Optimal and Heuristic Approaches for Constrained Flight Planning under Weather Uncertainty

Florian Geißer *
florian.geisser@anu.edu.au

Guillaume Povéda †
guillaume.poveda@airbus.com

Felipe Trevizan *
felipe.trevizan@anu.edu.au

Manon Bondouy †
manon.bondouy@airbus.com

Florent Teichteil-Königsbuch †
florent.teichteil-koenigsbuch@airbus.com

Sylvie Thiébaux *
sylvie.thiebaux@anu.edu.au

Abstract

Aircraft flight planning is impacted by weather uncertainties. Existing approaches to flight planning are either deterministic and load additional fuel to account for uncertainty, or probabilistic but have to plan in 4D space. If constraints are imposed on the flight plan these methods provide no formal guarantees that the constraints are actually satisfied. We investigate constrained flight planning under weather uncertainty on discrete airways graphs and model this problem as a Constrained Stochastic Shortest Path (C-SSP) problem. Transitions are generated on-the-fly by the underlying aircraft performance model. As this prevents us from using off-the-shelf C-SSP solvers, we generalise column-generation methods stemming from constrained deterministic path planning to the probabilistic case. This results in a novel method which is complete but computationally expensive. We therefore also discuss deterministic and heuristic approaches which average over weather uncertainty and handle constraints by scalarising a multi-objective cost function. We evaluate and compare these approaches on real flight routes subject to real weather forecast data and a realistic aircraft performance model.

1 Introduction

Aircraft trajectory optimisation is key for successful flight operation; it ensures aircraft safety and is valuable for airlines, as it helps to minimise fuel consumption and to reduce the environmental impact of the flight. The task is to compute a flight plan for a given aircraft mission which minimises some objective function — traditionally a combination of two conflicting components, fuel and time, weighted according to a ratio known as the *cost index* (AIRBUS Customer Services 1998). Weather phenomena play an important role when planning flights. Convective activity, i.e. manifestations of convection in the atmosphere, indicates showers and thunderstorms, and such areas have to be avoided. Weather is inherently uncertain, yet airline operations today are mainly based on deterministic weather forecasts (Cheung et al. 2014) and do not take uncertainty into account when optimising the flight trajectory. Finding the most cost-efficient flight plan is often not enough. In reality, a plan has to satisfy multiple constraints: for safety reasons, air traffic control rules may require a minimum amount of fuel reserve throughout the flight; airline operations may require the flight to arrive at a specified arrival time while concurrently avoiding convective areas.

In this paper, we view the problem of constrained flight planning under uncertainty as a *Constrained Stochastic Shortest Path* Problem (C-SSP) (Altman 1999; Trevizan et al. 2016). Like an ordinary SSP, a C-SSP consists in finding a plan with minimum expected cost from an initial state to a goal state. In addition, a C-SSP has a number of secondary cost functions (e.g. travel time, distance travelled through convection), and the plan must ensure that the expectation of each of these secondary costs remains within the given constraint.

We model constrained flight planning under uncertainty as a C-SSP where the uncertainty is obtained from Probabilistic Ensemble Weather Forecasts (Palmer 2019), and transitions and their costs are computed on the fly based on a realistic aircraft performance model (Nuic, Poles, and Mouillet 2010). This allows us to estimate fuel burn and flight time in a much more realistic fashion than is typically done with a model-based approach. However, a serious limitation of having a black box simulator to compute transitions, costs and probabilities, is that we cannot use state-of-the-art C-SSP solvers (Trevizan, Thiébaux, and Haslum 2017) which require a factored description of the problem to be available in advance.

To solve the resulting C-SSP, we therefore resort to *Column Generation*, which allows us to iteratively solve the Linear Program (LP) associated with the C-SSP. Each iteration solves an increasingly larger LP that typically remains much smaller than the original LP. Column generation is commonly used to solve *deterministic* constrained shortest path problems (Desrosiers and Lübbecke 2005), i.e., a special case of a C-SSP in which all actions are deterministic. We generalise that column generation approach to handle C-SSPs, which results in a sound and complete algorithm that allows us to compute optimal policies for constrained flight planning under weather uncertainty. As the policies re-

*The Australian National University
†Airbus - Artificial Intelligence Research

turned by column generation are stochastic, we also briefly discuss options to compute deterministic policies, which are currently viewed as more acceptable for safety-critical applications.

Optimally solving C-SSPs, especially in conjunction with an expensive performance model to compute transitions, can be computationally prohibitive. To assess the benefits of this approach and the computational trade-offs, we therefore investigate and compare with two simpler approaches. The first follows Franco, Rivas, and Valenzuela (2017) and computes a flight plan for a deterministic problem that averages over the members of the ensemble forecast. Since even this deterministic state space can be too large for efficient search, the second additionally decomposes the problem into a 2D search (ignoring altitude and speed), and a greedy choice of altitude and speed from a set of heuristic strategies. While this provides no guarantees, it allows reasoning over a larger state space and is therefore able to solve larger problems.

We evaluate these three approaches on real flight routes between different city pairs and consider constraints such as arrival time and distance spent in convective areas. Weather uncertainty is based on an ensemble prediction system for three different dates where weather was reported to be the most important factor for flight delays.

2 Related Work

Existing work on *probabilistic* flight planning dates back to at least the early 2000s. Nilim, El Ghaoui, and Duong (2002) model the uncertain flight planning problem as a Markov decision process with unknown but bounded transition probabilities and provide dynamic programming algorithms that minimise the expected delay or the expected probability to encounter storms. More recently, Franco, Rivas, and Valenzuela (2017) reason over wind uncertainty obtained from ensemble weather forecasts. They propose to either compute a flight plan minimising fuel consumption over a deterministic problem that represents the average of the ensemble forecasts, or to compute a flight plan for each forecast separately and apply a function that represents the probability distribution of the resulting cost on the ensemble. Neither of these approaches handles constraints.

Regarding constraints, Erzberger et al. (2016) describe algorithms that generate trajectories avoiding convection areas and compute scheduled arrival times, but they assume that convection areas are detected a priori by radar sensors. Taylor et al. (2018) apply a multi-objective genetic algorithm that treats constrained flight planning as a multi-objective problem and computes a Pareto set of solutions. However, they do not reason about weather uncertainty, and the genetic algorithm does not provide any guarantees. On the other hand, González-Arribas, Soler, and Sanjurjo-Rivo (2016) consider uniform distribution ensemble forecasts and provide an ad-hoc solution to compute a set of Pareto-optimal trajectories by sequential scalarisation of a multi-objective cost function. However, computing the Pareto-optimal trajectories quickly becomes infeasible once multiple constraints are involved. Knudsen, Chiarandini, and Larsen (2017) handle European airways constraints explicitly during A* search. These constraints capture e.g. locations whose visit prevents or mandates the visit of others, and are very different from the expected cost constraints we focus on.

3 Modelling the Problem

We model flight planning under weather uncertainty as a Constrained Stochastic Shortest Path problem (C-SSP), i.e. as an SSP that incorporates constraints on the expectation of various cost functions. In the following, we give the necessary background for SSPs and C-SSPs, and formally introduce the problem of constrained flight planning under weather uncertainty.

