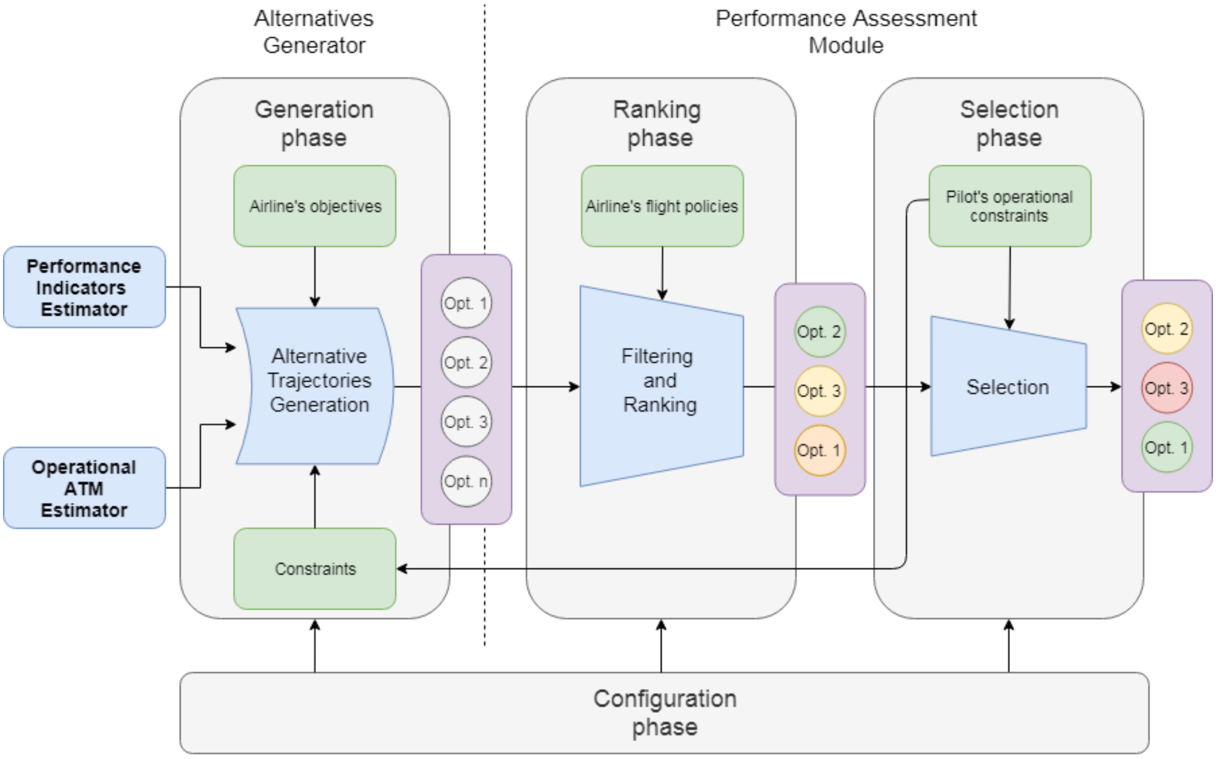

Figure 2. Pilot3 execution diagram with example

EUROPEAN UNION

1.5.1 Configuration

The configuration phase will be performed by the airline prior to the flight. This could be done strategically, or some parameters could be selected at dispatching level on a flight-by-flight basis. The objectives of this phase are to select how the indicators and the operational ATM parameters should be estimated, and to configure Pilot3 to reflect the airline flight policy. For example, indicating if a heuristic or an advanced model should be used to estimate a given parameter with air or ground information, in case of equivalent impact on different indicators, which ones should be prioritised, etc.

1.5.2 Generation phase

In a multi-objective optimisation problem, one might have a set of Pareto optimal solutions (i.e., solutions equally acceptable from a mathematical point of view) and manually assessing all trade-offs arising from various objectives might be a complex and time consuming task. Moreover, different trajectories might lead to equivalent values on the objectives (e.g., two different profiles might produce statistically equivalent expected cost and on time performance). A first automatic generation and selection of candidate solutions will be produced by the Alternatives Generator. The **Alternatives Generato**r uses a trajectory generation engine which considers:

- the objectives function as set by the airline key performance indicators;
- constraints: operational (e.g., airways) and *ad hoc* defined by the pilot (e.g., 'do not provide solutions which imply an altitude change');
- environment data (e.g., weather, aircraft performance); and
- information from the Indicators Estimator and the Operational ATM Estimator on how to estimate these indicators and operational parameters. Note that the performance indicators might not be known until the arrival of the flight, or even until the end of the operational day (e.g., total amount of reactionary delay), and may have to be estimated. These estimations will be made based on the information available at the moment of making the decision. This is the main objective of the Performance Indicators Estimator module: to indicate how to estimate the performance indicators from the trajectory parameters.

As introduced in Section 1.4, the total number of objectives that will be considered in Pilot3 might be small, which might allow for some estimation of the impact in one objective (cost) when achieving the other (OTP).

Moreover, when optimising the trajectories, different optimisation alternatives might be available. This means that the trajectory optimisation might produce a set of alternatives. Note that in some cases, if thresholds or rounding in the indicators are introduced (considering the fuel consumption at a resolution of ten kilograms, or arrival delay at a resolution of one minute, for instance), the number of potential trajectories which are equivalent might increase. Similarly, if uncertainties (see Section 1.3) are considered, their impact on the trajectories (e.g., time required to reach the arrival destination, or total fuel that will be consumed) might imply that some alternatives could be considered statistically equivalent, increasing the pool of trajectories to be considered for their presentation to the pilot.

More information on the generation phase is presented in Section 3.3.2.

1.5.3 Ranking phase

The alternatives' ranking phase is the first part of the **Performance Assessment Module** and it consists on the ranking of the alternatives provided by the Alternatives Generator. This post-processing of the trajectories generated by the Alternatives Generator is performed in order to pre-compute how the alternatives will be presented to the pilot.

This phase will consider the airlines' policies with respect to the different KPIs. For example, two solutions might provide the same cost but trading fuel cost and passenger cost (e.g., one solution might produce lower fuel usage with higher expected cost from compensation due to Reg. 261, while another alternative might use more fuel but reduce the expected cost due to passengers compensation, leading to equivalent total operating costs). In this case, even if the total expected cost for both alternatives is equivalent, the airline might define that passengers should be prioritised. Note that this ranking is produced with the information defined in the configuration phase of Pilot3 and if more than one alternative is produced by the Alternatives Generator.

Details on the ranking phase are provided in Section 3.3.3.

1.5.4 Selection phase

The final step consists in considering pilot operational related aspects. This is the final step of the **Performance Assessment Module** and it considers the interaction with the pilot via the **Human-Machine Interface** (HMI). Information on the trajectories and their impact on the different indicators will be presented to the pilot, who will be able to interact, rejecting solutions or, based on the information provided, adding new constraints and requesting a re-evaluation of the alternatives.

This process if further explored in Section 3.3.4 and in the activities of WP4.

1.6 Methodology to select multi-criteria optimisation method

Since Vilfredo Pareto firstly introduced the concept of Pareto optimality more than 120 years ago (Pareto, V., 1897), hundreds of researchers have addressed the problem of multi-criteria optimisation. Dozens of methods, with subsequent refinements and extensions, have been proposed, especially in the last half-century, with thousands of scientific publications in a wide diversity of applications (Adali & Asik, 2017). Yet, none of these methods or refinements can be said to be generally superior to all the others. In fact, selecting an appropriate multi-objective optimisation method by itself is a multi-objective optimisation problem.

There is a multiplicity of aspects to consider in this selection process, and many of the comparison criteria are difficult to quantify and based on expertise and decision maker preferences to a certain degree. Therefore, the features of the problem to be solved and the capabilities and the type of the decision maker have to be charted before a solution method can be chosen, since some methods may suit some problems and some decision makers better than others. For this reason, the **selection of the method(s) is domain driven**.

From a general point of view, some of the criteria to consider when evaluating methods, were collected in Hobbs (1986) as the following ones:

- *appropriateness*: the method should be appropriate to the problem to be solved,

- *ease of use*: refers to the effort and the knowledge required from the analyst and the decision maker,
- *validity*: the method measures what it is supposed to and the assumptions are consistent with reality,
- *sensitivity of the results to the choice of method*: solutions obtained by the method do not significantly differ from those of the methods.

Some researchers have expanded the number of criteria (up to 28 in Gershon and Duckstein, 1983), but crucial criteria has been reduced to 3 in Stewart (1992):

- input required from decision maker must be meaningful and unequivocal
- transparent method
- simple and efficient method

In Pilot3, a domain driven approach has been followed to perform this selection. Key aspects to consider in this selection have been identified from different sources and grouped in five different categories:

- Data required by the multi-criteria optimisation method;
- optimisation objectives that are considered in Pilot3;
- human machine interaction considerations;
- other functional considerations; and
- other no-functional considerations.

As described in Section 1.5 there are different phases that are part of the multi-criteria optimisation framework developed in Pilot3. Each phase has different characteristics, which means that the optimisation method selected might be different for each phase. However, some of the characteristics of the problem apply to all phases. Therefore, the process to select the method has followed two stage approach: first the characteristics of the problem (tactical trajectory optimisation) are considered with inputs from the proposal (as processed in D1.1 - Technical Resources and Problem definition (Pilot3 Consortium, 2020)), the Topic Manager and the first Advisory Board workshop. This filters the number of potential methods across all the optimisation (or decision-making) phases (generation, ranking and selection phase). Then, for each phase their specific characteristics are considered, and input from a follow-up consultation with the Advisory Board are used to finalise the benchmarking and selection of method process per phase. Section 3 presents in detail this selection process.

1.7 Deliverable structure

The deliverable is structured in five sections:

- Section 1 introduces the context of Pilot3 multi-criteria optimisation for crew support on trajectory management, and the approach followed to select the most suitable methods for the optimisation.
- A literature review on multi-criteria optimisation methods and on optimisation under uncertainty is presented in Section 2.

EUROPEAN UNION

- Section 3 describes with more details the criteria used to select the optimisation methods, and the selection process.
- Conclusions with main findings are collected in Section 4.
- The deliverable closes with next steps and look ahead, with the follow-up activities in Pilot3 project, and with an indication on how the selected methods will be modelled, in Section 5.

The document closes with references and acronyms.

2 State of the art

This section presents a summary of the literature review on the state of the art of multi-criteria optimisation methods (Section 2.1). It also includes a brief review of optimisation under uncertainty (Section 2.2); this is relevant as Pilot3 tool will be subject to different sources of uncertainty.

2.1 Multi-criteria optimisation methods

A domain-driven approach has been used to review the multi-criteria optimisation methods available in the literature. A summary of the main types of methods is this section. A comprehensive list of methods with more details, including some of their benefits and drawbacks, is included in Appendix A.

2.1.1 Methods where *a posteriori* articulation of preference information is used (*a posteriori methods*)

The underlying philosophy of these methods is that the Pareto front is generated first and presented to the decision maker, who will select the most preferred solution among a palette of alternatives. This approach could be useful when it is difficult for the decision-maker to express an explicit approximation of her preferences (see *a priori methods* below). Several *a posteriori* methods are proposed in the literature, as outlined in (Miettinen, 1999; Marler & Arora, 2004). The two principal methods in this class are:

- **Weighting method**, which is a particular case of the scalarisation approach presented above and despite being easy to implement, it might present some difficulties, as mentioned earlier. Some refinements, as response to the inability of the weighted sum method to capture points on non-convex portions of the Pareto front are also proposed in the literature, such as the e**xponential weighted criterion** (Athan and Papalambros, 1996).
- **Epsilon-constraint (or bounded objective function) method**, where one of the objective functions is selected to be optimised and all the others are converted into constraints by setting an upper bound to each of them. The problem is solved many times by changing the value of these bounds. This method can find solutions in non-convex areas of the Pareto front, but can be computationally expensive for certain applications.

Hybrid methods are also possible, either combining the previous two, or introducing weighting functions in compromise programming (see no-preference methods below), such as the weighted **Tchebycheff approach**, which is a popular method for generating Pareto optimal solutions.

These *a posteriori* methods present the advantage that the decision maker does not need to provide any explicit input. Nevertheless, many shortcomings arise with this approach, one of the main ones being simply the difficulty on the generation of the Pareto front which could be computationally too expensive. In this context, if only a limited number of Pareto solutions are presented, these methods can be ineffective in providing an even spread of points that accurately represents the complete Pareto optimal set. Finally, it is quite likely that the decision maker will have some difficulties to select from a large set of alternatives and in many cases, how to present or display these alternatives in an effective way might also be an issue (especially when a large number of objectives are considered).

2.1.2 Methods where no articulation of preference information is used (*no-preference methods*)

These methods can be used when no opinions of the decision maker are sought, or when she cannot concretely define what she prefers. Most of the methods are simplifications of methods included in other classes, typically with the exclusion of the parameters, coefficients, exponents, constraint bounds, etc. that are used to establish preferences. Thus, methods in this class will select one Pareto solution according to some specific criteria/metric and present it to the decision maker, who may either accept or reject it. These methods are only useful if the decision maker does not have any special expectations of the solution and she is satisfied simply with "some" optimal solution. Example methods in this class include:

- **Compromise programming (or global criterion)**, where the distance between some reference point and the feasible objective region is minimised. Several alternatives exist to define either this reference point or the metric for measuring the distances. For instance, the reference point could be the ideal (or utopia) objective vector (i.e., the best we can achieve for each objective function if optimised separately) and the metric the Euclidean distance. Another particular case of this method is the Tchebycheff solution (also known as Egalitarian solution or min-max optimisation), where the maximum distance to the ideal objective vector is chosen as decision performance index, in such a way that the system is no better-off than its worse-off individual.
- **Multi-objective Proximal Bundle Method**, which from a given starting point in the Pareto front moves in a direction where the values of all the objective functions improve simultaneously (Mäkelä, 1993).

2.1.3 Methods where *a priori* articulation of preference information is used (*a priori methods*)

In this case, the decision maker must specify her preferences, hopes or opinions before the process of generating the solution. This can be articulated in many ways: in terms of goals, relative importance of different objectives, etc. It is worth noting that the weighting methods presented above (including the hybrid methods using weights, such as the weighted Tchebycheff approach) could be considered an *a priori* method, if the decision maker specifies beforehand the weights for each objective function representing her preferences. Similarly, the Epsilon-constraint method can also be considered in this class if the bounds for each objective are also set *a priori*.

