Mixed Integer Second-Order Cone Programming for the Horizontal and Vertical Free-flight Planning Problem

Zhi Yuan[a], Liana Amaya Moreno[a], Armin Fügenschuh[a], Anton Kaier[b], Amina Mollaysa[a], Swen Schlobach[b]

[a]*Professorship of Applied Mathematics, Department of Mechanical Engineering, Helmut Schmidt University, Hamburg, Germany*
[b]*Lufthansa Systems AG, Kelsterbach, Germany*

Abstract

In the past, travel routes for civil passenger and cargo air traffic were aligned to the air traffic network (ATN). To resolve the network congestion problem, the free-flight system has recently been introduced in more and more regions around the globe, allowing flight operations to make full use of the four space-and-time dimensions. For the numerical computation of optimal flight trajectories under free-flight conditions, we separate the problem into a horizontal and a vertical optimization problem, and develop mixed integer nonlinear programs for both. It turns out that both models have second-order cones (SOC) as substructures. We use different modelling strategies for dealing with the SOC and other nonlinear structures, so that after a linear approximation we are able to apply a mixed-integer linear solver to compute global optimal free-flight trajectories.

Keywords: Free-flight planning, second-order cone programming, mixed-integer linear programming

1. Introduction

Nowadays, most flight trajectories must be aligned to an air traffic network (ATN), which is a two-dimensional (2D) earth spanning directed graph. The ATN was established several decades ago, at the dawning age of the commercial air travel in the 1950s, where the increasing number of flights created the demand for a safe system that could be surveilled from the ground and obeyed in the sky with the radar and navigation technology of that time. Since then, the international civil air traffic for passengers and cargo has doubled every 15 years, and is expected to continue its growth at this rate in the foreseeable future [1]. The rapid growth in commercial air traffic is putting immense pressure on the air traffic control (ATC).

Due to the finite structure of the graph, some parts of it are highly congested, so that special rules are introduced in order to avoid such congestions. These detours lead to suboptimal trajectories, leading to higher emissions and higher operating costs. For modernizing the ATC system to meet the demands for enhanced capacity, efficiency, and safety, the concept of free-flight was proposed by the Radio Technical Commission for Aeronautics (RTCA) [2].

The free-flight concepts have currently been applied in many airspaces around the world, including the Pacific Ocean, and many countries in Europe. It is planned to be implemented in almost the entire Europe by 2019 [3], and will foreseeably be established as standard over the next decades in most parts of the world. There are several versions of free-flight in the ongoing discussion. The greatest degree of freedom is airborne free-flight, which allows an arbitrary flight curve in the 4D space. But practically more relevant are restricted versions of free-flight, where the ATC still plays a central role in the surveillance of the airspace. In our approach, we assume such restricted free-flight situation, where a route needs to be computed before the take-off for fuel estimation and ATC validation, and airplanes can change the altitude, direction, and speed only at certain validated point, and can cruise only at certain admissible flight altitudes. We split the 4D planning problem into a horizontal optimization phase (2D trajectory on the earth's surface) and a vertical optimization phase (altitude plus time). This is in line with current industrial practice [4], because the full 4D problem is too difficult for the current computer hardware and solver software technology.

We develop mathematical models in order to calculate the free-flight trajectory for the cruise phase of the flight. The objective is to minimize the fuel consumption for a trip between given departure and arrival airports. Also given are a (forecasted) wind field, and aircraft performance data measuring unit distance fuel consumption depending on the altitude, the current weight, and the speed of the airplane. For flight planning on the ATN graph, usually graph-based algorithms are used, for example, Dijkstra's shortest path algorithm [4]. Since there is no underlying graph in the free-flight problem, new algorithmic approaches have to be developed. We formulate the two problems (horizontal, vertical) as mixed-integer nonlinear programming (MINLP) problems.

The MINLP problem is until today a class of computationally challenging problem. The development of the mixed integer linear programming (MILP) approaches in the last five decades may provide efficient techniques for solving MINLP problems [5]. One potential approach is to linearize the nonlinear formulations in MINLP. In the MINLP of the free-flight planning problems, the integer variables include the selection of a discrete altitude and the binary variables for piecewise linear interpolations; the nonlinear constraints can be classified as second-order cones (SOC). Lobo et al. [6] describe several applications of SOC Programs (SOCP)

to abstract problems such as robust least-squares as well as real-world planning problems such as filter design, equilibrium of spring systems, etc. Such SOC constraints can be approximated by linear ones using the lifted polyhedral relaxation approaches proposed by Ben-Tal and Nemirovski [7]. This linear approximation approach is further refined by Glineur [8], who tested it on the second-order cone programming (SOCP) problems without integer variables, and did not find the polyhedral relaxation computationally superior to solving the SOCP by a default interior point solver. Although the polyhedral relaxation approach may not be practically preferable for the continuous SOCP, it is shown to be especially useful for solving Mixed Integer SOCP (MISOCP) in a number of practical applications, e.g. portfolio optimization [9], train routing [10], or soft rectangle packing [11]. Another challenge in our MINLP model is to linearly interpolate the grid measured data of the wind field as well as the fuel consumption. To this end, standard piecewise linear interpolation techniques can be applied to formulate such problem into MILP models, see [12] for a recent survey. The MILP formulation of piecewise linear functions have been successfully applied in a number of practical applications such as gas network optimization [13], process engineering [14], etc.

Our contributions in this article are the following: (i) the first formulation of the horizontal and vertical free-flight planning problem as a MIP model; (ii) identify the second-order cone structure in both MIP models; (iii) computationally study different approaches for the second-order cone constraints and piecewise linear functions, as well as the use of parallel threads in solving this problem.

2. Freeflight Planning: The Problem Description

The flight planning problem concerns finding a fuel-optimal trajectory for an aircraft from the departure airport to the destination airport. Such a flight trajectory is four-dimensional: a horizontal two-dimensional trajectory on the surface of the globe; a vertical dimension consisting of discrete levels of flight altitudes; and a time dimension controlled by aircraft speed. Due to the computational difficulty of such a four-dimensional problem, it is usually divided into two separate phases [4]. In the first horizontal phase, an optimal trajectory on the earth's surface is searched; and in the subsequent vertical phase, the optimal altitude and speed are assigned to each of the segments that compose the optimal horizontal trajectory.

2.1. Horizontal Optimization

Horizontally, the shortest trajectory connecting two points on the surface of the globe is the great circle. However, such a shortest trajectory may not be preferable in practice, since wind plays an important role in the fuel consumption during a flight. A fuel-efficient flight trajectory should utilize as much as possible the tail

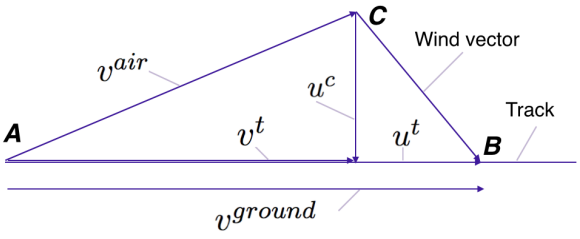


Figure 1: Wind triangle. v^{ground} denotes the ground speed; v^{air} denotes the aircraft speed; v^t and u^t denote the aircraft speed and wind speed in the track direction, respectively; u^c denotes the cross-track wind speed.

wind from behind, while avoiding the strong head wind antiparallel to the flight direction. Due to the ATC requirements and the limitation of the current flight management system, a free-flight trajectory cannot be an arbitrary curve, and needs to follow a straight line between any two specified locations. The task in the horizontal phase is to specify a set of ordered locations such that the segments connecting the adjacent locations comprises the trajectory. In order to ease the workload of the pilot, and to guarantee the passenger comfort, rules have been set up such that there should be at least a certain time period, e.g., 10 to 15 minutes, between two adjacent locations where flight direction can be changed. Note that for flying from A to B in the presence of wind as in Figure 1, the aircraft actually flies along the vector from A to C with a *true air speed* v^{air}. The distance $|\overrightarrow{AB}|$ and the speed v^{ground} observed from the ground is called the *ground distance* and *ground speed*, respectively, and the distance $|\overrightarrow{AC}|$ is called the *air distance*.