3.1 Stochastic Shortest Path Problems

A *stochastic shortest path problem* (SSP) (Bertsekas and Tsitsiklis 1996) is a tuple $\mathbb{S} = \langle \mathsf{S}, s_0, \mathsf{G}, \mathsf{A}, P, C \rangle$, where S is the finite set of states, $s_0 \in \mathsf{S}$ is the initial state, $\mathsf{G} \subset \mathsf{S}$ is the non-empty set of goal states, A is the finite set of actions, $\mathsf{A}(s)$ is the subset of actions applicable in state s, $P(s'|s,a)$ represents the probability that $s' \in \mathsf{S}$ is reached after applying action $a \in \mathsf{A}(s)$, and $C(s,a,s') \in \mathbb{R}_+^*$ is the immediate cost of applying action a to transition from state s to s'. A solution for $\mathbb{S}$ is a *deterministic* stationary policy $\pi : \mathsf{S} \to \mathsf{A}$ such that $\pi(s) \in \mathsf{A}(s)$ is the action to be applied in state s. A policy is *proper* if it reaches G from s_0 with probability 1. Assuming $\mathbb{S}$ has at least one proper policy and all improper policies have an infinite cost, there exists an *optimal* policy minimising the total expected cost of reaching G from s_0.

3.2 Constrained SSPs

A constrained SSP (C-SSP) is an SSP with $k+1$ cost functions, for $k \in \mathbb{N}$, in which one cost function is optimised while the remaining k costs are constrained by an upper bound.[1] Formally, a C-SSP is the tuple $\mathbb{C} = \langle \mathsf{S}, s_0, \mathsf{G}, \mathsf{A}, P, \vec{C}, \vec{u} \rangle$ where $\mathsf{S}, s_0, \mathsf{G}, \mathsf{A}$, and P are defined as for the SSP, $\vec{C} = [C_0, \ldots, C_k]$ is the cost function vector and $\vec{u} = [u_1, \ldots, u_k]$ is the cost upper bound vector. We refer to C_0 as *primary* cost and all other cost functions as *secondary* costs.

Differently from SSPs, a solution for a C-SSP is a *stochastic* stationary policy π (Altman 1999), i.e., π maps a state s to a probability distribution over $\mathsf{A}(s)$. We denote by $\pi(s,a)$ the probability of executing a in s. Let $E[C_i|\pi]$ be the total expected cost C_i when following a policy π from s_0 to the goal, then π is feasible (i.e., respects all cost constraints) if $E[C_i|\pi] \le u_i$, for all $i \in \{1, \ldots, k\}$. An optimal policy π for a C-SSP is any feasible and proper policy that minimises the total expected cost C_0 of reaching G from s_0, i.e., minimises $E[C_0|\pi]$.

In contrast to SSPs, there is no guarantee that the optimal policy for C-SSPs is deterministic. Moreover, whilst the complexity of finding the (potentially stochastic) optimal policy for C-SSPs is polynomial, finding the optimal deterministic policy is NP-complete (Dolgov and Durfee 2005). However, since stochastic flight planning policies are not yet

[1] Lower bounds $x \ge b$ can be modelled as $-x \le -b$.

used in the industry, we will also consider the generation of deterministic policies.

Notice that cost constraints limit the maximum *expected* value of the cost functions, i.e., $E[C_i|\pi]$. Although a limit on the maximum observed cost might be desired, they are often too pessimistic in a stochastic environment. For instance, under weather uncertainty, a transcontinental route requires more fuel than the plane's capacity in order to account for the extremely rare event of flying through heavy head wind for tens of thousands of kilometres.

3.3 Flight Planning under Weather Uncertainty

In the problem of flight planning under weather uncertainty, we are given an airway graph representing possible routes between waypoints, a source (the waypoint of a departure airport) and a target (the waypoint of an arrival airport). We want to find a policy from the source to the target that minimises expected fuel consumption and complies with constraints about the arrival time of the flight and the distance it spends in convection areas.

Formally, an *airway graph* is a tuple (N, E), where each node $(x, y) \in N \subset \mathbb{R}^2$ is a waypoint representing latitude and longitude, and an edge $(n_i, n_j) \in E \subseteq N \times N$ represents a path between n_i and n_j. To compute the fuel required to travel between waypoints we rely on an aircraft performance model. This model computes the fuel consumption between two waypoints n_i and n_j based on the following parameters: the current *mass* $m \in M \subseteq \mathbb{R}^*_+$ of the aircraft, its current *altitude* $h \in H \subset \mathbb{N}$ and *speed*[2] $v \in V \subset (0, 1]$, the current time $t \in T \subseteq \mathbb{N}$, and the current *weather* $w \in W$, which we will formally define below. Therefore, we can model the fuel consumption when travelling between n_i and n_j by the function $\mathsf{f} : E \times M \times H \times V \times T \times W \to \mathbb{R}^*_+$. The time t and distance d travelled between waypoints, as well as the distance c travelled through convective areas, are computed in a similar fashion using the performance model.[3]

We assume there exists a set of weathers W and the probability of observing $w \in W$ at time t is given by the *weather model* $P_w(w|n_i, n_j, t)$. In principle, altitude could be an input as well, but the weather data available to us does not consider this. We consider weather ensemble forecasts as the source of weather uncertainty, where each member of the weather ensemble W is a specific weather scenario and we assume a uniform probability distribution over the ensemble members. We want to emphasise that our approach is in principle independent of the underlying weather model with the only requirement that there exists a well-defined probability distribution on W. In our implementation, a particular weather consists of several attributes such as temperature and wind data, which is used by the performance model for its computations. But from the point of view of the C-SSP, the weather and performance model is just a black box.

We are now able to specify the state space, action space and transition function of the underlying SSP. Each state consists of a location, the current mass, altitude, and speed of the aircraft, and the current time, i.e. $\mathsf{S} = N \times M \times H \times V \times T$. The action space A is defined such that for each edge $e = (n_i, n_j) \in E$ we can fly from n_i to n_j at a specified speed level, targeting a specific altitude, i.e. $\mathsf{A} = \{a_{e,h,v} | e \in E, h \in H, v \in V\}$. The transition probability function is determined by the weather model and the aircraft performance model: given state $s = (n_i, m, h, v, t)$ and action $a_{e,h',v'}$ with $e = (n_i, n_j)$ we have $P(s' = (n_j, m', h', v', t')|s, a_{e,h',v'}) = \sum_{w \in W' \subseteq W} P_w(w|n_i, n_j, t)$ where W' is the set of weathers $w \in W$ such that $m - \mathsf{f}(e, m, h', v', t, w) = m'$ and $t + \mathsf{t}(e, m, h', v', t, w) = t'$. Informally, the set of successor states is computed by calling the performance model for each weather at time t, which yields the mass and time of a successor state.