Although several authors have proposed methods or guidelines to help the decision-maker to set weights (or bounding values) in an effective manner, understanding and correctly interpreting the conceptual significance of the weights is not always obvious for average decision makers. This is indeed the main difficulty of *a priori* methods, since the decision maker might not necessarily know beforehand what it is possible to attain in the problem and how realistic her expectations are. Representative examples of *a priori* methods are:

- **Value function method**, where the decision maker must be able to give an accurate and explicit mathematical form of the value function that represents her preferences globally. In this way, a complete ordering in the objective space is set and a single objective optimisation problem is solved. Although apparently simple, the difficulty of this method is to encode mathematically the real preferences of the decision maker.
- **Lexicographic ordering**, where the decision maker arranges the objective functions according to their absolute importance. Then the most important objective function is optimised (i.e., minimised or maximised). If the problem has a unique solution, this would the solution of the whole multi-objective optimisation problem. Otherwise, the second most important objective function is optimised, but adding a new constraint in the problem to guarantee that the most important objective function preserves its optimal value found in the previous step. If this new problem has a unique solution, this becomes the solution of the whole multi-objective optimisation problem, otherwise the process continues as described above with the remaining objectives. The main drawback of this approach is that it is very unlikely that the process can optimise lower prioritised objectives, since a unique solution is likely to be found in the first step(s) of the process. The method does not allow for a small worsening of an important objective to be traded against a larger improvement of a less important objective, which might be often appealing to the decision maker.
- **Hierarchical approach**, a modification of lexicographic ordering called hierarchical optimisation where the upper bounds obtained when optimising more important objective functions are relaxed by so-called worsening factors. These relaxations allow to trade off higher prioritised objectives in front of lower prioritised ones, exploring in this case, a widest area of the Pareto front containing solutions that can be more interesting to the decision maker. As commented before, setting the relaxation factors might not be an obvious task for the decision maker.
- **Goal programming**, where the decision maker specifies (optimistic) aspiration levels for some of the objective functions (or all of them) forming goals, which are added in form of constraints in the optimisation problem. Then, any deviations from these aspiration levels are minimised. Despite the popularity of this method, there is no guarantee that it provides Pareto optimal solutions and for large problems computational burden could be an issue (Marler & Arora, 2004).
- **Physical programming**, which maps general classifications of goals and objectives, and verbally expressed preferences, to a utility function. It provides a means of incorporating preferences without having to conjure relative weights (Marler & Arora, 2004).
- **Weighting method**, with weights set up beforehand.
- **Weighted Tchebycheff**, with weights set up beforehand.
- **Epsilon-constraint** (or bounded objective function).

As previously, many hybrid methods are also proposed in the literature, for instance, the combination of the weighted and global criterion methods, weighted and the lexicographic approaches, weighted and goal programming, lexicographic and goal programming, etc. Another popular hybrid method is:

- **VIKOR method** (Opricovic and Tzeng, 2004; 2007), which is a combination of compromise programming (see no-preference methods) and a weighting method and was originally developed to solve decision problems with conflicting and non-commensurable criteria. VIKOR ranks alternatives and determines the solution named compromise that is the closest to the ideal from an initial set of (given) weights.

2.1.4 Methods where progressive articulation of preference information is used (*interactive methods*)

If the decision maker has enough time and capabilities to interact with the system, many of the weak points of the previous classes could be overcome. Namely, only part of the Pareto front has to be generated and the decision maker does not have to know any global preference structure, since they are specified as the solution process evolves. At each iteration, some information is given to the decision maker and she is asked then to answer some questions or provide some other type of information. After a reasonable number of iterations, the process stops.

These methods differ by the type and amount of information that is given to (and provided by) the decision maker and how the overall problem is transformed into a single objective optimisation problem. To name a few, in this class we would find the **interactive surrogate worth trade-off method**; the **sequential proxy optimisation technique** (SPOT); the **step method**; the **reference point method**; the **GUESS method**; the **satisficing trade-off method**; the **light beam search**; the **reference direction approach**; and the **NIMBUS method**. Some hybrid approaches combining methods from other classes are also proposed in the literature, such as the **interactive weighted Tchebycheff procedure**, which has the advantage that the decision maker has a rather simple task (if compared with other interactive methods), which is to compare several alternative objective vectors and select the most preferred one.

A particular implementation of the light beam search methodology are the so-called **outranking methods**, a well-established method with a large history of successful real-word applications. The method compares all couples of alternatives and determine which are preferred by systematically comparing the alternatives for each criterion, trying to establish outranking relations between alternatives according on the basis of for how many components the decision maker judges indifference, weak preference, preference or no-preference. These decisions can be complemented, for instance, with veto thresholds, which prevents a good performance in some components of the objective vector from compensating for poor values on some other components. An important advantage of outranking methods is their ability to take ordinal scales into account without converting the original scales into abstract ones with an arbitrary imposed range, and at the same time maintain the original verbal meaning. A second advantage of outranking methods is that thresholds can be considered when modelling imperfect knowledge, permitting the utilisation of incomplete value information, such as judgements on ordinal measurement scales and partial prioritisation (Yanga et al. 2012). Popular examples of outranking methods are ELECTRE (ELimination Et Choix Traduisant la REalité) (Roy, 1968; Figueira et al., 2005) and PROMOTHEE (Brans et al, 1986) families.

A similar approach to outranking methods is the **analytic hierarchy process (AHP)**, primarily based on the pair wise comparison of matrices that the decision maker uses to establish preferences between alternatives for different criteria and the rating methods. This method includes both the rating and comparison methods. Rationality requires developing a reliable hierarchic structure or feedback network that includes criteria of various types of influence, stakeholders, and decision alternatives to determine the best choice (Saaty, 1994). According to the review done by (Sabaei et al., 2015), the

EUROPEAN UNION

most in common methods found in the literature for maintenance management applications are AHP, ELECTRE and PROMOTHEE.

Again, hybrid methods can be used, such as for example the combination of VIKOR and AHP methods, (San Cristóbal, 2011).

2.1.5 Method extensions and refinements

In an effort to overcome the shortcomings and difficulties present in all previous methods, a significant research effort has been devoted to refine them, or to expand them using approaches coming from other fields in mathematics, operational research and engineering. For instance, several normalisation (or function-transformation methods) are proposed by (Marler & Arora, 2010), in order to cope with non-commensurablility aspects. The VIKOR method has been extended by using interval numbers (Sayadi el al. 2009), with the aim to improve the ranking process; or with stability analysis determining the weight stability intervals (Opricovic and Tzeng, 2007).

It is worth noting that due to the fact that human preferences can be often vague, it is hard to estimate an alternative with crisp numerical values. In this context, several authors have extended classical multi-objective optimisation methods with concepts arising from the **fuzzy logic theory**. For instance, (Rao & Roy, 1989) already proposed to use fuzzy set concepts to effectively assign weights to objectives. (Yanga et al. 2012) used linguistic assessments instead of numerical values in such a way that assessments of alternatives with respect to criteria are assessed using linguistic variables whose values are words or sentences in a natural or artificial language in an outranking method. The VIKOR approach has also been extended with fuzzy sets (Opricovic, 2011; Hajiagha et al. 2014), as well as the SPOT methodology (Sakawa & Yano, 1985), to name a couple of extra examples.

2.2 Uncertainty, risk and robustness considerations for optimisation

Nearly every optimisation problem is featured by the presence of uncertainty to some extent. In order to efficiently deal with the disturbances, inaccuracy of input data, or wrongly estimated parameters, a variety of algorithms have been proposed that are capable of producing the solutions that are applicable under certain conditions.

According to Vallerio et al. (2015), two different approaches have been employed in literature when dealing with parametric uncertainty:

1. the first approach accounts for the states and/or parameters probability distribution by **specifying expected value optimisation problems** and chance constraints (Wendt et al.(2002); Mitra (2009); Recker et al. (2012); Li et al. (2002, 2008); Schenkendorf et al. (2009))
2. the second approach is based on formulating a wo**rst case scenario optimisation when the uncertainty is defined by a given set**, e.g., a box or ellipsoid (Nagy &Braatz (2004); Houska & Diehl (2009); Houska et al. (2012) for optimal control and Nagy & Braatz (2003) for an online application)
3. **Other strategies** to quantify and consider the system's variances are reported for example in Diwekar & Kalagnanam (1997); Nagy & Braatz (2007).

2.2.1 Stochastic approach

The first approach mainly involves the different criteria used for risk measure, such as **variance** (Markowitz, 1959), **value-at-risk (VaR)** (Linsmeier and Pearson, 1996), **conditional value-at-risk** (CVaR) (Rockafellar and Uryasev, 2000), etc. For example, Alexanderian et al., (2017) briefly explain the **risk-neutral** and **risk-averse** optimal control approach in the case of partial differential equations (PDEs) with uncertain parameters. The authors consider a real-valued optimisation objective $\Theta(z, m)$ that depends on a control variable z and an uncertain parameter m, both of which can be finite- or infinite-dimensional. Namely, $\Theta(z, m) := \widetilde{\Theta}(z, m, u)$ with $u = \mathcal{S}(z, m)$, where $\mathcal{S}$ is a PDE solution operator. In optimisation under uncertainty problem, it is **natural to seek optimal controls z that make Θ small in an average sense**. For example, a risk-neutral optimal control approach seeks controls that solve:

$$\min_z \mathrm{E}\{\Theta(z, m)\}$$

where $E\{\cdot\}$ denotes expectation over the uncertain parameter m. If we seek controls that, in addition to minimizing the expected value of Θ with respect to m, result in a small uncertainty in Θ, we are led to **risk-averse optimal control.** The authors use the variance of the control objective as a risk measure, and seek optimal controls that solve the problem

$$\min_z \mathrm{E}\{\Theta(z, m)\} + \beta\, \mathrm{Var}\{\Theta(z, m)\}.$$

where $\mathrm{Var}\{\cdot\}$ denotes the variance with respect to m, and $\beta > 0$ is a risk-aversion parameter that aims to penalize large variances of the control objective.

An example of this approach, applied to flight dispatching problems, is found in (González-Arribas, et al. 2018), where a compound objective function it is proposed such that the expected value of the direct operating costs is considered on one hand; and on the other hand, the arrival time window resulting from the earliest/latest arrival times due to weather uncertainty. This cost function is weighted by a "dispersion parameter", that is given by the user (dispatcher), and captures the relative importance given to punctuality with respect to direct operating costs.

This mean-variance formulation is only one of several formulations for finding risk-averse optimal controls. Other examples of more complex risk measures include the value at risk (VaR) and the conditional value at risk (CVaR). Among those, VaR remains the most widely accepted measure among practitioners. VaR estimates the maximum potential loss at a certain probability level, i.e., it provides information about the amount of losses that will not be exceeded with certain probability. Mathematically, $(1 - \varepsilon) - \mathrm{VaR}$ is defined as the minimum level γ such that the probability that the portfolio loss $f(x, u)$ is below γ exceeds $1 - \varepsilon$. Thus, the VaR can be formulated as:

$$V(x) = \min \gamma$$

$$\text{s.t.}\, P(f(x, u) \leq \gamma) \geq 1 - \varepsilon$$

where $x = (x_1, \dots, x_n)$ is the vector of asset weights, and $u = (u_1, \dots, u_n)$ is the vector of uncertain portfolio asset returns.

However, the application of this approach requires a solid knowledge of the data, which are very often prone to errors. Additionally, the probability distributions are often unknown in practice and can be computationally expensive.

2.2.2 Robust approach

In the case when there is no sufficient data, robust optimisation approaches are deemed as an appropriate tool to deal with uncertainty. In robust optimisation approaches, **typically the uncertainty is modeled as parameters whose exact values are not known but are assumed to stem from a set.** This set is called an uncertainty set. Possible realisations of the unknown parameters are called scenarios, which are the elements in the uncertainty set. We call the most typical or expected scenario the nominal scenario and the objective function values in the nominal scenario as the nominal quality.

Although there is not unified classification of the available methods within 'robust optimisation', Goerigk and Schöbel (2016) summarizes the application-driven concepts existing in the literature under different nomenclature. The classification provides the sound collection of nine different robustness concepts given in Appendix B.

2.2.3 Other strategies

In order to reduces the computational intensity of the stochastic optimisation problem, Diwekar and Kalagnanam (1997) proposed a new sampling technique based on Hammersley points. Nagy and Braatz (2007) employed the approaches based on the approximate representation of the full process model using power series or polynomial chaos expansions and provide a qualitative and quantitative estimation of the effect of parameter uncertainties on the states and output variables. However, the proposed approaches have not found the broad application in the practices and thus, will not be further investigated.

3 Selection of methods

In this section the multi-criteria optimisation methods for each phase are selected. Section 3.1 presents the details on the domain-driven selection criteria used in this process. A first general filtering of classes of (and specific) methods considering the characteristics of the problem are described in Section 3.2. Finally, the specific characteristics of each of the execution phases are considered in Section 3.3 to finalise the selection of method for each phase.

3.1 Domain-driven selection criteria

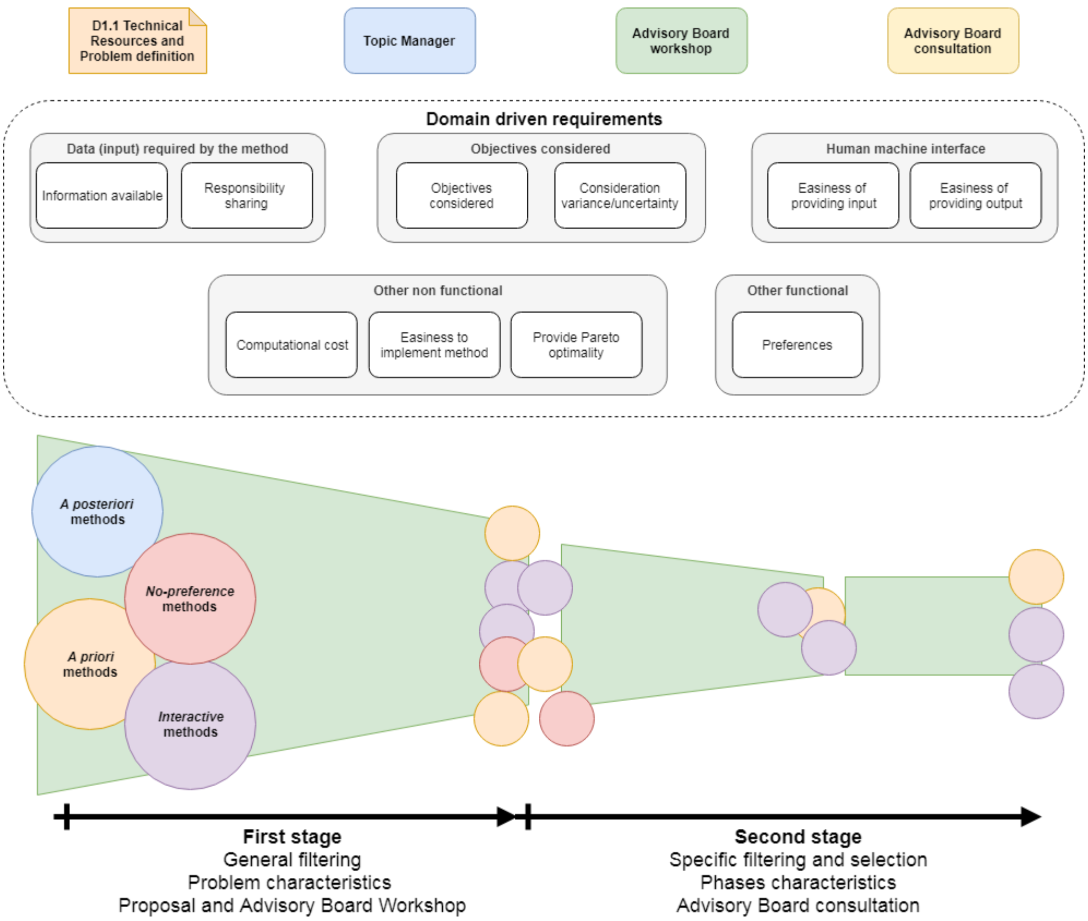

Figure 3. Diagram of approach followed to select optimisation method.

Figure 3 presents a diagram with the approach followed on this selection. As it was introduced in Section 1.6 the selection of a multi-criteria method is typically a complex and domain-driven task.