2.2. Vertical Optimization

The optimal trajectory found in the horizontal optimization phase consists of a number of segments. In the vertical optimization phase, to each of these segments an altitude and a speed will be assigned. Vertically, a flight consists of five stages: take-off, climb, cruise, descent, and landing. Here, we focus on the cruise stage, since it consumes most of the fuel and time during a flight, while the other stages are relatively short and not flexible due to safety restrictions and reglementations.

The cruise stage starts after the initial climb has brought aircraft above the crossover altitude of around 29,000 feet. Depending on the flight direction (eastward or westward), different *flight altitudes* are allowed. For the admissible flight altitudes, we followed the Instrumental Flight Rules (IFR) with Reduced Vertical Separation Minima (RVSM) [15], which has been introduced in most airspaces in the world during the last 15 years. It allows usually 2,000 feet between two adjacent admissible flight altitudes in the same flight direction. Due to safety control as well as pilot and passenger comfort rules, each aircraft is only allowed to change its altitude and speed at the beginning of each segment.

The speed for each segment should be also determined, such that fuel efficiency is ensured, and the time constraint is satisfied. Note that the trajectory is briefed

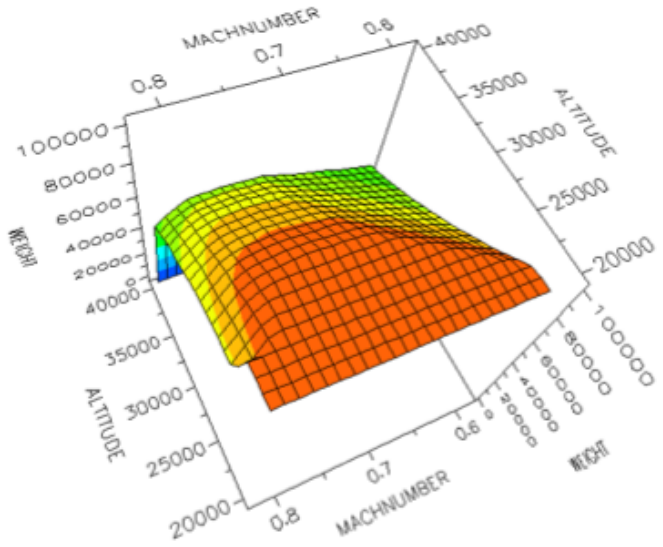


Figure 2: Grid data of unit distance fuel consumption with respect to aircraft weight (in kg), altitude (in feet), and speed (in Mach number).

to the pilot one hour before the take-off. However, there are uncertainties in the computation, for example, the actual aircraft weight is only known shortly prior to the flight. Besides, delay may be caused due to disruptions such as undesired weather conditions, unexpected maintenance or runway availability. Such disruption is usually confirmed shortly before the flight. To make up for the delay such that the passenger transit and aircraft connections can be met, a time constraint is usually imposed in the vertical phase, so that it is only required to change the speed and altitude of the briefed flight plan without changing completely the trajectory.

2.3. Aircraft Performance

The aircraft performance data concerns how much fuel is consumed by flying one unit air distance under various depending factors, including aircraft weight, its true air speed, and flight altitude. An example of such performance data is visualized in Figure 2. In general, the heavier the aircraft is, the more fuel it consumes per distance, vice versa, the higher it flies, the less fuel it consumes. The optimal speed usually depends on other factors such as altitude and wind. These data are provided by the aircraft manufacturers, such as Airbus or Boeing. The aircraft manufacturers have divided each of the depending factors into a number of discrete levels, such that the combination of all the levels of each factor form a grid. The unit distance fuel consumption is measured at each grid point with the highest possible accuracy. If a combination of factors weight, speed, and altitude does not exactly lie on a grid point, it is interpolated by the closest grid points around

it. Such local interpolation can be done linearly. So in such case, the global performance function can be seen as a piecewise linear function. Although it is also possible to use a continuous global function to approximate the performance data, such as in [16, 17], this may result in accuracy loss – the fuel estimation accuracy is the primary concern of a flight planning system.

2.4. Upper Air Weather

The weather data that affect our flight planning problem is mainly the wind. (The temperature may affect fuel consumption in the other flight stages than cruise, thus is not considered in this work.) These upper air weather prediction data is updated every 6 hour, e.g., 0, 6, 12 o'clock, etc. In each update, the data contains a weather prediction in every 3 hours, e.g., 0, 3, 6, 9 o'clock etc., for the next 36 hours. The wind information at 2 o'clock for example, needs to be interpolated linearly from the wind data given at 0 and 3 o'clock. Besides the time dimension, the wind is given above the earth in 13 different altitudes. If the flight altitude is not on the exact wind altitude, its wind needs to be interpolated linearly by the wind data of the two closest wind altitudes. Moreover, horizontally, the wind data is available on the surface of the globe by a $1.25^\circ \times 1.25^\circ$ longitude-latitude grid, and again, the wind between the grids should be obtained by linear interpolation. Note that the fixed-degree longitude-latitude grid may not be square: the distance of one longitude degree at the equator is 60 nautical miles, then it decreases as latitude increases, and approaches 0 at the pole as latitude approaches 90°. However, each grid unit locally can be treated as a trapezoid. Similar as the aircraft performance data shown in Section 2.3, the wind data can be regarded as a piecewise linear function.

3. Survey of Literature

The flight planning problem is a four-dimensional optimization problem. Most of the approaches in academia as well as in practice solve this 4D problem with two phases [4]: a horizontal phase that optimizes a 2D trajectory, and a vertical phase that assigns altitude and speed to the trajectory.

In the horizontal phase, conventionally, a trajectory has to follow the ATN. For a network-based 2D trajectory optimization, usually graph algorithms such as A^* [18] or Dijkstra's algorithm [4] are applied if no restrictions from the air traffic control (ATC) is considered. If the ATC restrictions such as airspace capacities are considered for avoiding congestion, this problem can be formulated as a mixed integer programming (MIP) model and solved by a branch-and-price approach [19]. The introduction of the free-flight concept [2] tries to solve the airspace congestion problem by freeing each flight trajectory from the ATN network. Theoretically,

such a free-flight trajectory can be an arbitrary curve connecting the departure and destination. Wickramasinghe et al. [17] formulated the 4D free-flight trajectory optimization as an optimal control problem, and solved it with dynamic programming approach to obtain a smooth 4D curve. However, such a theoretical 4D curve is not feasible in practice, since on the vertical dimension, only certain discrete flight altitudes are allowed for cruise by the ATC. Ng et al. [20] considered the discrete admissible cruise altitudes in their trajectory optimization by a separation of a vertical phase and a horizontal phase. The optimal vertical profile by time computed in the vertical phase is used as an estimate for the wind and aircraft performance function input for the horizontal trajectory optimization. Assuming constant speed, the horizontal optimization is reduced to finding a minimum-time trajectory, and can be solved by an optimal control problem assuming a differentiable function for the wind field exists. There were two practical drawbacks in this work: firstly, they assumed the speed was constant, while it has been shown that using variable speed consumes less fuel than using constant speed [21], especially in the presence of wind; secondly, in the case of passenger flight, minimizing flight time does not necessarily lead to lower operating cost, since early arrival may result in an extra cost for the usage of a waiting area and a boarding gate. The first attempt of formulating the free-flight horizontal optimization problem into a MIP model was presented in the master thesis of Peter [16]. The practical drawback in [16] as well as in the optimal control approaches [20, 17] is the assumption of an explicitly defined function for the wind field and the fuel consumption available. Such functions are also required to be differentiable for the optimal control approach to work. However, in practice, both the wind data and aircraft performance data are provided as measured data in a 2D grid, and need to be interpolated locally.