The primary cost to optimise is determined by the difference in mass induced by the fuel consumption, i.e. $C(s, a_{e,h',v'}, s') = m' - m$, where m and m' are the mass of s and s', respectively. Finally, the secondary cost constraints we consider capture bounds on the expected travel time or on the expected percentage of the distance that was travelled through convective areas. The first constraint uses a secondary cost function $C_t(s, a_{e,h',v'}, s') = t' - t$ where t and t' are the time in states s and s' respectively, and time bounds $[l_t, u_t]$ representing the arrival time window. For the second constraint we denote the maximum percentage of the distance that is allowed to go through convective areas as α. We are interested in policies that satisfy $\mathsf{c}(\pi) \leq \alpha \cdot \mathsf{d}(\pi)$, where $\mathsf{c}(\pi)$ denotes the distance that policy π travels through convection, and $\mathsf{d}(\pi)$ the total travel distance. Since constraints are defined over transition cost functions we break this constraint down to the transition level, resulting in the secondary cost function $C_c(s, a_{e,h',v'}, s') = \mathsf{c}(e, t, h, h') - \alpha \mathsf{d}(e, h, h')$ where t is the time in state s, h and h' are the altitude in states s and s' respectively, and the upper bound for the constraint is 0.

Given an airway graph (N, E), departure node n_I, arrival node $n_\star$, the C-SSP corresponding to our problem is $\langle \mathsf{S}, s_0, \mathsf{G}, \mathsf{A}, P, [C, C_t, -C_t, C_c], [u_t, -l_t, 0] \rangle$, where S, A, P, C, C_t, C_c, u_t, and l_t are defined as above, $s_0 = (n_I, m^0, 0, v^0, t^0)$ where m^0, v^0 and t^0 are the takeoff mass, speed and time, and G is the set of states representing $n_\star$.

4 Solving the Flight Planning C-SSP

The size of the full state space of the C-SSPs underlying the flight planning problem makes it infeasible to solve them using any algorithm that requires enumerating the full state space a priori. Moreover, since the aircraft performance model simulates the physical properties (e.g., drag and lift) of the aircraft, as well as, its engine efficiency for the given altitude, computing a single state expansion is expensive. Thus, even if we could represent the complete state space, it would be computationally expensive to generate it.

Unfortunately, the state-of-the-art heuristic search planner for C-SSPs, i^2-dual (Trevizan, Thiébaux, and Haslum 2017), cannot be used with black box models since it requires a factored representation of the problem to compute its inte-

[2]More precisely: the ratio between the aircraft speed and the speed of sound for current atmospheric conditions, known as Mach.

[3]These functions do not necessarily need all the parameters required by f.

grated heuristic. To the best of our knowledge, the only other heuristic search planner for C-SSPs is i-dual (Trevizan et al. 2016). I-dual can be used with black box models; however it requires $k+1$ heuristic functions, one for each cost function C_i, used to prune infeasible solutions. For our flight planning application, this requirement is not feasible since finding non-trivial lower bounds for cost functions such as convection is still an open topic. Moreover, non-informative heuristics (e.g., $h = 0$) would provide no early pruning, and in the presence of conflicting objectives (violating a constraint can save fuel), i-dual with non-informative heuristics would result in expanding almost the complete infeasible region of the state space before finding a feasible solution.

For the reasons above, we develop a new algorithm for C-SSPs based on *column generation*. Column generation is an Operations Research technique that allows Linear Programs (LPs) with a large number of variables (columns) to be solved by considering only a subset of them. Due to space limitations, we cannot provide an in-depth overview of column generation and instead refer the reader to the work by Desrosiers and Lübbecke (2005).

4.1 Problem Formulation

Let Π_{det} be the set of all deterministic proper policies, feasible or not, for a given C-SSP $\mathbb{C} = \langle \mathsf{S}, s_0, \mathsf{G}, \mathsf{A}, P, \vec{C}, \vec{u} \rangle$. Our approach consists in finding the optimal convex combination of deterministic policies (i.e., a probability distribution on Π_{det}) while enforcing that the resulting stochastic policy is feasible. This problem is formalised in LP1 where the decision variables are the probabilities p_π of applying policy $\pi \in \Pi_{det}$ and $E[C_i|\pi]$ is a constant for LP1 that is computed beforehand. C1 and C2 enforce that $\vec{p}$ is a probability distribution over Π_{det} and C3 enforces the cost constraints. Given the optimal solution $\vec{p}^*$ of LP1, the optimal stochastic policy for $\mathbb{C}$ is $\pi^*(s,a) = \sum_{\pi \in \Pi_{det}} \pi(s,a) p^*_\pi$ where, by abuse of notation, $\pi(s,a) = 1$ if $\pi(s) = a$ and 0 otherwise. LP1 is a reformulation of the usual LP based on *occupation measures* (Altman 1999) used for computing optimal policies for C-SSPs and it is obtained by applying a generalisation of the *arc to path reformulation* from constrained deterministic shortest paths (Ahuja, Magnanti, and Orlin 1993, Sec. 3.5) where arcs are probabilistic actions and paths are policies.

$$\min_{\vec{p}} \quad \sum_{p_\pi} E[C_0|\pi] p_\pi \tag{LP1}$$

$$\text{s.t.} \quad p_\pi \geq 0 \qquad \forall \pi \in \Pi_{det} \tag{C1}$$

$$\sum_{\pi} p_\pi = 1 \tag{C2}$$

$$\sum_{p_\pi} E[C_i|\pi] p_\pi \leq u_i \qquad \forall i \in \{1, \cdots, k\} \tag{C3}$$

4.2 Iterative Improvement

Since Π_{det} can be large and expensive to compute, we cannot solve LP1 directly and instead apply column generation to solve a sequence of increasingly larger LPs. In column generation, LP1 is known as the Master Problem (MP) and the generated LPs are known as Reduced Master Problem (RMP). Let $\hat{\Pi} \subseteq \Pi_{det}$, then our RMP is LP1 with Π_{det} replaced by $\hat{\Pi}$. A key concept in column generation is the *reduced cost of a column*, which is a function that estimates the impact of adding this column to the RMP. For our application, columns are deterministic proper policies and the reduced cost of a policy $\pi \in \Pi_{det} \setminus \hat{\Pi}$ is

$$rc(\pi) = E[C_0|\pi] - \lambda_{conv} - \sum_{i \in \{1, \cdots, k\}} \lambda_{cost\text{-}i} E[C_i|\pi],$$

where λ_{conv} and $\lambda_{cost\text{-}i}$ are the optimal value for the dual variable associated with the constraints C2 and C3, respectively (Desrosiers and Lübbecke 2005, p.7).

Since LP1 is a minimisation problem, we want to find one or more policies with strictly negative reduced cost to add to the RMP as they can potentially improve the solution quality of the RMP. Thus, given a feasible RMP, column generation performs the following iterative improvement loop: (i) solve the RMP; (ii) find one or more negative reduced cost columns; (iii) add these columns to the RMP and; (iv) repeat. The main theorem of column generation provides the termination condition of this iterative procedure: the RMP is equivalent to the MP (i.e., they have the same solution) iff $rc(\pi) \geq 0$ for all $\pi \in \Pi_{det} \setminus \hat{\Pi}$. The key insight of our approach is that, finding a policy π with negative reduced cost or proving that one does not exist can be done by *solving a single unconstrained SSP* as we explain next.