Pilot3 has used input from different sources to identify the key aspects that should be considered in this selection process:

- **Pilot3 Deliverable D1.1** - Technical Resources and Problem definition (Pilot3 Consortium, 2020): The proposal was further detailed with feedback from the Topic Manager and the Project Officer. This document then defines the problem and scope of the Pilot3 project, outlines the high-level concept and methodology, and identifies the high-level requirements of the Pilot3 software prototype and indicators that will be considered.
- **Topic Manager**: Interaction with the Pilot3 Topic Manager is key to identify some of the requirements from the tool and to ensure alignment with the goals and roadmap of the Clean Sky 2.
- **Advisory Board workshop**: A workshop was held with the Advisory Board (which includes stakeholders -airlines- and experts), where Pilot3 was presented and feedback was gathered on aspects relating to flight operations and performance monitoring, among others.
- **Advisory Board consultation**: A follow-up consultation was conducted to validate the approach followed by Pilot3 on the definition of objectives to be considered, how the information is presented and available, etc. This additional feedback includes as well views from pilots.

The domain-driven selection criteria have been classified into 10 different criteria grouped in five categories:

a. **Data (input) required by the method**

    1. the input needed for the method to function should be available
    2. responsibility sharing on user (dispatcher, pilot) providing the input required

b. **Objectives considered**

    3. ability to deal with high/low number of objectives
    4. consideration of variability/uncertainty

c. **Human-machine interface considerations**

    5. easiness of providing the input required
    6. easiness of providing the output required

d. **Other non functional considerations**

    7. computational cost of the method
    8. easiness to implement the method
    9. the method should provide a necessary and sufficient condition method for Pareto optimality

e. **Other functional considerations**

    10. other general preferences expressed by stakeholders

As described in Section 1.5, there are different phases that are part of the multi-criteria optimisation framework developed in Pilot3. Each phase has different characteristics, which means that the optimisation method selected might be different for each phase.

However, some of the above ten criteria used in this selection apply to all phases and significantly restrict the number of possible candidate methods to be used. Therefore, the selection process used to identify which method(s) will be implemented in Pilot3 follows a two-stage approach:

The characteristics of the problem (tactical trajectory optimisation) are considered with inputs from D1.1 (Pilot3 Consortium, 2020), the Topic Manager and the first Advisory Board workshop. This filters the number of potential methods across all optimisation phases. This is the first stage depicted in **Figure 3**.

Then, the remaining candidate methods were further analysed by the consortium considering the characteristics of each of the phases. And after additional feedback from a consultation with the Advisory Board, a reduced number of methods were kept for the final benchmarking and selection process.

3.2 General filtering

This filtering of potential methods is based on the characteristics of the problem and applied to classes and specific methods. Some of the criteria used to select the method(s) to be implemented significantly restrict the number of potential methods that could be considered.

3.2.1 Objectives considered

3.2.1.1 Number of objectives and their characteristics

It is paramount to understand the number of objectives and their characteristics as this affects the potential type of multi-criteria methods to be used (see selection **criteria b.3 the ability to deal with high/low number of objectives**).

When an airline is operating flights, their flight policies are reflected in their airline **Operations Manual** (OM), which serves as a communication tool that conveys the airline flight policy, aviation department's goals and procedures to the entire company. Information given in the OM is communicated to the crew and flight dispatch personnel through different internal training programmes and communication channels of the airline.

Insight on flight policies from Advisory Board consultation

Although flight policies may vary significantly from one company to another, there is a general consensus among the Advisory Board that these depend highly on:

- **the airline network structure** (hub-and-spoke network vs. point-to-point network),
- **characteristics of a particular flight** (long-haul flight vs. short-haul flight), and
- **type of passengers served** (individual end consumers vs. high-end business travellers).

For instance, a viable connection of its transfer passengers is of the utmost importance for an airline operating hub-and-spoke network. Such airline is highly committed to providing a connecting service to long-haul flights for high-yield premium passengers as well as frequent flyer passengers

EUROPEAN UNION

who contribute with a significant share in the airline's total operating revenue, and are thus highly valued by the airline. On the other hand, flight policies of airlines operating point-to-point networks are rather oriented towards fuel saving.

However, airlines generally allow for **certain level of flexibility in their flight policies** in order to accommodate:

- seasonal traffic characteristics,
- specific flight requirements, and
- pilot's decision (to a limited extent).

The objectives defined in flight policies are translated into operations through the Cost Index (CI). Most policies have a standard component (e.g., default CI set to 10 for all flights), plus a variable component as a function of the flight/event/situation (e.g., override the CI to 30). Note that the CI is used as a proxy to manage/estimate the flight in order to meet the airline's **objectives**.

Insight on optimisation objectives from Advisory Board consultation

During the Advisory Board workshop, the most relevant performance indicators (PIs) which are considered when selecting the major aspects of airlines' objectives and policies were identified. **Six main indicators** were selected as the most important (ordered per relative importance as reported by the Advisory Board):

1. Fuel cost
2. On-time performance(OTP)
3. Passenger missed connections
4. Time in holding
5. (Cost) of passenger disruption
6. Crew and maintenance cost

Fuel cost indicator is relevant, as airlines are highly sensitive to fuel consumption, and fuel costs still constitute a large portion of their total operating costs. Although the sensitivity to fuel costs could vary significantly among the airlines with different business models, there is still a clear consensus that fuel costs will play an important role in the future. Note that for Pilot3 we are interested on the extra fuel cost used or saved tactically due to the management of the trajectory.

Passenger missed connections is of high importance for airlines that operate very complex and large networks as it affects both hard-costs due to compensations (e.g., Regulation 261) and soft-cost as it directly affects the airline reputation. **Passenger disruption costs** are a direct monetisation of the cost due to passenger disruptions, including both, connecting and non-connecting passengers. Therefore, passenger missed connections were identified as a proxy to the cost of passenger disruption due to their large contribution on these costs for flight where these missed connections arise. However, overall these two indicators can be grouped into **IROPS costs**.

In addition to fuel cost, airlines are also keen to minimise other costs, such as **crew costs and maintenance cost**. Airlines may apply a variety of policies regarding crew wages and salaries. However, most of them acquire hourly-based policy in which a pilot is paid based on the hours spent in the air or/and on the ground. With strict policies regarding pilot working hours in place, disruptions may lead to increased crew costs and additional scheduled inefficiencies. Additionally, regular aircraft maintenance checks (A, B, C and D) are performed after predefined flight hours, requiring a large majority of the aircraft's components to be inspected and/or replaced. In order to reduce maintenance costs, a number of airlines leases aircraft.

The **time in holding** is up to a large extend out of the control capabilities of airlines. However, the prevalence of holdings and of sequencing and merging procedures (e.g., tromboning) could lead to sub-optimal decisions (e.g., speed up a flight to recover delay to end up in a holding stack). This is therefore not an indicator where airlines can act to reduce, but a parameter that should be considered when optimising the trajectories as part of the uncertainty in the system.

In addition to fuel costs, airlines are also concerned about the **on-time performance**, as this indicator is very often used to reflect the level of service provided to passengers. Nowadays, on-time performance is being monitored on a flight basis by most airlines in order to verify compliance with on-time performance targets defined in their respective airline flight policies.

After analysing the different indicators, it was deemed that four of them (fuel cost, passenger missed connections, cost of passenger disruption and crew and maintenance cots) could be directly translated into **cost**. **On-time performance** is a binary indicator (i.e., the flight arriving to the destination gate with a delay with respect to their schedule lower than 15 minutes) which was difficult to monetise (i.e., linked with level of service, reactionary effects, robustness in network among others), and therefore is kept independently. Finally, time in holding, is considered as part of the uncertainty in the optimisation.

More information on the process of consultation with the Advisory Board will be provided in D3.1 – Airlines data collection report.

It has been established that the main high-level objectives relevant for an airline can be reduced to only two:

3. **Cost**: which is a complex objective build from the aggregation of three key performance indicators (KPIs):

    a. cost of fuel,

    b. cost of IROPS, including hard and soft passenger costs (considering connecting and non-connecting passengers), and

    c. other costs, which account for extra crew and maintenance costs, but most importantly for reactionary costs.

4. **On-Time Performance (OTP)**: which is considered as a binary variable of achieving on-time performance (i.e., arrival delay ≤ 15 minutes or not)

Therefore, from an optimisation point of view, two objectives (cost and OTP) are considered. Note that to estimate the cost objective, its components need to be estimated and to estimate these, low level indicators (e.g., number of passengers missing connection) will need to be estimated too. This is the role of the Performance Indicators Estimator.

Validation of choices of top level KPIs with Advisory Board
After the first meeting with the AB (February 2020), 6 main KPIs had been identified (fuel cost, on-time performance, passenger missed connections, time in holding, (cost) of passenger disruption, crew and maintenance cost). Since most of them could be expressed as cost, they were reduced to only cost (comprising different components) and OTP (i.e., either the flight achieves OTP or not). This approach was validated by a second follow-up consultation with the AB.

The characteristics of these two objectives has a significant impact on the Pareto analysis of solutions. Since OTP is a binary objective function, the problem yields to 0, 1 or 2 possible Pareto efficient solutions. It could be case that the given constraints on the trajectory could make the optimisation infeasible. However, this should not be the typical situation, unless the Pilot is interacting with the system asking for potential solutions while setting different operational constraints in altitude, speed etc. In some cases, it would not be possible to tactically recover enough time to achieve the OTP objective. Therefore, only one Pareto efficient solution exists and Pilot3 should focus on minimising the total cost. In other cases, a trade-off might exist between achieving OTP and reducing the cost.

Figure 4 presents an example of one case where such trade-off exists. There are a set of trajectories that do not achieve the on-time performance, each one of them with an expected different cost; and a set of trajectories which meet the on-time performance with another set of costs. As shown in the Figure, the two highlighted points are the Pareto optimal solutions.

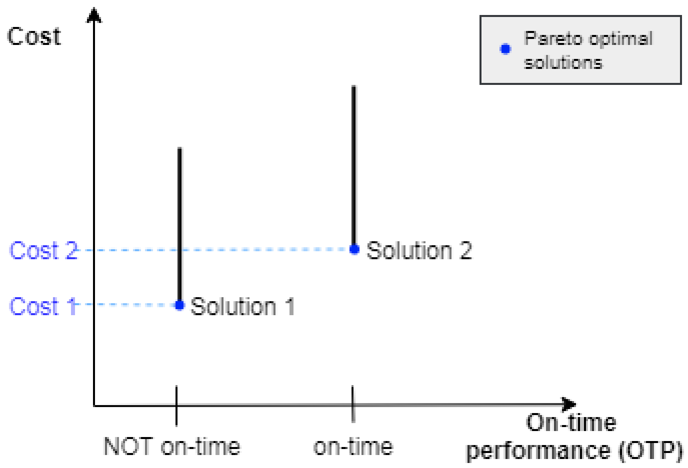

Figure 4. Pareto optimal solutions example Cost vs. OTP

Insight on on-time performance from follow-up Advisory Board consultation

The follow-up consultation with the Advisory Board confirmed that providing to the pilot information on the 'extra-cost' of achieving OTP with respect to the minimum cost, which does not respect the OTP, i.e., difference in cost between Solution 2 and Solution 1 in **Figure 4** would be desirable.

Due to the nature of the multi-objective optimisation problem, with only two objectives and up to two possible Pareto optimal solutions, the method to address it becomes almost trivial and straightforward: Pilot3 will first optimise the trajectory considering only cost. If on-time performance is not achieved with the optimised solution, then the Alternatives Generator will try to impose achieving OTP as a constraint, to compute the extra cost that it would represent (i.e., finding Solution 2 from **Figure 4**), in case this is achievable.

Finally, note that even if the problem has been reduced in terms of objectives, trade-offs might still occur with respect to the KPIs that compose the Cost objective (i.e., cost of fuel, IROPS and other costs), and that irrespectively of these potential trade-offs, more than one trajectory/solution could lead to a statistically equivalent total cost. Therefore, the trajectories generated by the Alternatives Generator could still be more than two solutions.

3.2.1.2 Consideration of uncertainty

As discussed previously, Pilot3 will operate in an uncertain environment (e.g., weather, holding, path stretching, taxi in times). In case of uncertainty, an extra objective could be added to the optimisation, which is the risk of the solution (**criteria b.4 consideration of variability/uncertainty**). This is typically expressed as the variance of KPIs at the optimal point. For instance, a solution saving in average 50 kg of fuel may be risky, because, for example, due to the uncertainty the saving can go down to 10 kg, or even be negative. Another seemingly less optimal solution, which would expect to save for instance 40 kg, may be less risky, ending up in 30 kg of fuel saved. As a consequence, there is sometimes a trade-off between the average and the variance of the objective function.

However, it is important to realise that **airlines are profit-driven entities**, i.e., that they seek to minimise their expected cost (apart from the OTP objective). As a consequence, if the underlying statistical distributions are correctly specified, the variance should be implicitly taken into account in the **expected** objective function, which thus should not include any extra variance term. This is typically referred as a risk-neutral optimisation, as presented in Section 2.2.1. **Since Pilot3 should support the crew by providing solutions which would produce the best outcome in average, the optimiser will be risk-neutral**.

Because companies are made of humans, and humans are notoriously risk-averse, their decisions sometimes reflect this bias, a fact that we acknowledge in Pilot3. As a consequence, while our objective function will not include any explicit variance term, we will monitor the level of risk (the variance) as an extra indicator, which could be presented to the crew. This may for instance lead the pilot to decide to change option because of the (perceived) level of risk. This is in line with the feedback obtained from the consultation with the Advisory Board.

Note that factors related to robustness, such as changes needed due to modifications on the forecast of the operational parameters (e.g., receiving a weather forecast update), are out of scope of the project. Pilot3 will use the most up-to-date available information to estimate uncertainties (e.g., latest available weather forecast) and the tool can be re-executed if this information is updated; but the fact that this information might be updated is not considered when the tool is executed.

3.2.2 Data required

Once KPIs have been identified, it is important to capture the preferences of the airlines of how to provide input necessary to the optimisation, in order to ensure the appropriate multi-criteria decision-making method to be selected (see criteria **a.1 the input needed for the method to function should be available**, **a.2 responsibility sharing on user (dispatcher, pilot) providing the input required**, **c.5 easiness of providing the input required** and **e.10 other general preferences expressed by stakeholders**).

Experiments in psychology show that the amount of information provided to the decision maker has a crucial role (Kok 1986). Though more information may increase the confidence of the decision maker in the solution obtained, it may also lead to less percentage of the information used, and thus the quality of the solution may be worse. In this context, some considerations on the visualisation of the results should also be considered (**c.6 easiness of providing the output required**). The graphical representation of solutions and the human machine interface with the decision-maker in general plays an important role and constitutes an important challenge itself.

Insight on preferences of providing input from Advisory Board consultation

While airlines generally have a clear idea that **on-time performance** is important for them and they can easily perceive that arriving early/late is not desirable, they acknowledge it is very hard to **quantify** this in terms of cost, since the implications of arriving early/late are many (i.e., waiting for gate, handling stuff, delay for passengers, crew stuff, etc.). For this reason, OTP is kept as an independent objective (not monetised). This also implies that it is difficult to compare objectives between them in a quantified manner.

If instead of the general objective of cost, different PIs and KPIs are considered in order to identify if priorities can be considered among them, the Advisory Board expressed the following concerns:

- In the case of quantifying cost of **passenger missed connections**, it seems that there is a discrepancy among airlines depending on their size and **network structure**. Namely, the larger airlines that operate through code-share agreements, as well as airlines that operate long-haul flights have more difficulties in quantifying this PI. On the other hand, airlines with point-to-point network can translate this indicator into costs much more easily.