In the vertical optimization aspect, the task is generally to assign speed and altitude to a flight trajectory. Computing an optimal altitude profile in the absence of wind can also provide estimated altitudes for the 2D horizontal trajectory optimization [20]. Such a steady-atmosphere optimal altitude profile increases approximately linearly as fuel burns. The altitude profile becomes irregular if altitude-dependent wind is considered [22]. A recent research by Lovegren and Hansman [23] confirmed a potential fuel saving of up to 3.5% by reassigning only altitude and speed to fixed flight trajectories, based on a study of 257 real flight operations in US. However, no time constraint is taken into account in these works as in practice. Time constraints are commonly considered in airline operations to reduce delays due to various disruptions, such as undesirable weather conditions, unexpected maintenance requirements, etc. Such delays are typically recovered by increasing cruise speed, such that the passenger connections as well as the connections for the aircraft and the crews are satisfied [24]. Aktürk et al. [24] modelled explicitly the time constraint into their MIP model in the context of air-

craft rescheduling. However, their mathematical model only considered assigning a constant cruise speed for the whole flight. Hagelauer and Mora-Camino [25] considered time constraints in the vertical flight planning, and used a heuristic based dynamic programming approach where global optimality can not be guaranteed. Furthermore, Yuan et al. [21] proposed a MIP model for solving the vertical flight planning problem with variable speed during a flight. It is experimentally confirmed that the use of variable speed leads to further fuel saving comparing with constant speed. In this work, our vertical model extends [21] by taking the weather influence such as wind into account.

4. Mathematical Preliminaries

Our MINLP free-flight planning models use 2D piecewise linear functions to interpolate the wind field and the aircraft performance data. The rest of the nonlinear constraints are identified as second-order cones. The techniques to formulate them as an MILP are presented in this section.

4.1. Modelling 2D Piecewise Linear Functions

Let X and Y be one-dimensional vectors of discrete values, and thus $X \times Y$ defines a two dimensional grid. Let $F_{i,j}$ be a two dimensional matrix of measured data on each grid point $(i, j) \in X \times Y$. We denote $\widehat{F}(x, y)$ the 2D piecewise linear functions interpolating F for all the continuous values x and y within the given grid of $X \times Y$. The 2D piecewise linear interpolation can be formulated as a mixed integer linear programming model using either Danzig's *convex combination method* [26] (or *lambda method*), or the *incremental method* [27] (or *delta method*). Wilson [28] further generalized both methods to higher dimensions.

4.1.1. Convex Combination (Lambda) Method

The $X \times Y$ grid is first partitioned by a set of triangles K, as shown in Figure 3. For each triangle $k \in K$, $X_{k,n}$, $Y_{k,n}$ and $F_{k,n}$ denote the X-, Y-, and F-value of a vertex $n \in N := \{1, 2, 3\}$ of triangle k, respectively. Each triangle is assigned a binary variable w_k, w_k equals 1 if (x, y) is inside triangle k. We further introduce for each triangle $k \in K$ three continuous variables $\lambda_{k,n} \in \mathbb{R}^+$ with $n \in N$ such that

$$\sum_{k \in K} w_k = 1 \tag{1a}$$

$$\sum_{n \in N} \lambda_{k,n} = w_k \qquad \forall k \in K \tag{1b}$$

$$x = \sum_{k \in K, n \in N} \lambda_{k,n} \cdot X_{k,n} \tag{1c}$$

$$y = \sum_{k \in K, n \in N} \lambda_{k,n} \cdot Y_{k,n} \tag{1d}$$

$$\widehat{F}(x, y) = \sum_{k \in K, n \in N} \lambda_{k,n} \cdot F_{k,n} \tag{1e}$$

where (1a) ensures only one triangle is selected, (1b) sums λ of each triangle to 1 only if the triangle is selected, together with (1c) and (1d), the value of non-zero λ is determined, such that the fuel estimation at a point (x, y) is given by (1e) as a convex combination of λ and the grid value.

4.1.2. Incremental (Delta) Method

The set of triangles $K := \{k_1, k_2, \ldots, k_m\}$ is ordered such that the *ordering assumption* [28] is fulfilled: This requires each pair of adjacent triangles k_i and k_{i+1} (for $i \in \{1, \ldots, m-1\}$) to have common vertices, and the vertices $N_i = \{n_{i,0}, n_{i,1}, n_{i,2}\}$ of each triangle k_i are ordered in a way that the ending vertex $n_{i,2}$ of k_i coincides with the starting vertex $n_{i+1,0}$ of the next triangle k_{i+1} for each pair of adjacent triangles. An example of such an ordering is illustrated in Figure 3 with red arcs, which connects the starting vertex of the previous triangle k_i to the starting vertex of the next triangle k_{i+1}. Each triangle k is assigned with a binary variable w_k and two real-valued variables $\delta_{k,n} : n \in \{1, 2\}$ with $0 \leq \delta_{k,n} \leq 1$, such that

$$\sum_{n \in \{1,2\}} \delta_{k+1,n} \leq w_k \qquad \forall k \in K \setminus k_m \tag{2a}$$

$$\delta_{k,2} \geq w_k \qquad \forall k \in K \setminus k_m \tag{2b}$$

$$\sum_{n \in \{1,2\}} \delta_{k,n} \leq 1 \qquad \forall k \in K \tag{2c}$$

$$x = X_{1,0} + \sum_{k \in K, n \in \{1,2\}} \delta_{k,n} \cdot (X_{k,n} - X_{k,0}) \tag{2d}$$

$$y = Y_{1,0} + \sum_{k \in K, n \in \{1,2\}} \delta_{k,n} \cdot (Y_{k,n} - Y_{k,0}) \tag{2e}$$

$$\widehat{F}(x, y) = F_{1,0} + \sum_{k \in K, n \in \{1,2\}} \delta_{k,n} \cdot (F_{k,n} - F_{k,0}). \tag{2f}$$

The equations (2a) and (2b) guarantee the *generalized filling condition* [28], i.e., for all $k \in K, n \in \{1, 2\}$, $\delta_{k,n}$ can be positive only if $\delta_{k-1,2} = 1$; (2c) constrains that all the δ variables of the same triangle sum up to no more than 1; the values of the δ variables in the triangle which contains the point (x, y) are determined in (2d) and (2e), such that the function value $\widehat{F}(x, y)$ can be linearly interpolated by (2f).

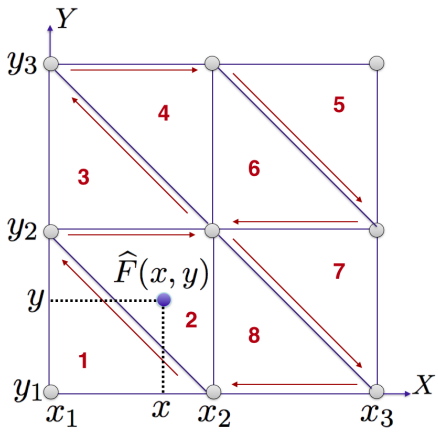


Figure 3: An example of triangulating a two dimensional grid $X \times Y$. The value of the piecewise linear function $\widehat{F}$ at a point (x, y) can be calculated as a linear combination of the F values at the three grid points that form its containing triangle, namely, $(x_1, y_2), (x_2, y_1), (x_2, y_2)$ of triangle 2. An example of an triangle order that fulfils the *ordering assumption* for the delta method is marked in red, including the red arc in each triangle k_i that points from n_i^0 to n_i^2 (see the text for more details).

4.2. *Linear Approximation of Second Order Cone*

A second order cone in n dimensions can be expressed as

$$Q_n = \left\{ x \in \mathbb{R}^n : x_1 \geq \sqrt{\sum_{i=2}^{n} x_i^2} \right\}.$$

Particularly in this work, we are interested in the second-order cone $Q_3 \in \mathbb{R}^3$. Ben-Tal and Nemirovski [7] proposed a *lifted polyhedral relaxation* approach to approximate a second-order cone by linear inequalities. This approach is further refined by Glineur [8]. Firstly, auxiliary continuous variables $\alpha_j, \beta_j \in \mathbb{R}$ for $j = 0, 1, \ldots, J$ are introduced, where the *approximation level* parameter J controls the approximation accuracy. Then the second-order cone Q_3 can be approximated by the following linear equations:

$$\alpha_0 = x_2, \quad \beta_0 = x_3, \tag{3a}$$

$$\alpha_{j+1} = \cos\left(\frac{\pi}{2^j}\right) \cdot \alpha_j + \sin\left(\frac{\pi}{2^j}\right) \cdot \beta_j, \qquad j = 0, 1, \ldots, J-1, \tag{3b}$$

$$\beta_{j+1} \geq -\sin\left(\frac{\pi}{2^j}\right) \cdot \alpha_j + \cos\left(\frac{\pi}{2^j}\right) \cdot \beta_j, \qquad j = 0, 1, \ldots, J-1, \tag{3c}$$

$$\beta_{j+1} \geq \sin\left(\frac{\pi}{2^j}\right) \cdot \alpha_j - \cos\left(\frac{\pi}{2^j}\right) \cdot \beta_j, \qquad j = 0, 1, \ldots, J-1, \tag{3d}$$

$$x_1 = \cos\left(\frac{\pi}{2^J}\right) \cdot \alpha_J + \sin\left(\frac{\pi}{2^J}\right) \cdot \beta_J. \tag{3e}$$

Each second-order cone can be linearly approximated by adding $2 \cdot J$ additional variables and $3 \cdot J + 3$ additional constraints. The approximation error is shown in [7] and [8] to be

$$\epsilon = \left(\cos\frac{\pi}{2^J}\right)^{-1} - 1.$$

The approximation level J trades off accuracy for efficiency. Larger values of J lead to more accurate approximation but increase the model size. Its value in our problem setting will be discussed in later sections.