Consider the problem of finding the policy π^*_{rc} with the most negative reduced cost rc^* for an RMP over $\hat{\Pi}$ with dual optimal solution $\vec{\lambda} = [\lambda_{conv}, \lambda_{cost\text{-}1}, \cdots, \lambda_{cost\text{-}k}]$. Formally, this problem is $rc^* = \min_{\pi \in \Pi_{det} \setminus \hat{\Pi}} E[C_0|\pi] - \lambda_{conv} - \sum_{i \in \{1, \cdots, k\}} \lambda_{cost\text{-}i} E[C_i|\pi]$. Let $C_{rc}(s,a) = C_0(s,a) - \sum_{i \in \{1, \cdots, k\}} \lambda_{cost\text{-}i} C_i(s,a)$, then, by linearity of expectations, we have that $rc^* = \min_{\pi \in \Pi_{det} \setminus \hat{\Pi}} E[C_{rc}|\pi] - \lambda_{conv}$. The search space of this minimisation problem can be extended to all deterministic policies (i.e., Π_{det}) without changing its solution since the reduced cost of all policies in the RMP are guaranteed to be non-negative (Bertsimas and Tsitsiklis 1997). Next, we exploit this property to compute rc^* by solving an SSP.

Let $\mathbb{S}_{rc} = \langle \mathsf{S}, s_0, \mathsf{G}, \mathsf{A}, P, C_{rc} \rangle$ be the SSP obtained from $\mathbb{C}$ by replacing C_0 by C_{rc} and removing all other cost functions. Suppose, w.l.o.g. that $\mathbb{S}_{rc}$ has a single (deterministic) optimal policy $\pi^*_{\mathbb{S}_{rc}}$. Since $E[C_{rc}|\pi] - \lambda_{conv}$ equals $rc(\pi)$ and $\pi^*_{\mathbb{S}_{rc}}$ is optimal, we have that $\pi^*_{\mathbb{S}_{rc}}$ equals π^*_{rc}. Thus, any optimal solver for SSPs (e.g., Nilsson; Hansen and Zilberstein (1968; 2001)) can be used for finding a policy with negative reduced cost or proving that one does not exist (i.e., when $E[C_{rc}|\pi^*_{\mathbb{S}_{rc}}] - \lambda_{conv} \geq 0$). However, the optimal solution for $\mathbb{S}_{rc}$ is only required to prove that the RMP and MP are equivalent, thus it is enough to find any policy for $\mathbb{S}_{rc}$ with negative reduced cost for the intermediate iterations. This allows us to use non-optimal SSP solvers for policy improvement and use an optimal solver only for proving optimally.

4.3 Initialisation

Our method so far is capable of iteratively adding columns to the RMP until it finds the optimal solution; however, it requires a feasible RMP to start with in order to obtain the dual optimal solution $\vec{\lambda}$. Thus, we need to provide a method to initially populate $\hat{\Pi}$. This is done by a similar approach as the reduced cost search using the SSP $\mathbb{S}_{rc}$ except that the *Farkas cost* (Lübbecke 2010, p.6) is used instead of the reduced cost. For our RMP, the Farkas cost of a policy is $fc(\pi) = -\sum_{i\in\{1,\cdots,k\}} \lambda^{\text{ray}}_{cost\text{-}i} E[C_i|\pi]$ where $\lambda^{\text{ray}}_{cost\text{-}i}$ is the *dual ray* of the infeasible RMP. The dual ray is trivially obtained by any LP solver as the "proof of infeasibility" of an LP and it represents a direction in which the dual of the RMP is unbounded. The main results related to Farkas cost are: (i) $fc(\pi) \geq 0$ for all $\pi \in \hat{\Pi}$; (ii) if $fc(\pi) \geq 0$ for all $\pi \in \Pi_{det} \setminus \hat{\Pi}$, then the MP (i.e., $\mathbb{C}$ for us) is infeasible; and (iii) adding a policy $\pi \in \Pi_{det} \setminus \hat{\Pi}$ s.t. $fc(\pi) < 0$ bounds the value of the RMP's dual in the $\vec{\lambda}^{\text{ray}}$ direction; therefore, either the RMP becomes feasible or a new dual ray can be found. Thus, similarly to C_{rc} and $\mathbb{S}_{rc}$, we can find new policies to be added to the RMP or prove that the original problem is infeasible by solving the SSP $\mathbb{S}_{fc} = \langle \mathsf{S}, s_0, \mathsf{G}, \mathsf{A}, P, C_{fc}\rangle$ where $C_{fc}(s,a) = -\sum_{i\in\{1,\cdots,k\}} \lambda^{\text{ray}}_{cost\text{-}i} C_i(s,a)$. This allows us to initialise the RMP with any policy or set of policies. Similarly to the policy improvement procedure, the optimal solution for $\mathbb{S}_{fc}$ is only required to prove the infeasibility of the MP and any policy with negative Farkas cost can be used for the initialisation procedure.

It is important to notice that both C_{rc} and C_{fc} can be negative depending on the values of $\vec{\lambda}$ and $\vec{\lambda}^{\text{ray}}$, respectively, and the secondary cost functions. This can lead to negative cost cycles and thus to potential algorithmic and theoretical issues regarding the solutions of $\mathbb{S}_{rc}$ and $\mathbb{S}_{fc}$; however, this is not an issue for our application because the state space S of our application is *acyclic*, as fuel monotonically decreases.

4.4 Optimal Deterministic Policy

As mentioned before, our approach computes an optimal *stochastic* policy for $\mathbb{C}$; however, this might not always be desired. For instance, current regulations do not allow stochastic policies and instead a deterministic policy needs to be provided. When a deterministic policy is required, we return the policy $\pi \in \hat{\Pi}$ that is feasible and has the minimum total expected cost. This procedure is quite efficient since, for all cost functions and all policies in $\hat{\Pi}$, their expected cost is already known. Since we do not search the complete set of deterministic policies Π_{det} in this procedure, the obtained solution is not guaranteed to be optimal; however, the optimality gap (i.e., the extra cost w.r.t. the optimal deterministic solution) of π is upper bounded by $E[C_0|\pi^*] - E[C_0|\pi]$ since $E[C_0|\pi^*] \leq E[C_0|\pi^*_{det}] \leq E[C_0|\pi]$, where π^* is the optimal stochastic policy computed by our approach. Also, notice that this approach to find a deterministic policy is not complete, i.e., it might not find a solution even if one exists. When all $\pi \in \hat{\Pi}$ are infeasible, then a Mixed Integer LP needs to be solved in order to find a solution or prove that none exist (Dolgov and Durfee 2005), increasing the complexity of the problem to NP-complete.

5 Deterministic Flight Planning

Column generation allows us to solve the C-SSP corresponding to our flight planning problem in a sound and complete way and provides us with an optimal stochastic policy. However, as mentioned above, we might not find a deterministic feasible policy, even if one exists. Additionally, computing reduced cost policies requires solving the underlying SSP multiple times. While the overhead of the performance model is mostly relevant for the initial iteration (for subsequent calls we can cache calls to the performance model), for larger problems even solving the initial SSP can become prohibitive. Therefore this section considers simpler deterministic approaches as a possible alternative.