- A similar conclusion can be drawn for **crew and maintenance cost**. It seems that quantification of these costs is highly related to the specific airline business model and policy adopted. The airlines may have different strategies in terms of the maintenance agreements and the type of outsourcing arrangements, which may affect their respective cost structure. However, there is a general consensus that the maintenance cost is more difficult to estimate than crew costs but that both are less relevant than the other indicators.

- **Targeting given values for PIs or quantifying them seems difficult for airlines**.

- **Ranking** the PIs could be another possibility, as this is highly related to airlines business model and their respective flight policies.

From the input of the Advisory Board in the workshop and with the follow-up consultation, it was clear that ranking of importance of KPIs was the only easily available input for the decision maker. Also, this ranking should be defined as part of the configuration of Pilot3 (e.g., at dispatcher or strategic level).

3.2.3 Pre-selection of classes of optimisation methods

Despite the computational power available today in ordinary desktops, tablets, mobile phones, etc. or using large infrastructure/architectures such as high-performance computing, grid-computing, etc., the **computational cost** of finding an acceptable solution to the multi-criteria optimisation problem may still be a limiting factor, especially for (quasi) real-time applications and/or large problems with many objectives and constraints. For real-time calculations needed in Pilot3, the computational cost of *a posteriori* method is assumed to be prohibitive (criteria **d.7 computational cost of the method**). This, and the fact that the decision maker would have difficulties selecting from a large panel of Pareto solutions (criteria **c.6 easiness of providing the output required**) were important enough reasons to disregard *a posteriori* methods for the present application.

Though they are quite simple to implement, no-preference methods (when used only) are also discarded, since the Pilot3 application seeks some implication of the decision maker (criteria **d.8 easiness to implement the method**, **e.10 other general preferences expressed by stakeholders**). Nevertheless, some of the interesting features of the particular method of *compromise programming* could be used along with a complementary method.

A priori and interactive methods are then left to solve our optimisation process, which, as described in Section 1.5, consists of the following phases:

1. From airline flight policies obtained in the strategic phase of Pilot3, a (reduced) subset of alternative trajectories is generated by the Alternatives Generator (**Generation phase**).
2. Once this set of alternatives has been generated, there is a process of ranking and selection, which will be performed by the Performance Assessment Module (**Ranking phase**). The objective of this process is to filter and rank the alternatives considering airlines' policies with respect to the different KPIs. For example, two solutions might provide the same cost but trading fuel cost and passenger cost (e.g., from compensation due to Reg. 261), in this case, even if the total expected cost for both alternatives is equivalent, the airline might define that passengers should be prioritised. Note that this ranking is produced with the information defined in the configuration phase of Pilot3 and if more than one alternative are produced by the Alternatives Generator.
3. The final step consists on the inclusion of pilot operational related aspects (**Selection phase**). The pilot must have a mechanism allowing to compare and rank the solutions. For example, some solutions might be dismissed, as not deemed adequate from an operational perspective. Others, if acceptable, might not be their preferred solution as, for example, the required workload might be too high (e.g., they require a high interaction with the ATC). An iterative approach where the pilot can prioritise the alternatives and add constraints could be considered. This might trigger another optimisation of trajectories with these new constraints.

Based on this description***, a priori* methods seem more suitable for the Generation phase**, where flight policies are set up beforehand, whereas **interactive methods could be of used for the Ranking and the Selection phases** where additional input from the decisions makers could be obtained.

3.3 Specific filtering and selection

3.3.1 Shortlist of optimisation methods

In Appendix A, some of the criteria indicated in Section 3.1 are considered to further reduce the number of possible methods. For example, within the available *a priori* methods, goal programming combines the drawbacks of not always leading to Pareto solution (criteria **d.9 the method should provide a necessary and sufficient condition method for Pareto optimality**), and the fact that, though goal-setting seemed at first to be an understandable and easy way of making decision, feedback from the AB would show that it is not easily computable (criteria **a.1. the input needed for the method to function should be available** and **c.5 easiness of providing the input required**).

After the follow-up consultation with the Advisory Board, it was made clear that airlines would not be able to provide numerical targets for KPIs, nor numerical bounds, nor relative weights of importance between KPIs. Only ranking of importance would be an available input from the decision maker.

Based on these criteria, the following optimisation methods were identified as suitable candidates, for *a priori* methods:

- **Lexicographic ordering**: The decision maker arranges objective functions according to their absolute importance. Then the most important objective function is minimised (or maximised). If the problem has a unique solution, it is the solution of the whole multi-objective optimisation problem. Otherwise, the second most important objective function is minimised, but adding a new constraint in the problem to guarantee that the most important objective function preserves its optimal value found in the previous step. It can be continued if there are more than two objectives.
- **Hierarchical approach**: it is a modification of lexicographic ordering, where the upper bounds obtained when minimising the most important objective function are relaxed by so-called worsening factors. These relaxations allow to trade off higher prioritised objectives in front of lower prioritised ones, exploring in this case, a widest area of the Pareto front containing solutions that can be more interesting to the decision maker.

These methods could be used in the Generation phase of the optimisation process: from airline flight policies obtained in the strategic phase of Pilot3, prioritisation of cost or of OTP is decided (with or without trade-off) and a (reduced) subset of alternative trajectories is generated by the **Alternatives Generator**.

After this phase, several alternative trajectories may have been obtained leading to equivalent values of both objectives (cost and OTP), but showing differences with respect to other KPIs such as cost of fuel, IROPS, etc. Once this set of alternatives has been generated, ranking and selection is performed by the **Performance Assessment Module** with interaction with the **Human-Machine Interface**. The ranking of alternatives is based on airline preferences in term of cost of fuel, IROPS and other kinds of cost. In the Ranking phase, using additional input from airline policies and in the Selection phase, where the pilot would have a mechanism allowing to compare and rank the solutions, the following interactive methods may be used :

- **VIKOR**: this is a combination of compromise programming (from the no-preference method family) and a weighting method. VIKOR ranks available alternatives and determines the solution named compromise that is the closest to the ideal from an initial set of (given) weights. Though initial weights of relative importance of the attributes would be needed, and that in our case they seem impossible to obtain directly from airlines, these weights may be computed using analytic hierarchy process.

- **Analytic Hierarchy Process** (AHP): AHP generates a weight for each evaluation criterion (or sub-criteria) according to the decision maker pairwise comparisons of criteria. The higher the weight, the more important the corresponding criterion. Next, for a fixed criterion, the AHP assigns a score to each alternative solution according to pairwise comparisons of the alternatives based on that criterion provided by the decision maker. The higher the score, the better the performance of the option with respect to the considered criterion. Finally, AHP combines the criteria weights and the alternative scores, thus determining a global score for each alternative, and a consequent ranking. The global score for a given alternative is a weighted sum of the scores it obtained with respect to all the criteria. It can either be used alone or combined with VIKOR method. It is applicable to our case given that the number of considered criteria and available trajectories would be quite limited; indeed, for problems with many criteria and available alternatives, it may require a large number of evaluations by the user.

3.3.2 Generation phase

This phase aims at generating a (reduced) subset of alternative trajectories based on preferences from airline flight policies obtained in the strategic phase of Pilot3. As previously commented, preferences can only be ranked and the main two objectives to consider from the optimisation point of view are cost and OTP. Though cost seems to always be the most important factor, the consultation with the Advisory Board showed that presenting the potential trade-off required to achieve OTP was of their interest.

Insight on trade-off possibility between KPIs from the Advisory Board

When asked whether they would prefer to be presented with two options, including the cost trade-off for OTP, (e.g., Alternative 1: cheapest but not achieving OTP, and Alternative 2: 10 EUR higher than Alternative 1 but achieving OTP), or if they would only be interested in minimising the cost, members of the AB showed interest in being presented the cost trade-off for OTP. For example, they acknowledged that they may prefer to go for a high fuel consumption in a flight that may avoid cancel the following flight.

For this purpose, either the pilot or the dispatcher should validate the value of the maximum extra cost allowed to achieve OTP.

Based on this feedback and taking into account the multi-objective optimisation problem can present up to two different Pareto points, we propose, for the Generation phase, to compute both (if they exist). The **Lexicographic ordering** approach will be used to generate these Pareto optimal solutions. As one of the objectives considered is binary (either achieving or not OTP), using the lexicographic ordering allows us for an exploration of the Pareto front as in an *a posteriori* method, corresponding in this case to an *a posteriori lexicographic ordering* used, where the two possible combinations of the objective rankings are considered:

1. Considering as **first objective the total cost** and as **second objective the achieving of OTP**. This will provide at least one possible trajectory (note that several could lead to equivalent total

cost) which minimise the total cost (first objective); and if OTP is reachable with that cost, only the trajectories meeting this criteria will be generated (second objective). This strategy is, therefore, robust against potential local minima issues: in situations with a flat Pareto front (i.e., **Figure 4** with Cost1=Cost2) ensuring that solution selected minimises cost and also achieves OTP.

2. If OTP is not achieved during the first step, then a possible trade-off might exist between cost and OTP. To generate this possible point, achieving **OTP will be set as a constraint** (first objective fulfilled) and cost will be then minimised (second objective). The computed trajectory(ies), if exist, will be kept as a trade-off alternative(s) to the one(s) generated in the first step (i.e., they will have a higher cost than the previous ones but will achieve OTP). Note that in some cases this might not be possible (i.e., there are no trajectories which can ensure OTP as the delay is too high).

The Generation phase thus aspires to provide several alternative trajectories leading to equivalent minimum total costs or at a higher cost but allowing reaching OTP.

3.3.3 Ranking phase

Once several trajectories have been generated, the first process in the Performance Assessment Module is to rank these trajectories based on additional cost KPIs in order to further reduce the number of optimal trajectories to present to the crew considering airlines policies. This phase would be useful if airlines have interest in prioritising some specific cost (e.g., cost of fuel) at the expense of others (e.g., IROPS cost). To capture better their preferences, the advisory board was consulted.

Insight on the possibility of ranking KPIs within the total cost (e.g., cost of fuel, IROPS) from the Advisory Board

When asked whether they would be interested in ranking the different components of the cost within available trajectories (leading to equivalent costs), or only interested in total cost regardless of the sub-components of this cost, members of the AB showed interest in this disaggregation since it may be interesting for decision making, both in planning and execution phases, and also because though the total cost could be the same there could be better brand perception or less risky choice in an option or another.

The Ranking phase will then aim to disaggregate total cost into sub-cost and to provide ranking of the alternative trajectories based on preferences established by airline policies. Depending on how these preferences can expressed, one or another optimisation method can be used. It is thus fundamental to capture if and how airlines can share these preferences. Since it was already established that ranking of KPIs was the only easily available way of sharing preferences, it is interesting to know if more detail can be obtained by ranking KPIs two by two and not only globally (see criteria a.1 the input needed for the method to function should be available).

Insight on the availability of information about relative importance of KPIs from the Advisory Board

Members of the Advisory Board ensured that they should be able to rate the most important KPIs two by two.

EUROPEAN UNION

For example, they should be able to decide what cost component is more important between fuel and IROPS:

- fuel is the more important; or
- IROPS is the more important; or
- fuel and IROPS are equally important.

Nevertheless, a more detailed grading of relative importance (e.g., indicating if this importance is moderate, strong, very strong) and or numerical relative importance on a scale (e.g., form 1 to 5), was deemed too complex.

To sum up, when starting the Ranking phase of the optimisation process, several alternative trajectories have been generated and the objective is to rank and select them following the preferences of the airlines. The multi-criteria optimisation problem is now a discrete problem, involving a limited number of alternatives, and the decision maker is able to rank two by two the KPIs of interest (OTP, cost of fuel, IROPS, other costs). Based on this, we propose to use the **Compromise Ranking Method**, also known as the **VIKOR** method, which is improved by introducing the **Analytical Hierarchy Process** for assigning the weights of relative importance of attributes (San Cristóbal, 2011).

The VIKOR method is an effective tool in multi-criteria decision making, particularly in situations where the decision maker is not able, or does not know to express their preference at the beginning of system design. Here, even though airlines are able to express their preferences, when coming to cost components it is not always an obvious decision to decide which one is the overall most important (if any). The VIKOR characteristics match problems with the following criteria (Opricovic and Tzeng, 2007):

- Compromising is acceptable for conflict resolution: at this stage of the problem, we believe this is the case, else, if a single KPIs overpasses all others, the selection of the corresponding trajectory is obvious.
- The decision maker is willing to approve solution that is the closest to the ideal (ideal or utopia point would correspond to the minimum possible value of each cost KPI).
- The criteria are conflicting.
- The alternatives can be evaluated according to all established criteria (performance matrix): here all KPIs can be computed for each trajectory.
- The decision maker's preference is expressed by weights, given or simulated: here the two by two preferences between KPIs will allow to assign weights of relative importance of KPIs using Analytical Hierarchy Process.

The VIKOR method is thus appealing as it fits most of the challenges encountered in this optimisation problem. When applied to a given set of alternatives, the obtained compromise solution aims to provide a maximum group utility of the *majority* (by minimising the weighted sum of the differences between KPI values and their respective minima), and a minimum individual regret of the *opponent* (by minimising the maximum difference between a KPI value and its minimum).

In a very simple toy example, we aim to illustrate how this method can be applied.

EUROPEAN UNION

Toy example

Let us suppose that seven trajectories were obtained from the Generation Phase: three trajectories reaching OTP at a total cost of 2,100 EUR, and four trajectories not reaching OTP at a total cost of 2,000 EUR. Costs of these trajectories are now disaggregated into sub-cost (KPIs) and can be presented in the so-called performance matrix (see Table 1).

Table 1. Performance matrix of VIKOR toy example

Alternative Trajectory	Cost of fuel (minimise)	IROPS cost (minimise)	Other costs (minimise)	OTP (maximise)
Trajectory 1	750	700	650	1
Trajectory 2	750	650	700	1
Trajectory 3	800	625	675	1
Trajectory 4	700	750	550	0
Trajectory 5	725	700	575	0
Trajectory 6	675	725	600	0
Trajectory 7	750	675	575	0

The optimisation problem is then split into two separated problems: rank the best options for OTP and rank the best options for minimising the total cost. To that end, ranking preference input is obtained from the airline policies (pre-flight) under the following form:

- IROPS cost is more important than cost of fuel.
- Cost of fuel is more important than other costs.
- IROPS cost is more important than other costs.

Based on this two-by-two ranking of relative importance of KPIs, the optimiser would assign the corresponding weights to each of the KPIs (using Analytical Hierarchy Process). In this example, highest weight of importance is assigned to IROPS cost, followed by cost of fuel and finally other costs. Note that if the airline were willing to, it could give more details of the relative importance (e.g., cost of fuel is *strongly* more important than other costs, IROPS cost is *moderately* more important than cost of fuel, etc.) or rate them with a numerical scale, in order to obtain a finer tuning of the relative weights of the KPIs. But, as commented from the insight of the Advisory Board, if this level of details is too complex to obtain the optimisation process would take place using only basic relative importance.

Then the Compromise Ranking Method (VIKOR) would be applied, which aims to reach a compromise between all KPIs being as close as possible to their optimal values for the present set of trajectories and minimising the maximum difference between current and optimal value of KPIs. After this process, a single solution can be obtained if it really has a consequent advantage over the others, or a set of compromise solutions if these are too much alike to be able to choose only one.

In this example, for the OTP trajectories, a set of two compromise solutions would be obtained by the VIKOR algorithm in the following order: trajectory 2, followed by trajectory 3. Though trajectory 2 is the best ranked, its advantage over trajectory 3 is not considered consequent enough, which is why this set of two solutions is proposed. But the advantage of these two solutions over trajectory 1 is big enough to eliminate this option.