5. A Model for Horizontal Optimization

In the horizontal optimization phase, we are given a predetermined total flight time T. The total flight time T is discretized into n equal time intervals, without loss of generality, $0 = t_0 < t_1 < t_2 < \ldots < t_n = T$, such that the pilot and passenger comfort requirements are met (see Section 2.1). The set of the resulting n flight segments are denoted by S, each of which has a flight time of $\Delta T = T/n$. The departure location and the destination location are denoted as (X^{start}, Y^{start}) and (X^{end}, Y^{end}). Furthermore, the dry aircraft weight W^{dry} is given, i.e., the weight of a loaded aircraft without trip fuel (reserve fuel for safety is included in the dry weight). The unit distance fuel consumption $F_{v,w}$ is defined on each speed-weight grid point $(v, w) \in V \times W$. The wind is a vector field $U : \mathbb{R}^2 \to \mathbb{R}^2$, where we denote the eastward- (or x-) component by $U^x(x, y)$ and northward- (or y-) component by $U^y(x, y)$, which are given on the grid points $(x, y) \in X \times Y$ on the earth surface.

The variables include the position vectors x_i and y_i at time $t_i, i \in S \cup \{0\}$; the aircraft's true air speed v_i and its x- and y-component v_i^x and v_i^y at a segment $i \in S$; the x- and y-component of the wind speed u_i^x and u_i^y at time $t_i, i \in S \cup \{0\}$, and the u_i^{x-avg} and u_i^{y-avg} denote the average wind influence on a segment $i \in S$; the aircraft weight w_i at time $t_i, i \in S \cup \{0\}$ and w_i^{avg} for $i \in S$ where w_{i-1}, w_i^{avg}, and w_i denote the start, average, and end weight at a segment i; and the fuel f_i consumed on a segment $i \in S$. The mathematical model for the horizontal optimization can be formulated as follows:

$$\min \quad w_0 - w_n \tag{4}$$

$$\text{s.t.} \quad (x_0,\ y_0) = (X^{start},\ Y^{start}),\quad (x_n,\ y_n) = (X^{end},\ Y^{end}) \tag{5}$$

$$x_i = x_{i-1} + (v_i^x + u_i^{x-avg}) \cdot \Delta T,\quad \forall i \in S \tag{6}$$

$$y_i = y_{i-1} + (v_i^y + u_i^{y-avg}) \cdot \Delta T,\quad \forall i \in S \tag{7}$$

$$2u_i^{x-avg} = u_i^x + u_{i-1}^x,\quad 2u_i^{y-avg} = u_i^y + u_{i-1}^y,\quad \forall i \in S \tag{8}$$

$$u_i^x = \widehat{U}^x(x_i, y_i),\quad u_i^y = \widehat{U}^y(x_i, y_i),\quad \forall i \in S \cup \{0\} \tag{9}$$

$$v_i^2 = (v_i^x)^2 + (v_i^y)^2,\quad \forall i \in S \tag{10}$$

$$w_n = W^{dry} \tag{11}$$

$$w_{i-1} = w_i + f_i,\quad \forall i \in S \tag{12}$$

$$2 \cdot w_i^{avg} = w_i + w_{i-1},\quad \forall i \in S \tag{13}$$

$$f_i = v_i \cdot \Delta T \cdot \widehat{F}(v_i, w_i^{avg}), \quad \forall i \in S. \tag{14}$$

The objective function (4) minimizes the total fuel consumption, i.e., the fuel burnt during the flight. The departure and destination position is fixed by (5). By the equation of motion (6) and (7), each flight segment in each time interval is travelled in a straight line, by considering the aircraft speed as well as the wind speed. The wind speed on a segment is calculated in (8) by averaging the wind speed at the starting and ending point of the segment. This estimation is accurate and widely adopted in practice when the segment is short enough, e.g., under 100 nautical miles. The wind speed at a given position (x, y) can be calculated by a two-dimensional piecewise linear functions $\widehat{U}^x(x, y)$ and $\widehat{U}^y(x, y)$ that interpolates U^x and U^y linearly by all the continuous values in the grid of $X \times Y$. The aircraft speed is calculated by the quadratic equality in (10). Constraint (11) initializes the weight vector by assuming all trip fuel is burnt during the flight; weight consistency is ensured by (12), and the average weight of each segment calculated in (13) will be used in the calculation of the fuel consumption in (14), where $\widehat{F}(v, w)$ is a piecewise linear function interpolating F for all the continuous values of v and w within the given grid of $V \times W$.

6. A Model for Vertical Optimization

In the vertical optimization phase, a set of n segments $S := \{s_1, \ldots, s_n\}$ computed in the horizontal optimization phase is given as input. The length of these segments are denoted by L_i for all $s_i \in S$. Note that the segments are defined by ground distance in the vertical optimization phase, since the speed and altitude can be only changed at the beginning of each segment. Since the segments are given in advance, the wind for a segment i at a cruise altitude h can be precomputed, namely, $U^t_{i,h}$ for wind speed in the track direction, i.e., the flight direction; and $U^c_{i,h}$ for the wind speed in the cross-track direction, i.e., perpendicular to the flight direction. Also given are the minimum and maximum trip duration $\underline{T}$ and $\overline{T}$, and a number of admissible flight altitudes H. The unit distance fuel consumption F depends on three factors: air speed, weight, and altitude. Since only a number of discrete flight levels H is allowed, F can be precomputed for each admissible altitude $h \in H$ as $F_h(v, w)$ for a grid $(v, w) \in V \times W$. We further denote $\widehat{F}_h(v, w)$ the piecewise linear function that interpolates for all continuous values (v, w) within the grid of $V \times W$. Additional variables in the vertical model include time Δt_i for travelling a segment $i \in S$; aircraft speed decomposition in the track direction v^t_i and cross-track direction v^c_i for each segment $i \in S$; similarly, wind speed decomposition in the track direction u^t_i and cross-track direction u^c_i for each segment $i \in S$; and binary variables $\mu_{i,h}$ that indicates whether a segment $i \in S$ is flown on an altitude $h \in H$. The

other notations are adopted from horizontal model in Section 5, such as parameters W^{dry} for dry weight, and variables time t_i, weight w_i and w_i^{avg}, aircraft speed v_i, and segment fuel consumption f_i. A general mathematical model for the vertical flight planning problem can be stated as follows:

$$\min \quad w_0 - w_n \tag{15}$$

$$\text{s.t.} \quad t_0 = 0, \quad \underline{T} \le t_n \le \overline{T} \tag{16}$$

$$\Delta t_i = t_i - t_{i-1} \quad \forall i \in S \tag{17}$$

$$L_i = v_i^{ground} \cdot \Delta t_i \quad \forall i \in S \tag{18}$$

$$v_i^{ground} = v_i^t + u_i^t \quad \forall i \in S \tag{19}$$

$$(v_i^{air})^2 = (v_i^t)^2 + (u_i^c)^2 \quad \forall i \in S \tag{20}$$

$$\sum_{h \in H} \mu_{i,h} = 1, \quad \forall i \in S \tag{21}$$

$$u_i^t = \sum_{h \in H} \mu_{i,h} \cdot U_{i,h}^t, \quad \forall i \in S \tag{22}$$

$$u_i^c = \sum_{h \in H} \mu_{i,h} \cdot U_{i,h}^c, \quad \forall i \in S \tag{23}$$

$$w_n = W^{dry} \tag{24}$$

$$w_{i-1} = w_i + f_i, \quad \forall i \in S \tag{25}$$

$$w_{i-1} + w_i = 2 \cdot w_i^{avg}, \quad \forall i \in S \tag{26}$$

$$f_i = v_i^{air} \cdot \Delta t_i \cdot \sum_{h \in H} \mu_{i,h} \cdot \widehat{F}_h(v_i^{air}, w_i^{avg}), \quad \forall i \in S. \tag{27}$$