5.1 Scalarising Constraints

Both deterministic approaches below handle constraints in a naïve fashion by incorporating them into the primary cost function. Given a cost function vector $\vec{C} = [C_0, \ldots, C_k]$, we define a vector of scalars $\vec{\Omega} = [\omega[1], \ldots, \omega[k]]$ and replace the primary cost function C_0 with the scalarised cost function $C_s = C_0 + \sum_{i=1}^{k} \omega[i] C_i$, i.e. a linear combination of secondary cost functions. We emphasise that this does not give any guarantees. In particular, an arbitrary $\vec{\Omega}$ does not necessarily correspond to an optimal solution of the C-SSP.

5.2 Average Determinisation

The first approach we consider, which we call *average determinisation*, follows Franco, Rivas, and Valenzuela (2017) by averaging uncertain weathers and therefore ending up with a deterministic planning problem. More formally, given state $s = (n_i, m, h, v, t)$ and action $a_{e,h'v'}$ with $e = (n_i, n_j)$, the resulting state $s' = res(s, a_{e,h',v'})$ is defined as follows: $s' = (n_j, m', h', v', t')$, where $m' = m - \frac{\sum_{w\in W} \mathsf{f}(s,w)}{|W|}$ and $t' = t + \frac{\sum_{w\in W} \mathsf{t}(s,w)}{|W|}$. The resulting deterministic problem, equipped with the scalarised cost function, can then be solved using classical search algorithms.

5.3 Heuristic Decomposition

While this deterministic transition system is considerably smaller than the probabilistic one, it can still be too large to allow reasoning over the full state and action space within the feasible planning time. We therefore consider a third approach, which we call *heuristic decomposition*. As is common in the industry, we decompose the above deterministic problem into two separate problems (Murrieta Mendoza 2013): 1) a horizontal (2D) planning phase that computes a least cost path between departure and arrival node based on the earth surface (i.e. without the use of a performance model), and 2) a vertical planning phase, where each node in the 2D plan is assigned a corresponding altitude and a speed level. We follow this idea and apply a heuristic approach for the vertical planning phase, which considers a pre-determined set of possible altitudes/speed strategies, and chooses greedily amongst them for each step in the 2D plan.

Formally, let $\pi^{\rightarrow}$ be a *horizontal plan*, i.e. a sequence of nodes $\langle n_I, \dots, n_\star \rangle$, where n_I and $n_\star$ are the departure and arrival node, respectively. We denote the i-th node in the sequence with $\pi_i^{\rightarrow}$. A *vertical strategy* $\pi^{\uparrow} : N \rightarrow H \times V$ assigns each node in $\pi^{\rightarrow}$ a corresponding altitude and speed. This induces a plan π_{4D} for the original (deterministic) flight planning problem which applies to the successive states $s_i = (\pi_i^{\rightarrow}, m, h, v, t)$ the action $a_{\pi_i^{\rightarrow}, h', v'}$ such that $(h', v') = \pi^{\uparrow}(\pi_i^{\rightarrow})$.

Computing $\pi^{\rightarrow}$ can be achieved by a standard best-first search on the airways graph with a transition function that is not based on a performance model and is therefore cheaper to compute. For vertical planning, we consider a pre-determined set of vertical strategies $\Pi^{\uparrow}$ and construct the final vertical strategy as a greedy combination of strategies in $\Pi^{\uparrow}$. This greedy combination is constructed iteratively, starting from s_0, by selecting, at each state s_i the strategy $\pi_i^{\uparrow} = \text{argmin}_{\pi^{\uparrow} \in \Pi} \sum_{j=i}^{|\pi^{\rightarrow}|} C_s(s_j, \pi^{\uparrow}, res(s_j, \pi^{\uparrow}))$. That is, at each step, we select the strategy which would minimise the scalarised cost if it was to be applied consistently at all future steps of the horizontal plan.

We construct the pre-determined set $\Pi^{\uparrow}$ of vertical strategies based on a lookup table of the aircraft performance model that maps a given state to a recommended altitude and speed. $\Pi^{\uparrow}$ consists of the following strategies:

1. A single strategy that just returns the recommendation of the performance model.
2. As before, but instead of choosing the speed recommended by the performance model we use varying speed levels, starting from 0.7 up to the maximum possible speed allowed for the aircraft in 0.01 steps. The number of resulting strategies depends on the aircraft and ranges from 12 up to 20 strategies.
3. Let h be a fixed altitude level. We consider again the recommendation of the performance model, but restrict the maximum altitude to h. For h, we consider an altitude range between 30000 feet to 42000 feet with a step size of 1000 feet, resulting in 13 strategies.
4. As in 3), but instead of using the speed recommended by the performance model we use the maximum possible speed, again: 13 strategies.
5. Finally, a single strategy that keeps the current altitude and speed levels.

Thus, $\Pi^{\uparrow}$ contains roughly 40–50 different vertical strategies.

6 Empirical Evaluation

We evaluate all approaches (column generation based on the C-SSP model, search on the average determinisation, and the heuristic decomposition) on a benchmark set of 9 city pairs (3 short, 3 medium, and 3 long distance flights); for each city pair we consider weather data of the 7th, 9th, and 11th of June 2018. On these dates, the weather was shown to be the most significant factor for the delay in public air transport (Eurocontrol 2018). Weather forecasts are provided by the European Center for Medium-Range Weather Forecasts (https://www.ecmwf.int/), and we use an ensemble size of 10 forecasts with uniform probability. We consider the BADA model (Nuic, Poles, and Mouillet 2010) as the underlying aircraft performance model to compute fuel burn, travel time and distance spent in convective areas. While we keep the airway graph size considerably small (around 20 waypoints for each problem; each waypoint connected to two to five neighbours with directed edges), the action space allows a speed between 0.7 and 0.9 Mach with 0.02 step size, and an altitude level between 30000 and 40000 feet, with a step size of 2000. This results in 50 combinations of altitude/speed levels and therefore between 100 and 250 applicable actions in each state. For the heuristic decomposition the horizontal planning phase considers a fixed altitude of 33000 feet and 0.8 Mach.

We note that when one wants to consider larger graphs, e.g. the complete European airspace, an additional precomputation step that prunes waypoints based on shortest lateral paths (i.e. 2D) is an option. However, even the smaller graphs we consider present a challenge: with potentially 250 applicable actions and each action resulting in 10 different outcomes (due to 10 different weathers), we end up with a branching factor of 2500 and nearly no duplicate states, since each weather can have a different (albeit small) effect on mass and time. We measured that 2500 transition calls to the performance model require on average 3 seconds. As a consequence, search on the C-SSP (as well as on the average determinisation) was not able to handle the complete state space in reasonable time. We therefore consider an abstraction of the state space instead of the full problem, and keep mass, speed[4] and time hidden from the state space. With this abstraction we still have to consider 2500 successors, but the number of non-duplicate successor states is much lower. In the experiments we use the abstraction of the problem to compute policies, but we evaluate all policies on the full problem, as described further below.