For the non-OTP trajectories, the VIKOR algorithm produces the following compromise solutions: trajectory 7 is the best ranked, followed by trajectory 5. Once again, the advantage of the first ranked (trajectory 7) is not substantial enough to discard the second one (trajectory 5), but the two remaining trajectories can be disregarded based on this optimisation process.

In this toy example, from a set of seven cost-equivalent (OTP or not) trajectories, the optimisation process conducted in the Ranking phase returns two ranked OTP trajectories and two ranked non-OTP trajectories. This thus reduces the set of options presented to the pilot and presents additional ranking information based on airline policies with respect to the results of the Generation phase.

3.3.4 Selection phase

The final phase of the execution of Pilot3 is the selection phase. This will be the final phase of the Performance Assessment Module and rely on the interaction with the pilot via the HMI.

The pilot must have a mechanism for the comparison of the solutions. The Alternative Generator will create trajectories which will be evaluated, filtered and ranked, as previously described. However, the pilot might want to further explore the implications of the solutions on the different performance indicators and on the required trade-off to achieve OTP if possible. The information provided to the crew should be simple and, as much as possible, predictable in its presentation, so that the pilot can easily understand the different trade-offs and make an informed decision. The crew will be able to obtain information on the high-level objectives but also on the different KPIs (e.g., cost of fuel, cost of IROPs), and even descend to the level of indicator (e.g., number of missed connections). This is required so that they have a full understanding of the alternatives.

Insight on information to provide to crew from Advisory Board consultation

It was made clear in the consultation that there was an interest on providing information on the trade-off required to achieve OTP with respect to extra cost. Therefore, this information will be provided to the crew. However, who should accept the trade-off was split between the crew independently or the airline operating centre (i.e., dispatchers). The most shared view was that the thresholds used on this decision process should be defined before the flight (as part of the configuration of Pilot3). This, however, seems to be dependent on the airline flight policies.

The disaggregated information of cost between its components is deemed relevant by the Advisory Board. Most of the respondents agreed that providing information on the variance on the solution (and not only the expected value) was interesting. However, it was also pointed out that in some cases the simplest the solution the highest the acceptance of the tool. Therefore, this might be a parameter (present the information or not) subject to configuration by the airlines.

Besides the impact of the different alternatives on the performance indicators, tactical operational aspects will be considered by the crew. This might affect the acceptability of some of the suggested solutions which might be dismissed by the crew. Moreover, even if acceptable, operational preferences might be considered (e.g., alternatives not preferred by the crew due to an expected high workload due to interaction with the ATC to obtain the required clearances).

These operational aspects could be captured by explicit tactical operational indicators, which could be considered as part of the optimisation (e.g., including as an objective the minimisation of number of flight level changes). However, after the interaction with the Advisory Board and the follow-up consultation with pilots, it is considered that this might be too difficult to capture by the Pilot3 engine. These factors are embedded with pilot preferences, knowledge, operational awareness, etc. For this reason, an **iterative approach** where the pilot can prioritise the alternatives and add constraints is preferred. Pilot3 will, however, be able to compute some estimation of relevant operational indicators to facilitate this selection process (e.g., number of flight level changes required). This will be computed *a posteriori*, i.e., once the trajectories have been generated, and not as part of the optimisation.

Insight on tactical information considered to accept solutions and add constraints from Advisory Board consultation

Some of the aspects highlighted include the need of use up to date information when optimising the trajectories and all safety related parameters (e.g., aircraft performance, fuel on board, meteorological conditions). The number of specific indicators that the crew could use to determine if they think that a solution is acceptable are crew and flight dependent. However, some indicators can be pre-computed and provided as information to the crew. Example of parameters that are considered relevant by pilots are:

- deviations on trajectories characteristics (e.g., variation in distance, fuel, time with respect to a baseline trajectory);
- difference in wind component and temperature deviation from baseline;
- number of speed changes required;
- number of flight level changes required;
- reported known issues along the new trajectories (e.g., turbulence, icing);
- estimation of workload due to ATC interventions;
- minimum temperature on new trajectories;
- ability to rejoin the original route.

Adding constraints (or even manually defining a new trajectory) might trigger another generation of trajectories with the pertinent execution of the Generation phase. Note that the newly generated trajectories could still be compared by the pilot with the non-dismissed previously generated. The Ranking phase will then be able to rank and filter the newly generated trajectories along with the previous ones. Note that, if constraints are added, it is possible that these new trajectories will perform worse, with respect to the objectives (cost and OTP), than the original set. Therefore, the Ranking phase must ensure to keep these trajectories (i.e., not filter them out), so that the crew can compare them with the previous (not-dismissed) solutions.

Overall, this Selection phase depends on the interaction of the pilot via the HMI, which will be further developed as part of WP4.

4 Conclusions

There are many different multi-criteria decision techniques that could be used when dealing with more than one objective. This deliverable has followed a domain-driven approach to select the most suitable methods for each of the Pilot3 execution phases. This has been done considering inputs from different sources:

- **Pilot3 Deliverable D1.1** - Technical Resources and Problem definition (Pilot3 Consortium, 2020),
- **Topic Manager**,
- **Advisory Board workshop**, and
- **Advisory Board consultation**.

The information gathered has been used to obtain information on ten different criteria grouped in five categories:

a. **Data (input) required by the method**
    1. the input needed for the method to function should be available
    2. responsibility sharing on user (dispatcher, pilot) providing the input required

b. **Objectives considered**
    3. ability to deal with high/low number of objectives
    4. consideration of variability/uncertainty

c. **Human-machine interface considerations**
    5. easiness of providing the input required
    6. easiness of providing the output required

d. **Other non functional considerations**
    7. computational cost of the method
    8. easiness to implement the method
    9. the method should provide a necessary and sufficient condition method for Pareto optimality

e. **Other functional considerations**
    10. other general preferences expressed by stakeholders

First, the characteristics of the problem have been considered to filter the number of potential methods across all the optimisation phases. Then, remaining candidate methods were further analysed considering the particularities of each of the execution phases.

In particular, it has been stablished that established that the main high-level objectives relevant for an airline can be reduced to only two:

1. **Cost**: which is a complex objective build from the aggregation of three key performance indicators (KPIs):
   a. cost of fuel,
   b. cost of IROPS, including hard and soft passenger costs (considering connecting and non-connecting passengers), and
   c. other costs, which account for extra crew and maintenance costs, but most importantly for reactionary costs.
2. **On-Time Performance (OTP)**: which is considered as a binary variable of achieving on-time performance (i.e., arrival delay ≤ 15 minutes or not)

Pilot3 should support the crew by providing solutions which would produce the best outcome in average, the optimiser will be **risk-neutral**. For this reason no further objectives linked to uncertainty will be added.

Note that the details on the optimisation technique that will be used to optimise the trajectories considering the multi-criteria framework will be defined in WP4. Further factors will then be considered, such as the possibility of exploring the space of search to provide more than one alternative which can be considered optimal, or the mathematical modelling of the flight trajectory with uncertainty.

4.1 Generation phase

The consultation with the Advisory Board has established that the problem faced by Pilot3 is a multi-objective optimisation with only two objectives: **Cost** and **On-Time Performance (OTP)**. Cost is a complex objective built from the aggregation of three key performance indicators (KPIs): cost of fuel, cost of IROPS, and other costs. OTP is considered a binary variable indicating if on-time performance is reached (i.e., arrival delay≤15 minutes) or not. The binary nature of the OTP objective implies that it can be considered as a requirement (to be checked) or as a constraint (to be maintained if possible).

It has also been indicated that the optimisation should focus on minimising cost, and on producing the trade-off (in cost) required to achieve the OTP, if possible. This has led to the selection of the **Lexicographic ordering** multi-criteria optimisation method to capture this trade-off. Trajectories will be generated considering cost as first objective and OTP as second. If OTP is not achieved, then the trajectory generator will be re-executed forcing OTP (if possible), by adding it as a constraint, and then minimising the cost as a second objective.

The generation phase will not select a solution but generate different trajectories in the Pareto front so that the decision maker can explore the alternatives after their ranking.

4.2 Ranking phase

The Ranking phase will perform a discrete optimisation to filter and rank the solutions produced by the Generation phase. As indicated, cost is a complex objective formed by three different KPIs. Different preferences between these costs could be pre-defined (during the configuration of Pilot3) by the airlines. Indicating a two-by-two ranking of relative importance of KPIs has been deemed possible by the Advisory Board.

The multi-criteria selection method most suitable for this process is the **Compromise Ranking Method**, also known as the **VIKOR** method, improved by introducing the **Analytical Hierarchy Process** for assigning the weights of relative importance of attributes.

The ranking will be performed independently among the trajectories which do not meet the OTP, and the trajectories which achieve OTP (if any).

4.3 Selection phase

The Selection phase will rely on the interaction with the pilot via the **Human-Machine Interface**. The pilot will receive information on the alternatives and will be able to explore and compare the ranked trajectories. Some tactical operational indicators will be computed to facilitate this process (e.g., number of flight level changes required). The crew will be able to dismiss trajectories, add constraints, and re-execute the optimisation to re-evaluate the alternatives. Note that the ranking of alternatives could include the previously generated (and not yet dismissed) trajectories.

The specific requirements and design of the interface will be performed as part of WP4 activities.

5 Next steps and look ahead

This deliverable has defined the multi-criteria optimisation methods that will be considered for implementation in Pilot3. These findings will be translated into detailed requirements and implemented as part of the activities of WP4 - Model development. The first prototype will be delivered by March 2021 (D4.1 Crew Assistant Decision model description (first release) and D4.2 - Crew Assistant Decision model software package (first release)). Note that the multi-criteria optimisation methods selected present the framework of optimisation in terms of how the different objectives will be used for the generation of optimised trajectories and their ranking and filtering. However, the details on the optimisation technique that will be used will be defined in WP4. The technique implemented will have to consider other requirements such as the need of providing a solution within a restricted time, and the possibility of exploring the space of search to provide more than one alternative which can be considered optimal (if they exist).

Detailed information on the interaction with the Advisory Board and data gathering activities (first Advisory Board meeting and follow-up consultation), which were used to produce D1.1 - Technical Resources and Problem definition (Pilot3 Consortium, 2020) and provided input for the selection process of the multi-criteria mechanism described in this deliverable, will be described in D3.1 - Airlines data collection report (due July 2020).

Additional feedback from the Advisory Board (and external experts and stakeholders) will be gathered as part of the First release Pilot3 workshop planned in April 2021. With the feedback gathered the final version of Pilot3 will be implemented. As part of the verification and validation activities (WP5), scenarios will be defined and further interaction with the Advisory Board be sought. The verification and validation plan will be reported in D5.1 - Verification and validation plan (due July 2020).

6 References

Adali, E. A. and Isik, A. T. 2017. The multi-objective decision making methods based on MULTIMOORA and MOOSRA for the laptop selection problem Journal of Industrial Engineering International. Vol. 13(2) pp. 229-237.

Airbus 1998 (May). Getting to grips with the cost index. Tech. report Airbus Flight Operations Support & Line Assistance, Blagnac, France.

Alexanderian, A., Petra, N., Stadler, G., Ghattas, O., 2017. Mean-Variance Risk-Averse Optimal Control of Systems Governed by PDEs with Random Parameter Fields Using Quadratic Approximations. SIAM/ASA Journal on Uncertainty Quantification, 5(1), 1166–1192.

Athan, T. W., Papalambros, P. Y., 1996, A note on weighted criteria methods for compromise solutions in multi-objective optimization, Engineering Optimzation, Vol. 27, pp. 155–176.

Brans, J. P., Vincke, P. and Mareschal, B., 1986, How to select and how to rank projects: The PROMETHEE method, Eur. J. Oper. Res., vol. 24, pp. 228–238.

Diwekar, U. M., Kalagnanam, J. R., 1997. Efficient sampling technique for optimization under uncertainty. AIChE Journal, 43, 440–447.

Figueira, J., Mousseau, V. and Roy, B., 2005, ELECTRE methods, in: Multiple Criteria Decision Analysis: State of the Art Surveys, Springer, Berlin, 2005, pp. 133–153.

Gershon M. and Durckstein L., 1983, An algorithm for Choosing of a Multiobjective Technique, Essays in Economics and Mathematical Systems 209, Springer-Verlag, Berlin, Heidelberg, pp. 53-62.

Goerigk, M., Schöbel, A., 2016. Algorithm Engineering in Robust Optimization. Lecture Notes in Computer Science, 245–279.

González-Arribas, D., Soler, M. and Sanjurjo-Rivo, M., 2018 (Mar), Robust aircraft trajectory planning under wind uncertainty using optimal control, Journal of Guidance, Control, and Dynamics, 41:3, pp. 673-688.

Hajiagha, S. H. R., Mahdiraji, H. A., Zavadskas E. K. and Hashemi, S. S., 2014, Fuzzy Multi-Objective Linear Programming Based on Compromise VIKOR Method. International Journal of Information Technology & Decision Making. Vol. 13(4) pp. 679–698

Hobbs, B.F., 1986, What Can We Learn From Experiments in Multiobjective Decision Analysis?, IEEE Transactions on Systems, Man, and Cybernetics 16:3, pp. 384-394.

Houska, B., Diehl, M., 2009. Robust nonlinear optimal control of dynamic systems with affine uncertainties. In Proceedings of the 48th IEEE Conference on Decision and Control (CDC09) (pp. 2274–2279). Shanghai (China).

Houska, B., Logist, F., Van Impe, J., Diehl, M., 2012. Robust optimization of nonlinear dynamic systems with application to a jacketed tubular reactor. Journal of Process Control, 22, 1152–1160

Kok, 1986 (Jul), The interface with decision makers and some experimental results in interactive multiple objective programming methods, European Journal of Operational Research, 26:1, pp. 96-107.

Li, P., Arellano-Garcia, H., Wozny, G., 2008. Chance constrained programming approach to process optimization under uncertainty. Computers and Chemical Engineering, 32, 25 – 45.

Li, P., Wendt, M., Wozny, G., 2002. A probabilistically constrained model predictive controller. Automatica, 38, 1171 – 1176.

Linsmeier, T.J., Pearson, N.D., 1996. Risk Measurement: An Introduction to Value at Risk. Office for Futures and Options Research Working Paper No. 96-04, University of Illinois, 1996.

Mäkelä, M. M., 1993, Issues of implementing a Fortran subroutine package NSOLIB fro nonsmooth optimisation, Report 5/1993, University of Jyväskylä, Deparment of Mathematics, Jyväskylä (Findland).

Markowitz, H., 1959. Portfolio Selection: Efficient Diversification of Investments. New York: Wiley.

Marler, R. T, Arora, J. S., 2004, Survey of multi-objective optimization methods for engineering, Structural and Multidisciplinary Optimization, Vol 26, pp. 369–395.

Marler, R. T, Arora, J. S., 2005, Function-transformation methods for multi-objective optimization, Engineering Optimization, Vol. 37(6), pp. 551–570

Marler, R. T, Arora, J. S., 2010, The weighted sum method for multi-objective optimization: new insight, Vol 41, pp. 853–862.

Miettinen, K., 1999, Nonlinear Multiobjective Optimization. Boston - London -Dordrecht: Kluwer Academic Publishers.

Mitra, K., 2009. Multiobjective optimization of an industrial grinding operation under uncertainty. Chemical Engineering Science, 64, 5043–5056.