Objective (15) minimizes the total fuel consumption; constraint (16) enforces the flight duration within a given interval; (17) ensures time consistency; the ground distance is calculated by the basic equation of motion on each segment in (18); the ground speed and air speed should satisfy the wind triangle (see Figure 1) as formulated in (19) and (20); the weight vector is initialized in (24) by assuming all trip fuel is burnt during the flight; weight consistency is ensured in (25), and the average weight of each segment calculated in (26) will be used in the calculation of segment fuel consumption in (27), where the unit distance fuel consumption $\widehat{F}_h$ on an altitude h selected by $\mu_{i,h}$ given air speed v_i^{air} and average weight w_i^{avg}, is multiplied by the air distance as a product of the air speed v_i^{air} and travel time Δt_i.

7. Solving the Model

Solving the MINLP models presented in Section 5 and 6 with the numerical solver SCIP takes prohibitively long computation time to reach the optimal or even

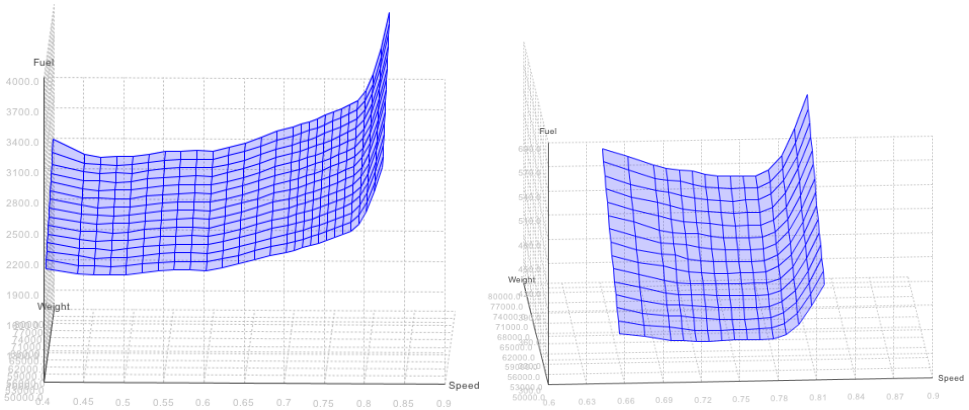


Figure 4: Left is the segment fuel consumption $\widehat{F^{\Delta T}}(v, w)$ per unit time ΔT given air speed v in mach number and aircraft weight w in KG. Right is the segment fuel consumption $F^L_{h^*}(v, w)$ on ground distance L with an optimal altitude h^* given v and w.

a feasible solution for a realistic-size instance. In this section, some techniques for reformulating the MINLP models are shown. In particular, some linearization techniques for nonlinear constraints are presented.

7.1. Fuel Consumption Linearization by Performance Data Preprocessing

The fuel consumption per segment given by equation (14) of the horizontal model and (27) of the vertical model are nonlinear functions. They calculate the segment fuel consumption by multiplying the unit distance fuel consumption $\widehat{F}$ with the air distance calculated as the product of aircraft speed and travel time. They can be linearized by preprocessing the performance data. For example, the fuel consumption on a segment of a fixed flight time ΔT, denoted by $F^{\Delta T}(v, w)$, can be exactly precomputed as

$$F^{\Delta T}(v, w) = F(v, w) \cdot v \cdot \Delta T, \qquad \forall (v, w) \in V \times W,$$

since the aircraft speed v is a given parameter, and the travel time ΔT is a constant. Then instead of interpolating F by $\widehat{F}$, we denote $\widehat{F^{\Delta T}}$ the piecewise linear function that interpolates the values of $F^{\Delta T}$. Then the equation (14) can be reformulated as

$$f_i = \widehat{F^{\Delta T}}(v_i, w_i^{avg}), \quad \forall i \in S. \tag{14'}$$

Similarly in the vertical model, the fuel consumption for a segment of ground distance L, denoted by $F^L_{i,h}$ for a segment i on an altitude h, can be exactly precomputed based on the wind triangle:

$$F^L_{i,h}(v, w) = F_h(v, w) \cdot L_i \cdot \frac{v}{\sqrt{v^2 - (U^c_h)^2} + U^t_h}, \quad \forall (v, w) \in V \times W,$$

since the aircraft speed v is given as a parameter. Interpolating values of $F^L_{i,h}$ by a piecewise linear function $\widehat{F}^L_{i,h}$ reformulates (27) as

$$f_i = \sum_{h\in H} \mu_{i,h} \cdot \widehat{F}^L_{i,h}(v_i, w_i^{avg}), \quad \forall i \in S. \tag{27'}$$

Both bivariate piecewise linear functions $\widehat{F}^{\Delta T}$ and $\widehat{F}^L$ can be illustrated in Figure 4. They can be formulated using linear constraint by either the lambda method in equations (1) or delta method in equations (2) by replacing the $\widehat{F}$. The set of triangles $K_{v,w}$ in (14') has a total number of $|K_{v,w}| = (|V|-1)\cdot(|W|-1)\cdot 2$ per segment; while the triangles $K^h_{v,w} := H \times K_{v,w}$ presented in (27') introduce one more dimension of altitude H, which sums up to a number of $|K^h_{v,w}| = |H|\cdot(|V|-1)\cdot(|W|-1)\cdot 2$ per segment. Note that the nonlinear equation (27') can be further linearized by constraining the binary variables in the piecewise linear function models, such that only the triangles of altitude h where $\mu_{i,h}$ equals one can be selected. Here we take only the lambda method of equations (1) as an example, where $K := K^h_{v,w}$, (1a) can be extended to

$$\sum_{k\in K^h_{v,w}} w_k = \mu_h, \quad \forall h \in H, \tag{1aH}$$

then f_i in (27') can be calculated as equal to the right hand side of (1e).

7.2. Linear Approximation of the Second Order Cone

Another difficulty in our MINLP models is the quadratic constraints for speed decomposition (10) and (20), and equation of motion (18). The equality (10) can be transformed into an equivalent inequality:

$$v_i^2 \geq (v_i^x)^2 + (v^y)_i^2, \quad \forall i \in S. \tag{10'}$$

for aircraft speed $v_i \geq 0.46$ Mach number. In such case, as observed in Figure 4, the fuel consumption function $\widehat{F}^{\Delta t}$ in (14') is a monotonically increasing function with respect to air speed, regardless of the value of weight. Therefore, the objective function (4) minimizing fuel consumption will guarantee the variable v_i taking its minimum value given by (10').

Similarly, we can also conify (20) in the vertical model as follows:

$$(v_i^{air})^2 \geq (v_i^t)^2 + (u_i^c)^2 \qquad \forall i \in S. \tag{20'}$$

The inequality (20') is equivalent to the equality (20), only if air speed v_i^{air} always takes its minimum possible value. As observed in Figure 4, the function F^L_h

is a monotonically increasing function with respect to v_i^{air} from its fuel-minimum speed at each weight level. In fact, if there is no time constraint as in (16), the unconstrained vertical profile can be easily computed by a *backward dynamic programming* approach introduced in Section 7.3. If the time constraint requires the aircraft to speed up from its unconstrained optimal speed, (20') and (20) are equivalent. Under such speeding-up scenarios, equation (18) can also be conified as:

$$L_i \leq v_i^{ground} \cdot \Delta t_i \qquad \forall i \in S$$

since neither increasing the speed nor increasing the travel time will lead to fuel saving. Applying the variable transformations

$$a_i = \frac{1}{2}(v_i^{ground} - \Delta t_i), \quad c = \frac{1}{2}(v_i^{ground} + \Delta t_i), \quad b_i = \sqrt{L_i}, \quad \forall i \in S, \text{ yields}$$
$$c_i^2 \geq a_i^2 + b_i^2, \tag{18'}$$

which is a standard form of second order cone. Equations (10'), (20'), and (18') can then be linearly approximated as outlined in equations (3).