Recall that the weather model is a function dependent on waypoints and time. Not considering time in the state space therefore does not allow reasoning over uncertainty. Thus, for the C-SSP we additionally consider a weather flag in the state space that determines if the transition went through a convective area, leading to potentially two stochastic outcomes per transition.

Column generation requires a subsolver to iteratively solve the underlying unconstrained SSPs (cf. Section 4.2). In the following, we evaluate column generation with AO* (Nilsson 1968) as the underlying SSP search algorithm (denoted as C-SSP) against the average determinisation solved with A* (Hart, Nilsson, and Raphael 1968), as well as the heuristic decomposition. We refer to the plans of the latter as policies, to preserve a consistent terminology throughout this section. Both search algorithms (AO* and A*) perform

[4]We also ran experiments that considered speed in the state space; this resulted in a more precise fuel burn estimation, but the outcome of the overall evaluation did not change.

blind search. We compute policies on the abstraction of the problem, but evaluate all policies on the full problem: given a policy π, for each weather in the ensemble forecast we will simulate the trajectory induced by π starting from the initial state. We record the outcome of this trajectory for the relevant primary and secondary (constraint) cost functions. Since the probability distribution of the weather ensemble is uniform it is sufficient to do this once for each weather and average the outcomes. Clearly, evaluating a policy based on an abstraction of the model on the full problem may introduce some inaccuracies. We will discuss this when it was a relevant factor for an experiment.

Before we move on, we briefly discuss the impact of handling uncertainty through averaging. For this, we considered all problems without any constraints and we compared the fuel burn of policies computed by AO* on the SSP model and A* on the average determinisation. The results show that there is no significant difference between the optimal value of both policies for our benchmark set. This mirrors the observation of Franco, Rivas, and Valenzuela (2017) and is partially attributed to the underlying probabilities of the ensemble forecasts: as we assume a uniform probability distribution on the ensemble, reasoning over averages is a valid strategy.

In the following, we consider two sets of experiments: 1) We target a specific arrival time window, resulting in two constraints on the allowed arrival time. For this experiment we only consider C-SSP and heuristic decomposition, as the policies generated by A* were never able to satisfy any of the constraints. 2) We constrain the allowed distance travelled through convective areas. Convective cells are one of the main root cause of delays in aircraft operations, and are highly subject to forecast uncertainty.

In the case where all approaches satisfy the given constraint we compare the fuel consumption (i.e. cost) of the associated policies. Additionally, we evaluate the potential loss of committing to a deterministic policy. While we do not compare run-time in these experiments, in general one set of all 27 flight problems took less than 3 hours of total run-time for a single algorithm. For the C-SSP approach, the first iteration of column generation, i.e. the initial call to AO* to compute a policy for the original SSP, had the most impact on run-time. For subsequent iterations many calls of the performance model were already cached, so AO* solves the reduced cost SSP quickly.[5]

We implemented all algorithms in the scikit-decide library (AIRBUS - Artificial Intelligence Research 2020). While we can not release the implementation of the flight planning domain that uses the performance model, the column generation code is available in scikit-decide.

6.1 Time Window Constraints

For our first experiment we restrict the allowed arrival time for each problem, as described in Section 3. The chosen cost scalarisation, cf. Section 5.1, penalises constraint violation with a factor of 500 per missed second. It is possible that the constraints we impose are too strict and there does not even exist a solution based on a stochastic policy. Therefore, we first determined which problems are feasible in the first place. We emphasise that this is one of the advantages of the C-SSP based approach: if there is no (stochastic) policy that is able to satisfy the constraints the problem is proven to be unsolvable. For the time constraints, all of the problems we consider were determined to be feasible.

For each problem, we computed a (deterministic) policy, evaluated this policy on all 10 weather scenarios, and compared the number of problems where the constraint was violated. The constraint penalty imposed on the heuristic decomposition is quite large, and as a result this approach always arrived within the given bound. This was not always true for the deterministic policies returned by C-SSP. Recall that column generation computes a *stochastic* policy, but we evaluate an extracted *deterministic* policy. The cost of the stochastic policy is a lower bound on the optimal cost we can get for a problem, but this does not imply that the deterministic policy returned by the solver is always able to satisfy the constraint. Intuitively, this makes sense: the stochastic policy can consist of two deterministic policies; one policy violates the constraint in one direction (i.e. is always below the bound), and the other always violates the constraint in the other direction (always above the bound). Clearly, a combination of both policies is able to satisfy the constraint in expectation. Nevertheless, comparing the C-SSP policies against the policies provided by heuristic decomposition shows that while the heuristic decomposition never violates the constraint, the policy cost, i.e. fuel consumption, is much higher than for the policies provided by the C-SSP. Figure 1 depicts the average fuel burn and the constraint violation for short (left), medium (mid) and long (right) distance flights. For most of the problems where constraints are not satisfied the time window was missed by less than 20 seconds. At the same time, deterministic policies provided by the C-SSP are more conservative on fuel burn, in some instances with a difference of up to 5 tons of fuel. Table 1 additionally shows the total flight time in minutes of the C-SSP policies; for each city-pair, we only show the problems with the worst-case constraint violation over the three different weather dates we considered.

Where the deterministic policy is feasible we also evaluate the optimality gap between the stochastic and the deterministic policy, i.e. the difference between the optimal cost of a stochastic policy and the cost of the extracted deterministic policy. For 9 of the 27 problems, usually long distance flights, a stochastic policy would have allowed saving more than 2 tons of fuel. On the other hand, for 9 other problems the gap is less than 20 KG of fuel. For the remaining problems where the deterministic policy is feasible the gap varies between 200 and 300 KG of fuel.

6.2 Convection Constraints

In the second set of experiments we evaluate all three approaches and introduce a constraint on the expected distance that a flight is allowed to travel through convective areas. To determine distance in convection we use the convection in-

[5]A representative example for short distance flights is 120 seconds runtime for the initial call to AO* and 5 seconds for subsequent calls.

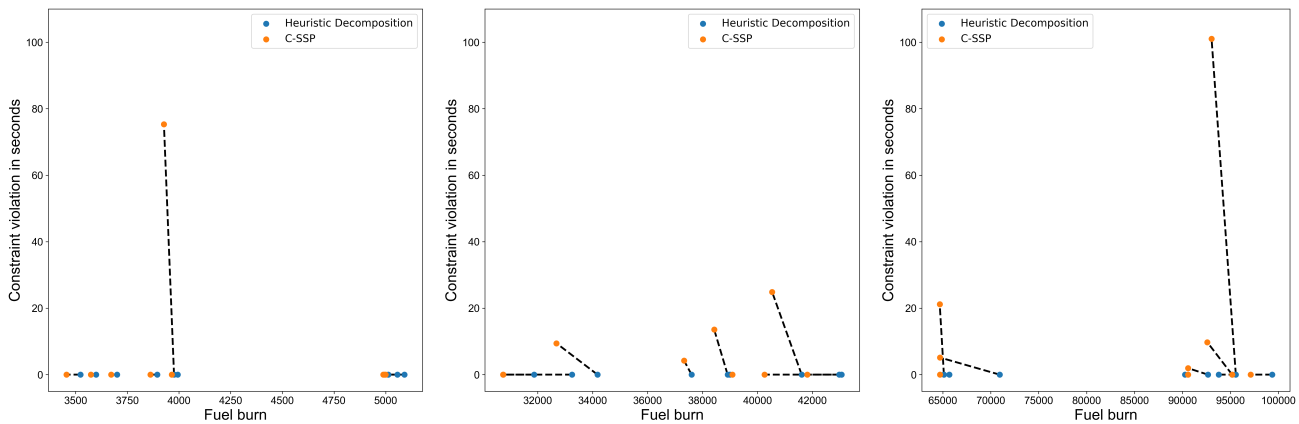

Figure 1: Comparison of time constraint violation and fuel consumption between C-SSP and heuristic decomposition.