Nagy, Z. K., & Braatz, R. D. (2003). Robust nonlinear model predictive controlof batch processes.AIChE Journal,49, 1776 –1786.

Nagy, Z., Braatz, R., 2004. Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis. Journal of Process Control, 14, 411–422.

Nagy, Z., Braatz, R., 2007. Distributional uncertainty analysis using power series and polynomial chaos expansions. Journal of Process Control, 17, 229– 240.

EUROPEAN UNION

Opricovic, S. and Tzeng, G. H., 2004, The compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS, European Journal of Operational Research Vol 156 pp. 445–455.

Opricovic, S. and Tzeng, G. H., 2007, Extended VIKOR method in comparison with outranking methods, European Journal of Operational Research, Vol 178 pp. 514–529.

Opricovic, S., 2011, Fuzzy VIKOR with an application to water resources planning, Expert Systems with Applications. Vol. 38, pp. 12983-12990.

Pareto, V., 1896, Cours d'Economie politique, Journal: Revue Economique. Droz, Geneva.

Pilot3 Consortium, 2020 (May), Technical Resources and Problem definition, Tech. report, Deliverable D1.1., Ed. 01.02.

Rao, J.R. and Roy, N., 1989, Fuzzy set theoretic approach of assigning weights to objectives in multicriteria decision making. International Journal of Systems Science, Vol 20, pp. 1381–1386.

Recker, S., Kuhl, P., Diehl, M., Bock, H., 2012. Sigmapoint approach for robust optimization of nonlinear dynamic systems. In Proceeding of SIMUL-TECH 2012, 199–207.

Rockafellar, R. T., Uryasev, S., 2000. Optimization of conditional value-at-risk. Journal of Risk, 2, 21–41.

Roy, B. 1968, Classement et choix en présence de points de vue multiples La méthode Electre. Rev. Fr. Inf. Rech. Op. Vol 8 pp. 57–75.

Saaty, T. L. 1994, How to Make a Decision: The Analytic Hierarchy Process. INFORMS Journal on Applied Analytics. Vol 24(6).

Sabaei, D., Erkoyuncu, J. and Roy, R., 2015, A review of multi-criteria decision making methods for enhanced maintenance delivery. Procedia CIRP 37. p30-35.

Sakawa, M. and Yano, H., 1985, Interactive multiobjective reliability design of a standby system by the fuzzy sequential proxy optimisation technique (FSPOT), International Journal of Systems Science. Vol 16(2), pp. 177-195.

San Cristóbal, J.R., 2011, Multi-criteria decision-making in the selection of a renewable energy project in Spain: The VIKOR method, Renewable Energy 36, pp. 498-502.

Sayadi, M. K., Heydari, M., Shahanaghi, K., 2009, Extension of VIKOR method for decision making problem with interval numbers, Applied Mathematical Modelling Vol 33, pp. 2257–226.

Schenkendorf, R., Kremling, A., Mangold, M., 2009. Optimal experimental design with the sigma point method. Systems Biology, IET, 3, 10–23.

Stewart T.J., 1992, A Critical Survey on the Status of Multiple Criteria Decision Making Theory and Practice, OMEHA 20:5-6, pp. 569-586.

Vallerio, M., Hufkens, J., Van Impe, J., Logist, F., 2015. An interactive decision-support system for multi-objective optimization of nonlinear dynamic processes with uncertainty. Expert Systems with Applications, 42(21), 7710–7731.

Yanga, W-E., Wanga, J-Q., Wanga, X-F., 2012, An outranking method for multi-criteria decision making with duplex linguistic information. Fuzzy Sets and Systems. Vol 198, pp. 20–33

7 Acronyms

AB: Advisory Board

AHP: Analytic Hierarchy Process

ATC: Air Traffic Control

ATM: Air Traffic Management

CI: Cost Index

CSJU: Clean Sky 2 Joint Undertaking

CVaR: Conditional Value at Risk

DM: Decision Maker

DX.Y: Deliverable number (X=workpackage, Y=deliverable numbering within workpackage)

ELECTRE: ELimination Et Choix Traduisant la REalité

GDF: Geoffrion-Dyer-Feinberg Method

H2020: Horizon 2020 research programme

HMI: Human machine interface

IROPs: Irregular Operations costs

ISWT: Interactive Surrogate Worth Trade-off method

KPI: Key Performance Indicator

OM: Operations Manual

OTP: On-Time Performance

PI: Performance Indicator

SPOPT: Sequential Proxy Optimisation Technique

STOM: Satisficing Trade-Off Method

TMA: Terminal Manoeuvring Area

VaR: Value at Risk

VIKOR: VIsekriterijumska Optimizacija I Kompromisno Resenje (Multicriteria Optimization and Compromise Solution)

Appendix A Comparative tables of multi-objective optimisation methods

This Annex presents a detailed literature review of multi-objective optimisation methods classified by families considering the domain characteristics of Pilot3. The tables have been colour-coded to indicate their suitability for the general characteristics of Pilot3 problem:

- Red: some criteria make this method not compatible for Pilot3.
- Yellow: some criteria would raise difficulties when applying this method to our problem.
- Green: desired criteria are all met when using these methods for our problem.

In the pros/cons column, text (following the same colour coding as the cells, with black for additional information) is included to emphasise advantages and drawbacks of the methods, in light of Pilot3 needs and characteristics.

Note also that some methods might not neatly fit in one category but present characteristics of more than one. Finally, and since no method is found to be perfect for all criteria, a combination of several methods can be considered.

A.1 *A posteriori* methods

These methods are appealing as they do not require any input from the decision maker (DM) to select candidate Pareto efficient solutions. Instead a large number of these solutions (ideally the whole Pareto front) is generated first and presented to the DM. These methods could be considered a systematic approach, but they then have a high computational cost, which makes them usually prohibitive for real time application

Table 2. Review *a posteriori* methods

Method	Short description	Pros/cons
Weighting method	Associates each objective function with a weighting coefficient and minimise the weighted sum of the objectives → single objective function Problem solved repetitively by changing the weight values	• Pareto optimal as long as non-zero weights are used • Easy to implement • Inability to capture points on non-convex portions of the Pareto front (Possibility of convexifying the non-convex Pareto set by raising the objective functions to a high enough power under certain assumptions) • Computationally expensive
Epsilon-constraint (or bounded objective function) method	One of the objective functions is selected to be optimised and all the other ones are converted into	• Solutions can be found in non-convex areas of the Pareto front • Computationally expensive for certain applications (more expensive than

Method	Short description	Pros/cons
	constraints by setting an upper bound to each of them. Problem solved repetitively by changing the value of these bounds.	weighting methods since number of constraints increases)
Hybrid method (combining weighting method and Epsilon-constraint method)		• Any Pareto optimal solution can be found independently of the convexity of the problem • Computationally expensive (see Epsilon-constraint method)

A.2 *No preference* methods

These methods are only useful if the decision maker does not have any special expectations of the solution and she is satisfied simply with *some* optimal solution

Table 3. Review *no preference* methods

Method	Short description	Pros/cons
Compromise programming (or global criterion)	Distance between some reference point and the feasible objective region is minimised. Several alternatives exist to define either this reference point or the metric for measuring the distances. Another particular case of this method is the Tchebycheff solution (also known as Egalitarian solution or min-max optimisation), where the maximum distance to the ideal objective vector is chosen as decision performance index, in such a way that the system is no better-off than its worse-off individual.	• Simple method to use • Obtain a solution where no special hopes are set • Pareto optimal
Multi-objective Proximal Bundle Method	From a given starting point in the Pareto front moves in a direction where the values of all the objective functions improve simultaneously (Mäkelä, 1993).	• Weakly Pareto optimal

EUROPEAN UNION

A.3 *A priori* methods

The main difficulty of these methods consists in understanding and correctly interpreting the conceptual significance of the preferences, which is not always obvious. Also, the decision maker might not necessarily know beforehand what is possible to attain in the problem and how realistic her expectations are. Because they are a key point in the selection of the method, the required (type of) inputs are specifically identified for each method.

Table 4. Review *a priori* methods

Method	Short description	Pros/cons
Value/Utility function	Weighting method can be considered as special case of value function, where utilities are linear and additive Input: DM gives an accurate and explicit mathematical form of the value function that represents her preferences globally.	• Pareto optimal as long as non-zero weights are used • A complete ordering in the objective space is set and a single objective optimisation problem is solved. • Difficulty to encode mathematically the real preferences of the decision maker.
Lexicographic ordering	DM arranges the objective functions according to their absolute importance. Then the most important objective function is minimised (or maximised). If the problem has an unique solution, it is the solution of the whole multi-objective optimisation problem. Otherwise, the 2nd most important objective function is minimised, but adding a new constraint in the problem to guarantee that the most important objective function preserves its optimal value found in the previous step. If this new problem has an unique solution, it becomes the solution of the whole multi-objective optimisation problem, otherwise the process goes on as described above with the third, fourth, etc. objectives. Input: absolute order of importance of objectives (but the most important may remain the only one optimised)	• Always provides Pareto optimal • A maximum of N optimisations are done (N=number of objectives) • Robust method • Simplicity • It is very unlikely that the process can optimise lower prioritised objectives, since an unique solution is likely to be found in the first steps of the process. • The method does not allow a small increment of an important objective function to be traded off with a great decrease of a less important objective function, which might be often appealing to the DM, i.e, order is definite and rigid.

Method	Short description	Pros/cons
Hierarchical approach	Modification of lexicographic ordering where the upper bounds obtained when minimising more important objective functions are relaxed by so-called worsening factors. These relaxations allow to trade off higher prioritised objectives in front of lower prioritised ones, exploring in this case, a widest area of the Pareto front containing solutions that can be more interesting to the DM. Input: order of importance of objective + relaxing/trade-off factor, i.e: most important goal is goal M but if I can improve goal N by only worsening M by less than X%, it is ok.	• Always provides Pareto optimal • A maximum of N optimisations are done (N=number of objectives) • Reduces the sensitivity of the final solution to the initial objective-function ranking process • Setting the relaxation factors might not be an obvious task for the decision maker.
Goal programming	The DM specifies (optimistic) aspiration levels for some of the objective functions (or all of them) forming goals, which are added in form of constraints in the optimisation problem. Then, any deviations from these aspiration levels are minimised, using a weighted function Input: goal/target value for each objective function	• Goal-setting can be an understandable and easy way of making decision, but this is highly dependent on the application • Weights do not have so direct an effect on the solution obtained (as in the a-priori weighting method) but are still relative to each other • No direct physical meaning of weights • There is no guarantee that it provides Pareto optimal solutions. • For large problems, computational burden could be an issue. • Not appropriate to use if it is desired to obtain trade-offs
Physical programming	Maps general classifications of goals and objectives, and verbally expressed preferences, to a utility function. DM quantitatively classifies	• Pareto optimal • Able to effectively optimise objective functions with

EUROPEAN UNION

Method	Short description	Pros/cons
	different ranges of values for each metric. It provides a means of incorporating preferences without having to conjure relative weights. Input: design metrics (KPI) functions of design parameters (variables), then DM specifies numerical ranges corresponding to different degrees of preference (desirable, tolerable, undesirable, etc.). These ranges include limits on the values of the metrics, which are modelled as additional constraints.	significantly different orders of magnitude • Allows one to make effective use of available information • Superior to the weighted sum method and to compromise programming in its ability to represent the complete Pareto optimal set with an even distribution of points • Requires significant familiarity with each objective and constraint • Initial coding can be relatively complicated
Weighting method	Associate each objective function with a weighting coefficient and minimise the weighted sum of the objectives → single objective function DM specifies beforehand a weighting vector representing his preferences, but some consider that instead of relative importance weighting coefficients should represent the rate at which DM is willing to trade off values of the objective functions Input: relative weight for each objective	• Pareto optimal as long as non-zero weights are used • Objective function should be normalised, otherwise role of the weighting coefficients may be greatly misleading • Weighting coefficients not easy to interpret and understand for average DM
Weighted Tchebycheff	Utopian objective vector is established. Distance from utopy point to feasible objective region, measured by a weighted Tchebycheff metric is minimised. Input: relative weight for each objective	• Pareto optimal as long as non-zero weights are used • Weighting coefficients not easy to interpret and understand for average DM

EUROPEAN UNION

Method	Short description	Pros/cons
Epsilon-constraint (or bounded objective function)	One of the objective functions is selected to be optimised and all the other ones are converted into constraints by setting an upper bound to each of them. Input: one objective function to be minimised (single most important) and upper bounds for the other ones	• Pareto optimal under certain assumptions • Non trivial choice of objective function + upper bounds to obtain a desirable solution → application dependent
HIBRID between weighted and bounded	The primary objective function is a weighted sum of all the objective functions and is subject to the constraints of the ε-constraint method. Input: weights + bounds of criteria	• Yields a Pareto optimal solution for any ε.
HIBRID between weighted and lexicographic (combined approach)	Several objective functions may belong to the same class of importance in the lexicographic order. In each priority class, a weighted sum of the deviational variables is minimised Input: ranking of objective + weights	
HIBRID between lexicographic and goal programming	Lexicographic approach of goal programming: DM must specify a lexicographic order for the goals in addition to the aspiration levels. Goal at the highest priority level is supposed to be infinitely more important than goal at the 2nd level, etc Input: ranking of objective + goal values	• Order is definite and rigid
VIKOR	Ranking and selecting from a set of alternatives, and determines compromise solutions for a problem with conflicting criteria, which can help DM to reach a final decision Combination of compromise programming (see no-preference methods below) and a weighting method. VIKOR ranks alternatives and determines the solution named	• Effective tool in multicriteria decision making, particularly in situations where DM is not able, or does not know to express her preference at the beginning of system design • Initial weights of relative importance of the attributes not

Method	Short description	Pros/cons
	compromise that is the closest to the ideal from an initial set of (given) weights. Can be started without interactive participation of DM, but DM is in charge of approving final solution and her preference must be included Input: give weights of criteria (expressing the DM's preference as the relative importance of the criteria) + best and worst value of each criteria	easy to interpret and understand for average DM • Weights may be assigned using AHP (see interactive methods) • Suitable for discrete decision problem with non-commensurable (different units) and conflicting criteria • Suitable when compromising is acceptable for conflict resolution • DM should be willing to approve solution closest to the ideal

A.4 Interactive methods

If the decision maker has enough time and capabilities to interact with the system, many of the weak points of the previous classes could be overcome. Namely, only part of the Pareto optimal point has to be generated and the decision maker does not have to know any global preference structure, since they are specified as the solution process evolves. At each iteration, some information is given to the decision maker and (s)he is asked then to answer some questions or provide some other type of information. After a reasonable number of iterations, the process stops. These methods differ by the type and amount of information that is given to (and provided by) the decision maker and how the overall problem is transformed into a single objective optimisation problem.

Stopping criteria are:

- DM gets tired of the process;
- some algorithm stopping (convergence) rule is fulfilled; or
- DM finds a desirable solution and wants to stop.

Gardiner and Vanderpoorten (1997) have studied that median number of iterations has been between 3 and 8.