7.3. Fuel-optimal Vertical Profile without Time Constraint

A minimum-fuel flight without time constraint can be easily found by a backward dynamic programming (BDP) approach. Given the dry weight W^{dry} as the end weight of the last segment, our BDP approach iteratively assigns a speed and an altitude to a segment, starting from the last segment to the first. Recall that the segment fuel consumption function $\widehat{F}_h^L(v, w)$ is a piecewise linear function based on measurements at each discrete level of speed and weight. For each segment, given its end weight, a fuel-minimum speed and altitude can be selected by enumerating all discrete levels of speed and altitude, since the minimum of a piecewise linear function is at its extreme points. (Each segment fuel estimation by average weight requires two iterations: a first iteration estimate the segment fuel consumption using the given end weight, and then the average weight calculated as half of the estimated fuel plus the end weight is used for fuel estimation in the second iteration. Our numerical experiments confirmed that such two-iteration fuel estimation differ from the average weight fuel estimation by a relative error under 0.0002%.) Then we compute the minimum segment fuel consumption using the selected speed and altitude, determine the starting weight of the incumbent segment which is also the end weight of the previous segment, and proceed backwards to the previous segment, until the first segment. The Bellman's principle of optimality holds, since less fuel is consumed on a segment with smaller end weight. The BDP approach takes only $O(|S| \cdot |V| \cdot |H|)$ time complexity, and runs within centiseconds. However, if a time constraint is imposed so as to avoid delays [24], BDP cannot guarantee

optimality anymore. In particular, this study of vertical flight planning focuses on time constraints that require speeding up the aircraft from its unconstrained optimal travel time computed by BDP.

8. Experimental Setup

All experiments were run using the open-source solver SCIP 3.1.0 [29], which can solve both MINLP and MILP models. Each experiment was performed on a computing node with 12-core Intel Xeon X5675 at 3.07 GHz and 48 GB RAM. Both the sequential version and the parallel version of SCIP were used. The parallel run of SCIP was performed with 12 threads with shared memory using FiberSCIP provided by UG 0.7.3 [30].

Two of the most common aircraft types are used for our empirical study: Airbus 320 (A320) and Boeing 737 (B737). The data are provided by Lufthansa Systems AG. The maximum flight range for both aircraft types are considered in our work, i.e., 3000 nautical miles (NM) for A320, which amounts to the distance from London to New York, or from London to Dubai; or 1500 NM for B737, which is around the distance from Los Angeles to Chicago, or from Hong Kong to Tokyo; The minimum flight range considered for both aircrafts is 750 nautical miles (NM), which is around the distance from Frankfurt to Madrid, or from Boston to Chicago.

For each flight range, the departure and destination point can be projected on the equator so that the great circle connecting the two points corresponds to the equator. Additionally, the solution space for the horizontal trajectory is set to 5 latitude degrees (600 NM) to the north as well as to the south. Full information of wind is used in the solution space, i.e., 1.25×1.25 degrees or ca. 75×75 NM. This results in a wind grid of size 10×8 for a 750 NM trip, and 40×8 for a 3000 NM trip. The number of wind triangles is twice the size of the grid.

The wind data is issued every six hours. Each issue contains wind data every 3 hours in the next 36 hours, and for 13 different altitudes. It is possible that these altitudes in the wind data do not include some of the admissible flight altitudes. In such case, the wind information for each admissible flight altitudes needs to be interpolated linearly by two closest wind altitudes. For the time dimension, the passing time for each wind grid point in the horizontal phase or each segment in the vertical phase is first estimated based on their distance to the departure and destination. The wind data for each grid point or segment is linearly interpolated by the two adjacent time points. An initial vertical optimization is performed to determine the altitude, from which the wind data of each wind grid point is taken for the horizontal optimization, as is also done in [20]. We first divide a great circle trip between the departure and destination into equidistance segments and perform one run of BDP (see Section 7.3) without wind to estimate a vertical profile by

Table 1: Two aircraft types A320 and B737 with forward trip (westwards) and return trip (eastwards, `-r`) with normal wind and ten-times stronger wind (`-w`). Listed are number of segments ($|S|$), altitudes ($|H|$), speed levels ($|V|$), weight levels ($|W|$), minimum and maximum speed (V^{min} and V^{max} in Mach number), dry weight and maximum weight (W^{dry} and W^{max} in kg), and conic approximation level J.

Instance	$\|S\|$	$\|H\|$	V^{min}	V^{max}	W^{dry}	W^{max}	Range	$\|V\|$	$\|W\|$	J
`A320`	10 – 40	5	0.71	0.82	56614	76990	3000	12	15	10
`A320r`	10 – 40	6	0.74	0.82	56614	76990	3000	9	15	10
`A320w`	10 – 40	5	0.64	0.82	56614	76990	3000	16	15	10
`A320rw`	10 – 40	6	0.66	0.82	56614	76990	3000	15	15	10
`B737`	10 – 20	4	0.70	0.76	43190	54000	1500	7	12	10
`B737r`	10 – 20	5	0.70	0.76	43190	54000	1500	7	12	10
`B737w`	10 – 20	4	0.70	0.76	43190	54000	1500	7	12	10
`B737rw`	10 – 20	5	0.69	0.76	43190	54000	1500	8	12	10

time. Using the estimated passing time of each wind grid point above, an altitude is assigned based on the estimated vertical profile.

Our numerical experiments for fuel estimation accuracy test confirmed that when dividing a longest possible 3000 NM trip into equidistance segments of 100 NM, the total fuel estimation error is under 1 kg ($< 0.005\%$ relative error). With the same 1 kg error threshold, we experimentally determine a second-order cone approximation level $J = 10$ for both aircrafts.

The segment fuel functions $F^{\Delta t}$ and F^{L} take weight and speed as parameters for the piecewise linear interpolation. The dry weight W^{dry} of the aircraft is determined by summing up the weight of the empty aircraft (with contingency fuel for safety) and the weight of usual passenger and luggage load; The maximum aircraft weight W^{max} follows the aircraft data. Then we can safely exclude the weight levels that are not within or adjacent to the range $[W^{dry}, W^{max}]$. This reduces the number of weight levels for A320 from originally 29 to 15, and for B737 from 31 to 12. The speed levels can be reduced similarly. We can compute the unconstrained fuel-optimal speed for each wind grid point on each admissible altitude with each weight level by enumerating all speed levels in the preprocessing. The minimum speed V^{min} takes value of the minimum of all the unconstrained optimal speeds. Since only speeding-up of an unconstrained optimal flight is studied in our work, only the speeds ranging from V^{min} to the maximum possible speed V^{max} are considered. Note that different wind situations may lead to different speed ranges. As shown in Table 1, the originally given 27 speed levels for A320 are reduced to 9 to 16, and the 10 speed levels for B737 are reduced to 7 or 8.

9. Experiment Results

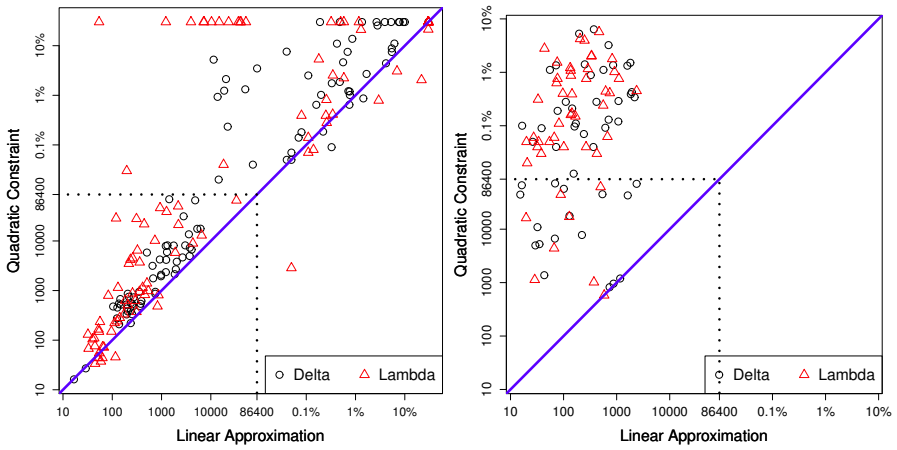
Eight approaches from a full factorial design of three computational factors were considered: (1) the use of quadratic constraints or linear approximation for the second-order cone (SOC) constraints; (2) delta or lambda method for the 2D piecewise linear interpolation; (3) sequential or parallel computing with a 12-core CPU. We compare the computational performance of each approach in terms of computation time. The maximum computation time for each run is set to 24 hours. If an instance is not solved to optimality after it was cut at 24 hours, its gap between the best known integer solution and the lower bound is also compared.