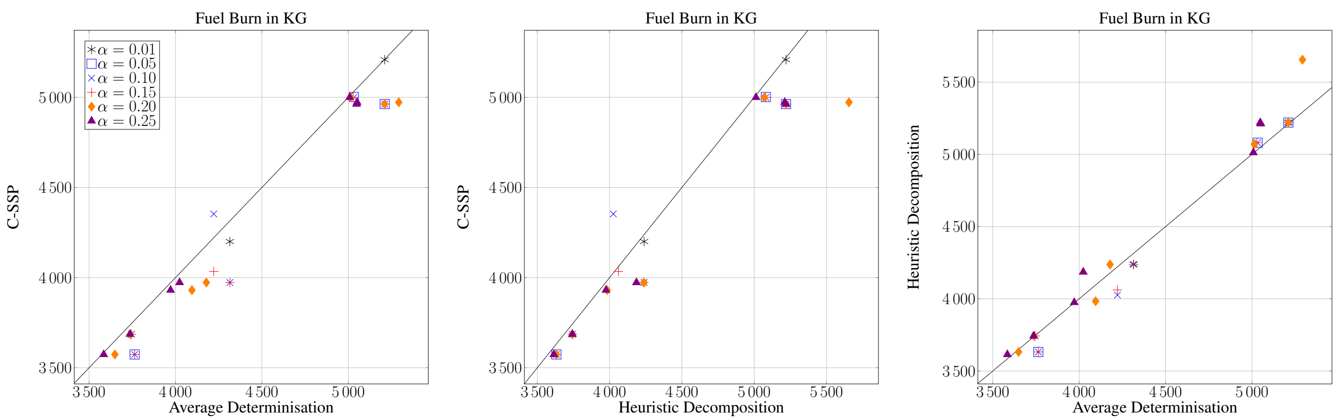

Figure 2: Comparison of average fuel consumption between the different techniques for short distance flights.

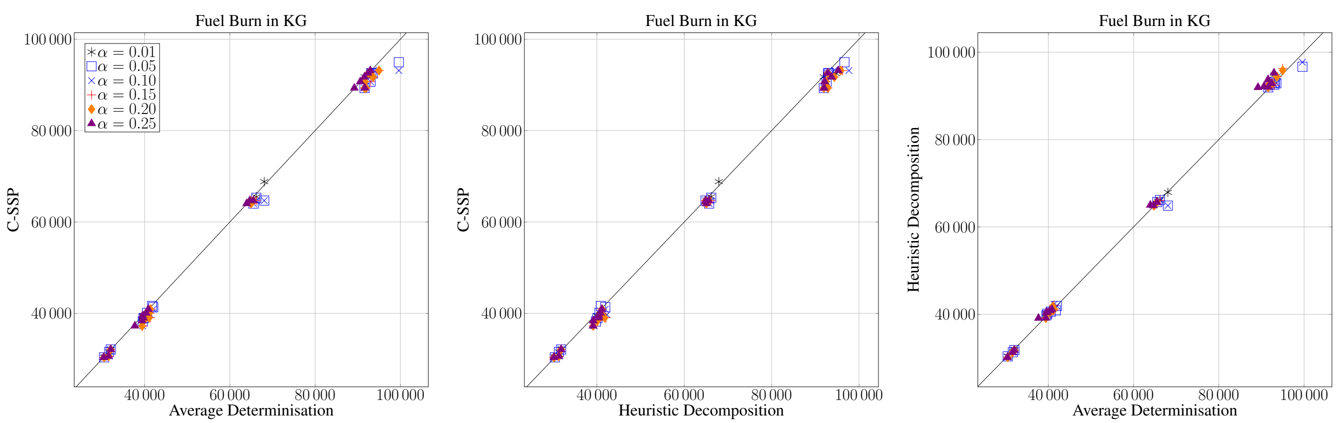

Figure 3: Comparison of average fuel consumption between the different techniques for medium and long distance flights.

dicator provided by the EPS forecast in conjunction with the performance model to estimate the convection value for any traversed edge. Recall that $\alpha \in \mathbb{R}$ denotes the maximum percentage of the distance that is allowed to go through convective areas. We perform several experiments with different bounds $\alpha \in \{0.01, 0.05, 0.1, 0.15, 0.2, 0.25\}$, which results in a total of $27 \cdot 6 = 162$ problems. For the convection cost scalar used by the deterministic approaches, we empirically chose $\omega(\alpha) = 5 + 95 \cdot \frac{(\alpha - 0.25)}{(0.01 - 0.25)}$ for the average determinisation as well as for the vertical planning phase of the heuristic decomposition, and $\omega(\alpha) = 0.5 + 24.5 \cdot \frac{(\alpha - 0.25)}{(0.01 - 0.25)}$ for the horizontal planning phase. We observed that due to

City Pair	Fuel Diff. (KG)	Flight Time (m)	Violation (s)
BOD–AGP	67	100.4	0.0
TLS–PRG	49	115.9	75.0
MUC–JMK	62	150.4	0.0
DXB–FCO	1494	369.7	9.4
DXB–LGW	490	448.5	13.5
JFK–MXP	1076	448.5	24.8
FRA–BKK	437	613.7	21.2
LAX–DXB	2588	909.1	9.7
DXB–IAH	2530	936.7	101.0

Table 1: Worst case constraint violation (in seconds) and flight times (in minutes) of policies generated by column generation for different city pairs. Fuel diff. is the difference of fuel burn compared to the policy generated by heuristic decomposition.

abstraction some policies that were determined to satisfy the constraints marginally miss the constraint when evaluated on the full problem. To compensate for this, we consider a constraint as satisfied if it holds for the bound $\alpha+0.01$. Once again, we first determine (with the C-SSP) which problems are solvable. In total, 28 problems are proven to be unsolvable, resulting in 134 remaining problems.