Table 5. Review *iterative* methods

Method	Short description	Pros/cons
VIKOR	DM can give new values of parameters during algorithm **Input**: updated values of weights after being presented by possible solutions	• See a-priori methods

Method	Short description	Pros/cons
Interactive Surrogate Worth Trade-off method (ISWT)	Based onϵ-constraint method. Aims to maximise an underlying (implicitly known) value function. The opinions of the DM concerning the trade-off rates at the current solution point determine a search direction **Input**: at the beginning of the process: objective function to minimise + upper bounds of others. During process: select the most satisfactory solution for the continuation (using provided trade-off rates)	• All alternatives are Pareto optimal • Convergence rate of ISWT greatly depends on the accuracy and consistency of the answers of the DM • Lots of different assumptions to be satisfied to guarantee that the algorithm works (correctness of trade-off rate for instance)
Geoffrion-Dyer-Feinberg Method (GDF)	Same idea as ISWT, at each iteration, a local approximation of an underlying value function is generated and maximised. In GDF, marginal rates of substitution specified by DM are used to approximate the direction of steepest ascent of the value function, which is then maximised by a gradient-based method. **Input**: at the beginning of the process: reference objective function to minimise + upper bounds of others. During process: specify marginal rates of substitution between the reference function and the other objectives at the current solution	• Difficulties of DM in determining rates of substitution • Final solution not necessarily Pareto optimal (neither the different alternatives to choose from) • Consistent and correct marginal rates of substitution needed at each iteration • Value function must be continuously differentiable, strongly decreasing, etc (hard to check for practical applications)
Sequential Proxy Optimisation Technique (SPOT)	Includes some properties of ISWT + GDF	• Pareto optimal • Same drawbacks of previous methods
Tchebycheff method	Utopian objective vector is established. Distance from utopia point to feasible objective region, measured by a weighted Tchebycheff metric is minimised. Different solutions are obtained with different weighting vectors Solution space reduced by working with sequences of smaller and smaller subsets of weighting vector space leading to smaller subsets of Pareto optimal set	• Pareto optimal if using lexicographic weighted Tchebycheff • Role of DM quite easy to understand: only has to compare several alternative objective vectors and select the most preferred one • Ease of comparison depends on number of objective functions

EUROPEAN UNION

Method	Short description	Pros/cons
	At each iteration, different alternative objective vectors are presented to DM who is asked to select the most preferred one **Input**: initial weights, at each iteration choice of preferred solution for DM	• Personal capabilities of DM play an important role • Reduced flexibility of method since discarded parts of the weighting vector cannot be restored if DM changes her mind → some consistency is required • Computational cost for large problems
Step method	Similar to Tchebycheff but aims to find satisfactory solutions instead of optimising **Input**: initial weights (not critical) and then at a certain Pareto optimal objective vector, DM indicates both: functions that reached acceptable values and those whose values are too high (unacceptable). DM then allows the values of some acceptable objective to increase so that unacceptable can have lower values	• Information handled easy to understand (no complicated concepts introduced to DM) • Not necessarily Pareto optimal • It may be difficult to estimate appropriate amounts of increment that would allow desired amount of improvement in objective to be decreased → indirect control of the solution • Nadir vector (upper bounds of Pareto set) not easy to determine
Reference Point Method	Based on a reference point of aspiration level (feasible of infeasible point that is reasonable of desirable for DM) **Input**: initial reference point (aspiration level for each obj). Once alternatives (obtained with several close ref points) are presented to DM, choose if one is satisfactory, else DM specifies new ref point	• Reference points should be easy and intuitive for DM to specify and their consistency is not essential • Perturbating the reference point allows DM to get better conception of possible solutions • More direct and more explicit way than using for example weighting coef • Does not need consistency from DM (though convergence not necessarily fast then)
GUESS method	Similar to ref point method but only one optimal solution presented to DM at each iteration Trial and error method	• Reference points should be easy and intuitive for DM to specify and their consistency is not essential

Method	Short description	Pros/cons
	Input: initial reference point + bounds (optional). If solution presented is not satisfactory DM specifies new ref point	• More direct and more explicit way than using for example weighting coef • Does not need consistency from DM (though convergence not necessarily fast then) • Weakly pareto optimal • Nadir vector (upper bounds of Pareto set) not easy to determine
Satisficing Trade-Off Method (STOM)	Pareto optimal solution (obtained by optimising a scalarising function) is presented to DM **Input**: initial reference point + DM classifies possible solutions into: 1. unacceptable → then specify aspiration levels, 2. acceptable and can be relaxed, 3. acceptable and values must be kept as they are	• Pareto optimal if scalarising function well chosen • Same comments as Ref point and GUESS method, and in practice, classifying the objective functions into 3 classes and specifying amounts of increment and decrement for their values is a subset of specifying new reference point • Does not need consistency from DM
Light beam search	Projects a focused beam of light from the reference point onto the Pareto optimal set **Input**: DM can specify best and worst values for each objective function, as well as preference and veto thresholds. Present one solution + Pareto neighbours of it to DM. DM can compare a set of alternatives and affect this set in different ways	• Pareto optimal • Specifying threshold may be demanding for DM, but they can be altered at any time • May be computationally expensive to find Pareto neighbours
Outranking methods	Well-established method with a large history of successful real-word applications. The method compares all couples of alternatives and determine which are preferred by systematically comparing the alternatives for each criterion, trying to establish outranking relations between alternatives according on the basis of for how many components the decision maker judges indifference,	• Ability to take ordinal scales into account without converting the original scales into abstract ones with an arbitrary imposed range and at the same time maintain the original verbal meaning thresholds can be considered when modelling imperfect knowledge, permitting the utilisation of incomplete value information, such as judgements

Method	Short description	Pros/cons
	weak preference, preference or no-preference. These decisions can be complemented, for instance, with veto thresholds, which prevents a good performance in some components of the objective vector from compensating for poor values on some other components. Popular examples of outranking methods are ELECTRE (ELimination Et Choix Traduisant la REalité) (Roy, 1968; Figueira et al., 2005) and PROMOTHEE (Brans et al, 1986) families **ELECTRE**: 2 main parts: 1st, the construction of one or several outranking relations, which aims at comparing in a comprehensive way each pair of actions; 2nd, an exploitation procedure that elaborates on the recommendations obtained in the first phase. The nature of the recommendation depends on the problem being addressed: choosing, ranking or sorting **Input**: quantify the relative importance of criteria (importance/weight coefficients) + use thresholds of indifference and preference (veto thresholds: express the power attributed to a given criterion to be against the assertion "a outranks b", when the difference of the performances between g(b) and g(a) is greater than this threshold.) **PROMOTHEE**: The preference structure of PROMETHEE is based on pairwise comparisons. In this case the deviation between the evaluations of two alternatives on a particular criterion is considered. For small deviations, the decision-maker will allocate a small preference to the best alternative and even possibly no preference if he considers that this deviation is negligible.	on ordinal measurement scales and partial prioritisation (Yanga et al. 2012) • Usually the Electre Methods are used to discard some alternatives to the problem, which are unacceptable. After that we can use another MCDA to select the best one. The Advantage of using the Electre Methods before is that we can apply another MCDA with a restricted set of alternatives saving much time. • Number of interactions with the DM (i.e., number of queries to the DM asking for each pair) grows quadratically with the number of optimisation objectives.

Method	Short description	Pros/cons
	The larger the deviation, the larger the preference **Input**: information between the criteria (weights of relative importance of the different criteria) + information within each criterion (pairwise comparison)	
Analytic hierarchy process (AHP)	Primarily based on the pair wise comparison of matrices that DM uses to establish preferences between alternatives for different criteria and the rating methods. AHP generates a weight for each evaluation criterion according to DM pairwise comparisons of the criteria. The higher the weight, the more important the corresponding criterion. Next, for a fixed criterion, the AHP assigns a score to each option according to DM pairwise comparisons of the options based on that criterion. The higher the score, the better the performance of the option with respect to the considered criterion. Finally, the AHP combines the criteria weights and the options scores, thus determining a global score for each option, and a consequent ranking. The global score for a given option is a weighted sum of the scores it obtained with respect to all the criteria This method includes both the rating and comparison methods. **Input**: DM pairwise comparisons of criteria + for a fixed criterion DM pairwise comparisons of the options based on that criterion.	• Very flexible and powerful tool because scores, and therefore final ranking, are obtained on the basis of the pairwise relative evaluations of both the criteria and the options provided by user. • Computations made by the AHP are always guided by the DM experience, and the AHP can thus be considered as a tool that is able to translate the evaluations (both qualitative and quantitative) made by DM into a multicriteria ranking. • Simple because there is no need of building a complex expert system with DM's knowledge embedded in it • No weight • Can be used to assign weights of relative importance of different attributes needed in other methods • May require a large number of evaluations by the user, especially for problems with many criteria and options. Although every single evaluation is very simple, since it only requires DM to express how two options or criteria compare to each other, the load of the evaluation task may become unreasonable. In fact the number of pairwise comparisons grows quadratically with the number of criteria and options.

Method	Short description	Pros/cons
		• For instance, when comparing 10 alternatives on 4 criteria, 4·3/2=6 comparisons are requested to build the weight vector, and 4·(10·9/2)=180 pairwise comparisons are needed to build the score matrix That is for p alternatives and n criteria, n(n-1)/2 comparisons to build weight vector and n.p(p-1)/2 pairwise comparisons to build score matrix • To reduce DM's workload the AHP can be completely or partially automated by specifying suitable thresholds for automatically deciding some pairwise comparisons

A.5 References

Brans, J. P., Vincke, P. and Mareschal, B., 1986, How to select and how to rank projects: The PROMETHEE method, Eur. J. Oper. Res., vol. 24, pp. 228–238.

Figueira, J., Mousseau, V. and Roy, B., 2005, ELECTRE methods, in: Multiple Criteria Decision Analysis: State of the Art Surveys, Springer, Berlin, 2005, pp. 133–153.

Gardiner, L. R., and Vanderpooten, D. Interactive Multiple Criteria Procedures: Some Reflections, Multicriteria Analysis, pages 290-301, 1997. Springer Verlag, Berlin.

Mäkelä, M. M., 1993, Issues of implementing a Fortran subroutine package NSOLIB fro nonsmooth optimisation, Report 5/1993, University of Jyväskylä, Deparment of Mathematics, Jyväskylä (Findland).

Roy, B. 1968, Classement et choix en présence de points de vue multiples La méthode Electre. Rev. Fr. Inf. Rech. Op. Vol 8 pp. 57–75.

Yanga, W-E., Wanga, J-Q., Wanga, X-F., 2012, An outranking method for multi-criteria decision making with duplex linguistic information. Fuzzy Sets and Systems. Vol 198, pp. 20–33

Appendix B Review on robustness in optimisation

This Annex presents the review of different considerations of robustness in optimisation under uncertainty:

1. Strict robustness,
2. Cardinality constrained robustness,
3. Adjustable robustness,
4. Light robustness,
5. Recoverable robustness,
6. Regret robustness, and
7. Some further robustness concepts

B.1 Strict robustness

The most common concepts used in robust optimisation belong to the family of min-max robustness concepts (e.g., Bokrantz and Fredriksson, 2017; Ehrgott et al., 2014; Eichfelder et al. 2017; Kuroiwa and Lee, 2012). The approach is also known as classic robust optimisation, one-stage robustness, absolute deviation or simply robust optimisation. For minmax robustness, the objective functions are optimised in the worst case over all scenarios. The solutions computed are said to be min-max robust efficient.

Denoting the set of strictly robust solutions with respect to the uncertainty set $\mathcal{U}$ by

$$SR(\mathcal{U}) = \bigcap_{\xi \epsilon \mathcal{U}} \mathcal{F}(\xi)$$

the strictly robust counterpart of the uncertain optimisation problem is given as:

$$\text{(SR)} \quad min \sup_{\xi \epsilon \mathcal{U}} f(x, \xi)$$

$$\text{s.t.}\, x \,\epsilon\, SR(\mathcal{U})$$

$$x \epsilon \mathcal{X}.$$

This approach is particularly applicable in the case when all scenarios that may occur should be taken into consideration (when constructing a bridge that must be stable, no matter which traffic scenario occurs, or for constructing airplanes or nuclear power plants, for instance). However, high degree of conservatism of strict robustness is not applicable to all situations which call for robust solutions. For example, the application of strict robustness in designing the timetable in public transportation would mean that all announced arrival and departure times have to be met, no matter what happens. This could be achieved by introducing high buffer times, which will lead to a practically inapplicable timetable.

In order to overcome the conservatism pertaining in the mentioned approach, the following methods have been introduced with the aim to relax the strict robustness.

B.2 Cardinality Constrained Robustness

This concept has been proposed by Bertsimas and Sim (2004) for linear programming problems for the purpose of overcoming the conservatism of strict robustness by shrinking the uncertainty set $\mathcal{U}$. The author considered that it is unlikely that all coefficients of one constraint change simultaneously to their worst-case values. Instead, they restricted the number of coefficients which are allowed to be changed.

Considering a constraint of the form $a_1x_1+\ldots+a_nx_n \leq b$ with an uncertainty $\mathcal{U} = \{a\epsilon\mathbb{R}^n: a_i \in [\hat{a}_i - d_i, \hat{a}_i + d_i], i = 1, \ldots, n\}$, their robustness concept requires a solution x to satisfy:

$$\sum_{i=1}^{n} \hat{a}_i\, x_i + \max_{\substack{S\subseteq\{1,\ldots,n\},\\ |S|=\Gamma}} \left\{ \sum_{i\in S} d_i |x_i| \right\} \leq b$$

for a given parameter $\Gamma \in \{0, \ldots, n\}$. In this way, solution x to this model hedges against all scenarios in which at most Γ many uncertain coefficients may deviate from their nominal values at the same.

B.3 Adjustable Robustness

Similar to two-stage programming in stochastic optimisation, Ben-Tal et al. (2003) introduced the new approach in robust optimisation called adjustable robustness. The authors assume that variables can be decomposed into two sets. The values for the ***here-and-now variables*** have to be found by the robust optimisation algorithm in advance, while the decision about the ***wait and-see variables*** can wait until the actual scenario $\xi \,\epsilon\, \mathcal{U}$ becomes known. The approach assumes that the variables $x = (u, v)$ are split into $u\epsilon\mathcal{X}^1 \subseteq \mathbb{R}^{n_1}$ and $v \,\epsilon\, \mathcal{X}^2 \subseteq \mathbb{R}^{n_2}$ with $n_1 + n_2 = n$, where the variables u need to be determined before the scenario $\xi \,\epsilon\, \mathcal{U}$ becomes known, while the variables v may be determined after ξ has been realised. Thus, we may also write $x(\xi)$ to emphasise the dependence of v on the scenarios. The uncertain optimisation problem $(P(\xi), \xi \,\epsilon\, \mathcal{U})$ is rewritten as:

$$P(\xi)\ \min f(u, v, \xi)$$

$$F(u, v, \xi) \leq 0$$

$$(u, v) \in \mathcal{X}^1 \times \mathcal{X}^2.$$

When fixing the here-and-now variables, one has to make sure that for any possible scenario $\xi \,\epsilon\, \mathcal{U}$ there exists $v \,\epsilon\, \mathcal{X}^2$ such that (u, v) is feasible for ξ. The set of adjustable robust solutions is therefore given by:

$$\mathrm{aSR} = \{u\epsilon\, \mathcal{X}^1 : \forall \xi \in \mathcal{U}\ \exists v\, \epsilon \mathcal{X}^2\ s.t. (u, v) \in \mathcal{F}(\xi)\} = \bigcap_{\xi\epsilon\mathcal{U}} Pr_{\mathcal{X}^1}(\mathcal{F}(\xi))$$

There are several variations of the concept of adjustable robustness. Instead of two stages, multiple stages are possible. For example, Bertsimas and Caramanis (2010) proposed the approach of finitely adaptable solutions in which instead of computing a new solution for each scenario, a set of possible static solutions is computed, such that at least one of them is feasible in each stage.