9.1. Linear Approximation vs. Quadratically Constrained Model

The advantage of using linear approximation reformulation of the SOC constraints can be clearly visualized in Figure 5. In the horizontal planning phase with one set of SOC constraints, the linear approximation model outperforms the quadratically constrained model except few outliers with the lambda formulation, as shown in Figure 5a. The runtime development plot in Figure 5c also confirms that using the linear approximation model with either the delta or the lambda formulation performs better than its counterpart using the quadratically constrained model. In the vertical planning phase where two sets of quadratic cone constraints exist, the advantage of using the linear approximation model becomes dominant, as shown in Figure 5b and 5d. Here, only the smallest instances with 10 segments are compared, since no feasible solution can be found by the quadratically constrained model for most of the larger instances. As typically most of the instances are solved by the linear approximation model within 1,000 seconds, almost none of the instances can be solved by the quadratically constrained model within 1,000 seconds, and most of the instances cannot be solved to optimality even in 24 hours. The comparison of the two piecewise linear function formulations is less conclusive. In the horizontal model, the lambda method appears to perform better for around 80% of the smaller instances that can be optimally solved within 24 hours; and the delta method stands out for the rest of the larger instances that have not reached optimality. In the vertical model, the lambda method appears to perform better than the delta method in the linear approximation, while the delta formulation appears to be a better alternative over the worse performing quadratic model.

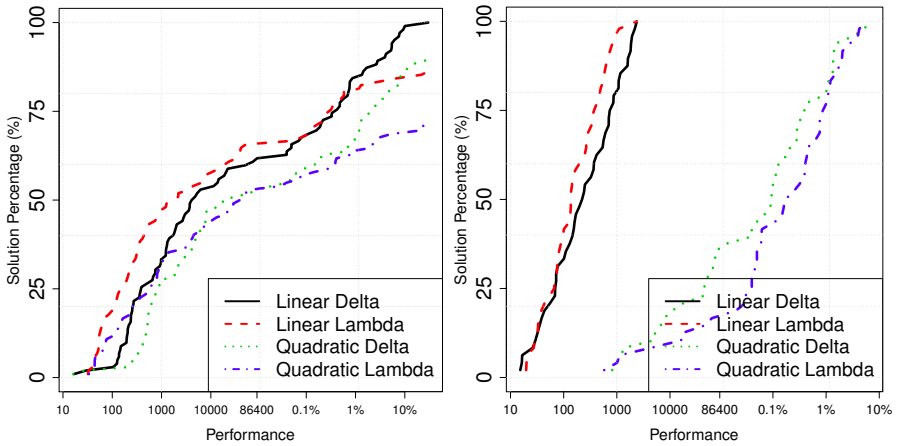
9.2. Further Computational Studies in Linear Approximation Model

We further focus on the better performing linear approximation model in this section, so that all the instances in the vertical flight planning can be included in this study. A more in-depth computational analysis of the delta and lambda formulation as well as the use of parallel and single thread is given here. The performance distribution is shown in Figure 6, and the runtime development of the



(a) Linear vs. Quad. Horizontal

(b) Linear vs. Quad. Vertical

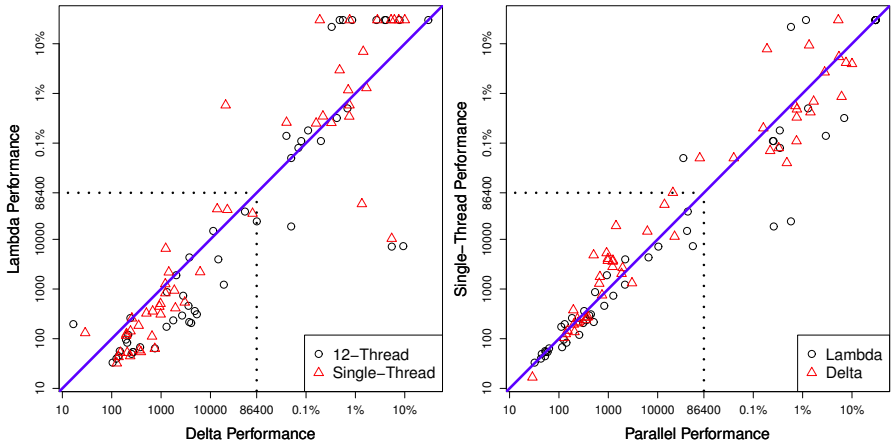


(c) Runtime Development Horizontal

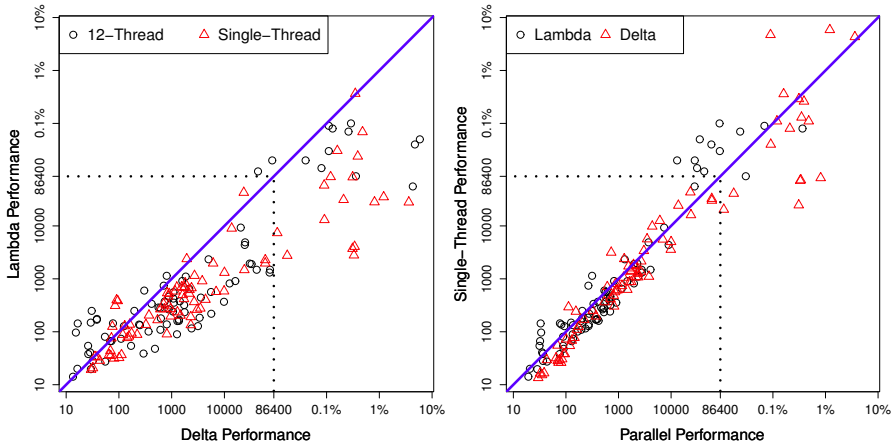
(d) Runtime Development Vertical

Figure 5: The comparison of the linear approximation model and the quadratically constrained model in both horizontal (left) and vertical (right) flight planning. The performance is measured in computation time or the gap after cutoff at 24 hours.

four alternatives is shown in Figure 7. In the horizontal phase, as shown in Figure 6a, the lambda formulation performs better for smaller instances that can be solved within 24 hours; however, for the instances that are not solved to optimality within 24 hours, the delta formulation usually leaves a smaller gap than the lambda formulation. Figure 6b further shows that for the large instances of the horizontal planning problem, using a single-thread is a computationally better choice than using a parallel solver. This shows that using multiple threads under shared memory may not always pay off, especially for some memory critical computations such as the large horizontal instances in our case. These observations in the horizontal planning are also presented in the runtime development plot in Figure 7a. The



(a) Delta vs. Lambda Horizontal (b) Parallel vs. Single-Thread Horizontal

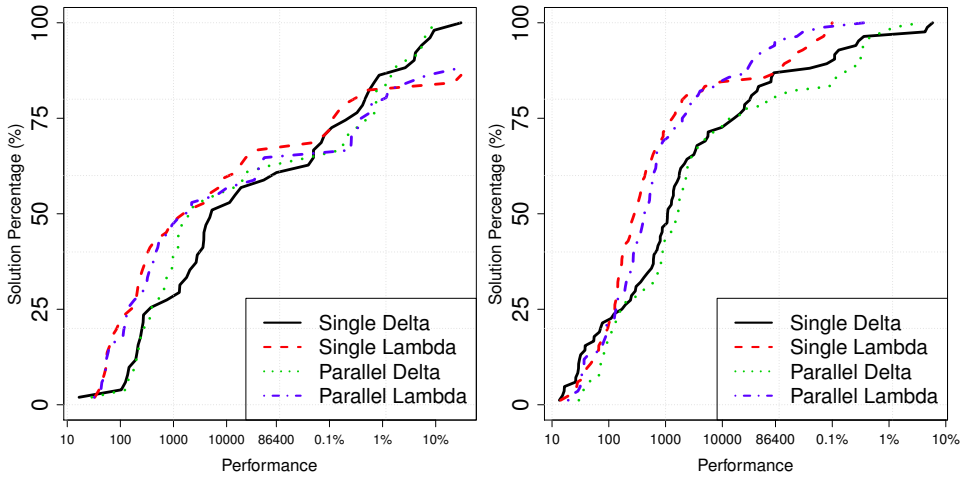


(c) Delta vs. Lambda Vertical (d) Parallel vs. Single-Thread Vertical

Figure 6: The comparison in the linear approximation model between delta and lambda method (left) or between parallel-thread and single-thread (right) in horizontal (above) or vertical (below) flight planning. The performance is measured in computation time or the gap after cutoff at 24 hours.