Again, with each approach we compute a policy for the problem and evaluate this policy on all 10 weather scenarios. Heuristic decomposition and average determinisation do not guarantee that a constraint is satisfied, and the abstraction underlying the C-SSP can also result in inaccuracies. However, all algorithms satisfied the constraint in 130 out of the 134 problems. We compare the average fuel burn between the algorithms for the problems where every algorithm satisfied the constraint. Figure 2 shows short distance flights, and Figure 3 depicts medium to long distance flights. Each data point corresponds to one flight problem on a specific convection constraint. Data points below the diagonal correspond to problems where the algorithm on the y axis required less fuel. The policy provided by C-SSP almost always results in less fuel consumption while guaranteeing constraints, although the difference is mostly less than one ton of fuel. A notable exception is a problem for a convection constraint of 0.1 – a consequence of the underlying deterministic policy. Results between the heuristic decomposition and average determinisation are mixed, but slightly in favour of the latter. We also evaluated the gap between the stochastic and the deterministic policies, which is in most cases below one ton of fuel, sometimes even below 50 KG. In 11 cases it exceeds a ton, up to of 2700 KG fuel in the best case.

7 Discussion

The experiments demonstrate some of the advantages and disadvantages of the C-SSP approach. First, the optimality guarantees allow us to determine beforehand if the problem is feasible in the first place. This is a useful property, as a reasonable constraint bound is not always clear from the beginning. Second, the approach significantly minimises fuel consumption and produces cheaper policies, and the expected cost of the stochastic policy provides a lower bound on what is possible to achieve.

The graphs we considered in the experiments were relatively small. While we also conducted experiments on larger graphs with more than 1000 nodes, the impact of the inaccuracy introduced by abstraction was significant; as a consequence, the policies returned by the C-SSP solver often violated the constraints when evaluated on the full problem. We conclude that the unavailability of heuristics for constraint cost functions, such as convection distance, is a serious hindrance and requires more research. Once such heuristics exist they are easily integrated into our framework.

Another drawback of the C-SSP approach is the current requirement of deterministic policies, as these do not always guarantee that constraints are met. However, we conjecture that once airline operations are fully autonomous, a stochastic policy will be a realistic solution. Moreover, we argue that for some constraints it is reasonable that they are guaranteed in expectation, i.e. when evaluated over multiple flights. A possible application is the avoidance of condensation trails (contrails) which manifest in the atmosphere as a result of aircraft traversal through specific areas, and which have been shown to have a significant impact on climate (Yin et al. 2018). Here, a stochastic policy that guarantees a lower bound on contrail manifestation in expectation can become an important tool to oppose the ongoing climate change.

Acknowledgements

This work was supported by the Airbus R&T project *Dynamic Operations Optimization Under Uncertainty for Aircraft and Satellite Applications*

Florian Geißer, Sylvie Thiébaux and Felipe Trevizan were also partially supported by ARC project DP180103446, *Online planning for constrained autonomous agents in an uncertain world*.

References

Ahuja, R. K.; Magnanti, T. L.; and Orlin, J. B. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc.

AIRBUS - Artificial Intelligence Research. 2020. scikit-decide library. https://github.com/airbus/scikit-decide.

AIRBUS Customer Services . 1998. Getting to grips with the cost index. *Flight Operations Support and Line Assistance* II.

Altman, E. 1999. *Constrained Markov Decision Processes*, volume 7. CRC Press.

Bertsekas, D., and Tsitsiklis, J. 1996. *Neuro-Dynamic Programming*. Athena Scientific.

Bertsimas, D., and Tsitsiklis, J. 1997. *Introduction to Linear Optimization*. Athena Scientific, 1st edition.

Cheung, J.; Brenguier, J.-L.; Heijstek, J.; Marsman, A.; and Wells, H. 2014. Sensitivity of flight durations to uncertainties in numerical weather prediction. In *4th SESAR Innovation Days*.

Desrosiers, J., and Lübbecke, M. E. 2005. A primer in column generation. In *Column generation*. Springer. 1–32.

Dolgov, D. A., and Durfee, E. H. 2005. Stationary deterministic policies for constrained MDPs with multiple rewards, costs, and discount factors. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, 1326–1331.

Erzberger, H.; Nikoleris, T.; Paielli, R. A.; and Chu, Y.-C. 2016. Algorithms for control of arrival and departure traffic in terminal airspace. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 230(9):1762–1779.

Eurocontrol. 2018. All-causes delay to air transport in europe. https://www.eurocontrol.int/publication/all-causes-delay-air-transport-europe-june-2018.

Franco, A.; Rivas, D.; and Valenzuela, A. 2017. Optimal aircraft path planning considering wind uncertainty. In *Proceedings of the 7th European Conference for Aeronautics and Space Sciences*, 3–6.

González-Arribas, D.; Soler, M.; and Sanjurjo-Rivo, M. 2016. Wind-based robust trajectory optimization using meteorological ensemble probabilistic forecasts. In *4th SESAR Innovation Days*.

Hansen, E. A., and Zilberstein, S. 2001. LAO*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence* 129(1):35–62.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2):100–107.

Knudsen, A. N.; Chiarandini, M.; and Larsen, K. S. 2017. Constraint handling in flight planning. In *Principles and Practice of Constraint Programming - 23rd International Conference, CP 2017*, 354–369.

Lübbecke, M. E. 2010. Column generation. *Wiley encyclopedia of operations research and management science*.

Murrieta Mendoza, A. 2013. *Vertical and lateral flight optimization algorithm and missed approach cost calculation*. Ph.D. Dissertation, École de technologie supérieure.

Nilim, A.; El Ghaoui, L.; and Duong, V. 2002. Robust dynamic routing of aircraft under uncertainty. In *Proceedings of the 21st Digital Avionics Systems Conference*, volume 1, 1.A.5–1–1.A.5–13.

Nilsson, N. J. 1968. Searching problem-solving and game-playing trees for minimal cost solutions. In *Information Processing, Proceedings of IFIP Congress 1968*, 1556–1562.

Nuic, A.; Poles, D.; and Mouillet, V. 2010. BADA: An advanced aircraft performance model for present and future atm systems. *International Journal of Adaptive Control and Signal Processing* 24:850 – 866.

Palmer, T. 2019. The ECMWF ensemble prediction system: Looking back (more than) 25 years and projecting forward 25 years. *Quarterly Journal of the Royal Meteorological Society* 145(S1):12–24.

Taylor, C.; Liu, S.; Wanke, C.; and Stewart, T. 2018. Generating diverse reroutes for tactical constraint avoidance. *Journal of Air Transportation* 26(2):49–59.

Trevizan, F.; Thiébaux, S.; Santana, P.; and Williams, B. 2016. Heuristic search in dual space for constrained stochastic shortest path problems. In *Proceedings of the 26th International Conference on Automated Planning and Scheduling (ICAPS)*, 326–334.

Trevizan, F.; Thiébaux, S.; and Haslum, P. 2017. Occupation Measure Heuristics for Probabilistic Planning. In *Proceedings of the 27th International Conference on Automated Planning and Scheduling (ICAPS)*, 306–315.

Yin, F.; Grewe, V.; Frömming, C.; and Yamashita, H. 2018. Impact on flight trajectory characteristics when avoiding the formation of persistent contrails for transatlantic flights. *Transportation Research Part D: Transport and Environment* 65:466–484.