B.4 Light Robustness

The approach is applied by Fischetti and Monaci (2009) who further develop the concept of cardinality constrained robustness. The idea of light robustness is that a solution must not be too bad in the nominal case and thus, a certain nominal quality is fixed. Among all solutions satisfying this standard, the concept asks for the most "reliable" one with respect to constraint violation. The general lightly robust counterpart is of the following form:

$$(LR) \quad \min \sum_{i=1}^{m} \omega_i \gamma_i$$

$$\text{s.t.}\, f\left(x, \hat{\xi}\right) \leq f * \left(\hat{\xi}\right) + \rho$$

$$F(x, \xi) \leq \gamma \;\; \forall \xi \in \mathcal{U}$$

$$x \in \mathcal{X}, \gamma \in \mathbb{R}^m$$

where ω_i models a penalty weight for the violation of constraint i and ρ determines the required nominal quality.

B.5 Recoverable Robustness

The recoverable robustness is a two-stage concept very similar to adjustable robustness. The concept has been developed in Cicerone et al. (2007), Conde and Candia (2007), Liebchen et al. (2009) and Stiller (2008). Its basic idea is to allow a class of *recovery algorithms* $\mathcal{A}$ that can be used in case of a disturbance. A solution x is called ***recovery robust*** with respect to $\mathcal{A}$ if for any possible scenario $\xi \in \mathcal{U}$ there exists an algorithm $A \in \mathcal{A}$ such that A applied to the solution x and the scenario ξ constructs a solution $A(x, \xi) \in \mathcal{F}(\xi)$, i.e., a solution which is feasible for the current scenario. The recovery robust counterpart according to the formulation given in Conde and Candia (2007) is the following:

$$(RR) \quad \min_{(x,A) \epsilon \mathcal{F}(\hat{\xi}) \times \mathcal{A}} f(x)$$

$$\text{s.t.}\, A(x, \xi) \epsilon \mathcal{F}(\xi) \; \forall \xi \epsilon \mathcal{U}$$

It can be extended by including the recovery costs of a solution x: Let $d(A(x, \xi))$ be a possible vector-valued function that measures the costs of the recovery, and let $\lambda \epsilon \Lambda$ be a limit on the recovery costs, i.e., $\lambda \geq d(A(x, \xi))$ for all $\xi \in \mathcal{U}$. Assume that there is some cost function $g{:}\, \Lambda \rightarrow \mathbb{R}$ associated with λ.

Settings

$$A(x, \xi, \lambda) \epsilon \mathcal{F}'(\xi) \Leftrightarrow d(A(x, \xi,) \leq \lambda \; \wedge A(x, \xi) \epsilon \mathcal{F}(\xi)$$

gives the recovery robust counterpart with limited recovery costs:

$$(\mathrm{RR-LIM}) \quad \min_{(x,A,\lambda) \epsilon \mathcal{F}(\hat{\xi}) \times \mathcal{A} \times \Lambda} f(x) + g(\lambda)$$

$$\text{s.t.}\, A(x, \xi) \epsilon \mathcal{F}(\xi) \; \forall \xi \epsilon \mathcal{U}$$

The computational tractability of this robustness concept heavily depends on the problem, the recovery algorithms and the uncertainty under consideration.

B.6 Regret Robustness

The concept of regret robustness differs from the other robustness concepts insofar it usually only considers uncertainty in the objective function. Instead of minimising the worst-case performance of a solution, it minimizes the difference to the objective function of the best solution that would have been possible in a scenario. In some publications, it is also called deviation robustness. Let $f^*(\xi)$ denote the best objective value in scenario $\xi \in \mathcal{U}$. The min-max regret counterpart of an uncertain optimisation problem with uncertainty in the objective is then given by:

EUROPEAN UNION

$$\text{(Regret)} \quad min \sup_{\xi \epsilon \mathcal{U}} (f(x,\xi) - f^*(\hat{\xi}))$$

$$\text{s.t.} \, F(x) \leq 0$$

$$x \in \mathcal{X}.$$

The regret robustness concept finds its broad application in location theory and in scheduling problems.

B.7 Some Further Robustness Concepts

- *Reliability.* Another approach to robust optimisation is to relax the constraints of strict robustness. This leads to the concept of reliability of Ben-Tal and Nemirovski (2000), in which the constraints $F(x,\xi) \leq 0$ are replaced by $F(x,\xi) \leq \gamma$ for some $\gamma \in \mathbb{R}^m_{\geq 0}$. A solution x which satisfies $F(x,\xi) \leq \gamma$ for all $\xi \in \mathcal{U}$:

$$F(x,\xi) \leq \gamma \; for \; all \; \xi \in \mathcal{U}$$

is called reliable with respect to γ. The goal is to find a reliable solution which minimises the original objective function in the worst case.

- *Soft Robustness.* The basic idea of soft robustness as introduced in Ben-Tal et al. (2010) is to handle the conservatism of the strict robust approach by considering a nested family of uncertainty sets, and allowing more deviation in the constraints for larger uncertainties. Specifically, instead of an uncertainty set $\mathcal{U} \subseteq \mathbb{R}^m$, a family of uncertainties

$\{\mathcal{U}(\varepsilon) \subseteq \mathcal{U}\}_{\varepsilon>0}$ with $\mathcal{U}(\varepsilon_1) \subseteq \mathcal{U}(\varepsilon_2)$ for all $\varepsilon_2 \geq \varepsilon_1$ is used. The set of soft robust solutions is then given as:

$$\text{softR} = \{x \in \mathcal{X} \colon F(x,\xi) \leq \varepsilon \; \forall \xi \in \mathcal{U}(\varepsilon), \; \varepsilon > 0\}$$

Note that strict robustness is a special case with $\mathcal{U}(\varepsilon) = \mathcal{U}$ for all $\varepsilon > 0$.

- *Comprehensive Robustness*. While the adjustable robust approach relaxes the assumption that all decisions have to be made before the realised scenario becomes known, the approach of comprehensively robust counterparts Ben-Tal et al. (2006) also removes the assumption that only scenarios defined in the uncertainty set $\mathcal{U}$ need to be considered. Instead, using a distance measure $dist$ in the space of scenarios, and a distance measure $\overline{dist}$ in the solution space, they assume that the further away a scenario is from the uncertainty set, the further away the corresponding solution is allowed to be from the set of feasible solutions. As in adjustable robustness, the dependence of x on the scenario ξ is allowed, and we may write $x(\xi)$. The adjustable robust counterpart is extended to the following problem:

$$\text{CRC} \quad \min z$$

$$s.t. \, f(x(\xi),\xi) \leq z + \alpha_0 dist(\xi,\mathcal{U}) \; \forall \xi$$

$$\overline{dist}(x(\xi),\mathcal{F}(\xi)) \leq \alpha dist(\xi,\mathcal{U}) \; \forall \xi$$

where α, α_0 denote sensitivity parameters.

- *Uncertainty Feature Optimisation*. Instead of assuming that an explicit uncertainty set is given, which may be hard to model for real-world problems, the uncertainty feature optimisation (UFO) approach (Eggenberg et al. 2011) rather assumes that the robustness of a solution is given by an explicit function. For an uncertain optimisation problem $P(\xi)$, let $\mu: \mathbb{R}^n \rightarrow \mathbb{R}^p$ be a measure for p robustness features. The UFO-counterpart of the uncertain problem is then given by:

$$(\text{UFO}) \ \ \text{vecmax} \mu(x)$$
$$\text{s.t.} \, F(x) \leq 0$$
$$f(x) \leq (1+\rho) f^*(\hat{\xi})$$
$$x \in \mathcal{X}$$

where $f^*(\hat{\xi})$ denotes the best objective value to the nominal problem.

B.8 Application of robustness concept in different engineering problems

Table 6, taken from (Source: Goerigk and Schöbel, 2016), lists some relevant applications where robust optimisation is applied comparing at least two algorithms to the same problem.

Table 6. Papers presenting experiments comparing at least two algorithms for the same robustness concept (Source: Goerigk and Schöbel, 2016).

Year	Paper	Problem	Concept	Algorithms
2005	Montemanni and Gambardella	Spanning tree	Regret	Branch and bound, MIP
2006	Montemanni	Spanning tree	Regret	Bender's decomp., MIP, branch and bound
2008	Nikulin	Spanning tree	Regret	Simulated annealing, branch and bound, Bender's decomp
2008	Taniguchi, Yamada and Kataoka	Knapsack	Strict	Branch and bound with and without preprocessing
2008	Velarde and Marti	Capacitated sourcing	Adjustable	Tabu search
2009	Conde	Critical path	Regret	MIP and heuristic
2010	de Farias, Zhao and Zhao	Machine scheduling	Strict	MIP with and without cuts
2010	Bohle, Maturana and Vera	Wine harvesting	cc^* robust	MIP and scenario generation
2010	Ng, Sun and Fowler	Lot allocation	Strict	Branch-and-price and heuristics

Year	Paper	Problem	Concept	Algorithms
2011	Catanzaro, Labbe and Salazar-Neumann	Shortest path	Regret	IP with and without preprocessing
2011	Pereira and Averbakh	Assignment	Regret	MIP, Bender's decomp., genetic algorithms
2012	Kasperski, Makuchowski and Zielinski	Spanning tree	Regret	Tabu search and IP
2012	Fischetti and Monaci	Diverse	cc* robust	MIP and cutting planes
2012	Song, Lewis, Thompson and Wu	Knapsack	Strict	Local search and branch and bound
2013	Monaci, Pferschy and Serafini	Knapsack	cc* robust	Dynamic programming and IP
2013	Ouorou	Capacity assignment	Adjustable	Approximations

*"cc" abbreviates "cardinality constrained"

B.9 References

Ben-Tal, A., Nemirovski, A., 2000. Robust solutions of linear programming problems contaminated with uncertain data. Mathematical Programming, 88, 411–424.

Ben-Tal, A., Goryashko, A., Guslitzer, E., Nemirovski, A., 2003. Adjustable robust solutions of uncertain linear programs. Mathematical Programming, 99, 351–376.

Ben-Tal, A., Boyd, S., Nemirovski, A., 2006. Extending scope of robust optimization: comprehensive robust counterparts of uncertain problems. Mathematical Programming, 107 (1–2), 63–89 (2006)

Ben-Tal, A., Bertsimas, D., Brown, D.B., 2010. A soft robust model for optimization under ambiguity. Operation Research, 58(4), 1220–1234.

Bertsimas, D., Sim, M., 2004. The price of robustness. Operation Research, 52(1), 35–53.

Bertsimas, D., Caramanis, C., 2010. Finite adaptability in multistage linear optimization. IEEE Transaction on Automatic Control, 55 (12), 2751–2766.

Bohle, C., Maturana, S., Vera, J., 2010. A robust optimization approach to wine grape harvesting scheduling. European Journal of Operational Research, 200 (1), 245–252.

Bokrantz, R., Fredriksson, A., 2017. Necessary and sufficient conditions for Pareto efficiency in robust multiobjective optimization. European Journal of Operational Research, 262 (2), 682–692.

Catanzaro, D., Labbe, M., Salazar-Neumann, M., 2011. Reduction approaches for robust shortest path problems. Computers & Operations Research, 38 (11), 1610–1619.

Cicerone, S., D'Angelo, G., Stefano, G.D., Frigioni, D., Navarra, A., 2007. Robust algorithms and price of robustness in shunting problems. In: Proceedings of the 7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization and Systems (ATMOS 2007).

EUROPEAN UNION

Conde, E., Candia, A., 2007. Minimax regret spanning arborescences under uncertain costs. European Journal of Operational Research, 182(2), 561–577.

Conde, E., 2009. A minmax regret approach to the critical path method with task interval times. European Journal of Operational Research, 197 (1), 235–242.

de Farias, J.R., Zhao, H., Zhao, M., 2010. A family of inequalities valid for the robust single machine scheduling polyhedron. Computers & Operations Research, 37 (9), 1610–1614.

Eggenberg, N., Salani, M., Bierlaire, M., 2011. Uncertainty feature optimization: an implicit paradigm for problems with noisy data. Networks, 57 (3), 270–284.

Ehrgott, M., Ide, J., Schöbel, A. 2014. Minmax robustness for multi-objective optimization problems. European Journal of Operational Research, 239 (1), 17–31.

Eichfelder, G., Krüger, C., Schöbel, A., 2017. Decision uncertainty in multiobjective optimization. Journal of Global Optimization, 69(2), 485–510.

Fischetti, M., Monaci, M., 2009. Light robustness. In: Ahuja, R.K., Mohring, R.H., Zaroliagis, C.D. (eds.) Robust and Online Large-Scale Optimization. LNCS, 5868, pp. 61–84. Springer, Heidelberg.

Fischetti, M., Monaci, M., 2012. Cutting plane versus compact formulations for uncertain (integer) linear programs. Mathematical Programming Computation, 4 (3), 239–273.

Goerigk, M., Schöbel, A., 2016. Algorithm Engineering in Robust Optimization. Lecture Notes in Computer Science, 245–279.

Kasperski, A., Makuchowski, M., Zielinski, P., 2012. A tabu search algorithm for the minmax regret minimum spanning tree problem with interval data. Journal of Heuristics, 18 (4), 593–625.

Kuroiwa, D., Lee, M. G., 2012. On robust multiobjective optimization. Vietnam Journal of Mathematics, 40 (2,3), 305 – 317.

Liebchen, C., Lubbecke, M., Mohring, R., Stiller, S., 2009. The concept of recoverable robustness, linear programming recovery, and railway applications. In: Ahuja, R.K., Mohring, R.H., Zaroliagis, C.D. (eds.) Robust and Online Large-Scale Optimization. LNCS, vol. 5868, pp. 1–27. Springer, Heidelberg (2009).

Monaci, M., Pferschy, U., Serafini, P., 2013. Exact solution of the robust knapsack problem. Computers & Operations Research, 40 (11), 2625–2631.

Montemanni, R., Gambardella, L., 2005. A branch and bound algorithm for the robust spanning tree problem with interval data. European Journal of Operational Research, 161 (3), 771–779.

Montemanni, R., 2006. A Benders decomposition approach for the robust spanning tree problem with interval data. European Journal of Operational Research, 174(3), 1479–1490.

Ng, T.S., Sun, Y., Fowler, J., 2010. Semiconductor lot allocation using robust optimization. European Journal of Operational Research, 205(3), 557–570.

Nikulin, Y., 2008. Simulated annealing algorithm for the robust spanning tree problem. Journal of Heuristics, 14 (4), 391–402.

Ouorou, A., 2013. Tractable approximations to a robust capacity assignment model in telecommunications under demand uncertainty. Computers & Operations Research, 40 (1), 318–327.

Pereira, J., Averbakh, I., 2011. Exact and heuristic algorithms for the interval data robust assignment problem. Computers & Operations Research, 38 (8), 1153–1163.

Song, X., Lewis, R., Thompson, J., Wu, Y., 2012. An incomplete m-exchange algorithm for solving the large-scale multi-scenario knapsack problem. Computers & Operations Research, 39 (9), 1988–2000.

Stiller, S., 2008. Extending concepts of reliability. Network creation games, real-time scheduling, and robust optimization. Ph.D. thesis, TU Berlin.

Taniguchi, F., Yamada, T., Kataoka, S., 2008. Heuristic and exact algorithms for the max–min optimization of the multi-scenario knapsack problem. Computers & Operations Research, 35 (6), 2034–2048.

Velarde, J.L.G., Marti, R., 2008. Adaptive memory programing for the robust capacitated international sourcing problem. Computers & Operations Research, 35(3), 797–806.

-END OF DOCUMENT-