single-thread lambda method appears to be the best choice for smaller instances, and it solves the most instances to optimality within 24 hours, around two thirds. The delta method stands out for 20% of the largest instances, and its parallel version finds feasible solution within 5% gap for all test instances. On the other hand, the lambda counterpart cannot find feasible solution within 24 hours for around 15% of the instances.

Some different performance patterns are observed in the vertical flight planning. Figure 6c shows that the lambda formulation clearly outperforms the delta formulation, especially for the large instances. As can be also observed from the runtime development plot in Figure 7b, the two lambda alternatives clearly out-



(a) Runtime Development Horizontal

(b) Runtime Development Vertical

Figure 7: The comparison in the linear approximation model of using delta or lambda method and using parallel or single thread in the horizontal (a) and vertical (b) flight planning. The performance is measured in computation time required for finding optimum or the gap after cutoff at 24 hours.

performs the delta counterpart. The superior performance of lambda method over delta method is a bit surprising, since the delta method was proved to be the theoretically better alternative [31]. The comparison of parallel and sequential computation in Figure 6d shows an interesting correlation: while using single thread performs slightly better in most of the small instances, using multiple threads gives an edge to the better performing lambda formulation for the large instances. The multi-thread lambda method is overall the best alternative for vertical planning instances, especially for the large instances, while single-thread lambda method performs slightly better for smaller instances that are solved within 10,000 seconds, as shown in Figure 7b.

10. Conclusions and Outlooks

In this work, we proposed mathematical models for the horizontal and vertical flight planning problem under free-flight conditions. We identified the second-order cone constraints from the original mixed integer nonlinear programming (MINLP) models, and used linear approximation techniques from literature to transform them into mixed integer linear programming model. Numerous computational experiments extracted from real-world weather data and aircraft performance data for Airbus 320 and Boeing 737 were performed. The linear approximation approach was experimentally validated to outperform the original

MINLP model, especially for the vertical planning problem where more second-order cone constraints exist. We also studied two different approaches of handling two-dimensional piecewise linear functions, namely, the lambda and the delta method, as well as explored the potential of using parallel computing with the shared-memory multi-core CPU. However, no clear winner can be nominated, their relative performance is problem dependent. The lambda method appears to perform better in the vertical planning, and in the small instances of the horizontal problem, while the delta method performs better for large horizontal instances. Parallel computation does not show a significant advantage over the single-thread approach in most of the scenarios, however, the parallel lambda approach appears to be the best strategy for the vertical planning problem.

Although the current MILP approach still requires undesirably long computation time, the optimal solutions and lower bounds obtained in this work may be used to assess the quality of our future heuristic approaches, which may provide a feasible initial solution to the MILP solver to further speed up the solution process.

Acknowledgements This work is supported by BMBF Verbundprojekt E-Motion.

[1] Airbus, Future Journeys 2013-2032, Global Market Forecast, 2013.

[2] Final report of the RTCA Task Force 3, Free Flight Implementation, RTCA, Incorporated, 1995.

[3] EUROCONTROL, European Route Network Improvement Plan Part 2: European ATS Route Network – Version 2014-2018/19, Section 7.2, June 2014.

[4] S. Altus, Flight planning – the forgotten field in airline operations, `http://www.agifors.org/studygrp/opsctl/2007/`, presented at AGIFORS Airline Operations 2007.

[5] S. Burer, A. Saxena, The MILP road to MIQCP, in: Mixed Integer Nonlinear Programming, Springer, 2012, pp. 373–405.

[6] M. S. Lobo, L. Vandenberghe, S. Boyd, H. Lebret, Applications of second-order cone programming, Linear Algebra and its Applications 284 (1998) 193–228.

[7] A. Ben-Tal, A. Nemirovski, On Polyhedral Approximations of the Second-Order Cone, Mathematics of Operations Research 26 (2) (2001) 193 – 205.

[8] F. Glineur, Computational Experiments with a Linear Approximation of Second-Order Cone Optimization, Tech. rep., Image Technical Report 0001, Faculté Polytechnique de Mons, Belgium (2000).

[9] J. P. Vielma, S. Ahmed, G. L. Nemhauser, A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs, INFORMS Journal on Computing 20 (3) (2008) 438–450.

[10] A. Fügenschuh, H. Homfeld, H. Schülldorf, Single-car routing in rail freight transport, Transportation Science 49 (1) (2013) 130–148.

[11] A. Fügenschuh, K. Junosza-Szaniawski, Z. Lonc, Exact and Approximation Algorithms for a Soft Rectangle Packing Problem, Optimization 63 (11) (2014) 1637–1663.

[12] J. P. Vielma, S. Ahmed, G. Nemhauser, Mixed-integer models for nonseparable piecewise-linear optimization: unifying framework and extensions, Operations research 58 (2) (2010) 303–315.

[13] A. Martin, M. Möller, S. Moritz, Mixed integer models for the stationary case of gas network optimization, Mathematical programming 105 (2-3) (2006) 563–582.

[14] A. Fügenschuh, C. Hayn, D. Michaels, Mixed-Integer Linear Methods for Layout-Optimization of Screening Systems in Recovered Paper Production, Optimization and Engineering 15 (2) (2014) 533–573.

[15] International Virtual Aviation Organization, IFR cruise altitude or flight level, `http://ivao.aero/training/documentation/books/SPP_ADC_IFR_Cruise_Altitude.pdf`.

[16] A. Peter, Ein MILP, ein MINLP und ein graphentheoretischer Ansatz für die Free-Flight Optimierung. Diplomarbeit, Technische Universität Darmstadt (2007).

[17] N. K. Wickramasinghe, A. Harada, Y. Miyazawa, Flight trajectory optimization for an efficient air transportation system, in: 28th International congress of the aeronautical sciences, 2012, p. 12 pages.

[18] R. Howe-Veenstra, Commercial aircraft trajectory optimization and efficiency of air traffic control procedures, Master's thesis, University of Minesota (2011).

[19] O. Richard, S. Constans, R. Fondacci, Computing 4D near-optimal trajectories for dynamic air traffic flow management with column generation and branch-and-price, Transportation Planning and Technology 34 (5) (2011) 389–411.

[20] H. K. Ng, B. Sridhar, S. Grabbe, Optimizing aircraft trajectories with multiple cruise altitudes in the presence of winds, Journal of Aerospace Information Systems 11 (1) (2014) 35–47.

[21] Z. Yuan, A. Fügenschuh, A. Kaier, S. Schlobach, Variable Speed in Vertical Flight Planning, in: Operations Research Proceedings, Springer, 2014, p. 6 pages, to appear.

[22] S. Liden, Optimum cruise profiles in the presence of winds, in: Proceedings of IEEE/AIAA 11th Digital Avionics Systems Conference, IEEE, 1992, pp. 254–261.

[23] J. A. Lovegren, R. J. Hansman, Estimation of potential aircraft fuel burn reduction in cruise via speed and altitude optimization strategies, Tech. rep., ICAT-2011-03, MIT International Center for Air Transportation (2011).

[24] M. S. Aktürk, A. Atamtürk, S. Gürel, Aircraft rescheduling with cruise speed control, Operations Research 62 (4) (2014) 829 – 845.

[25] P. Hagelauer, F. Mora-Camino, A soft dynamic programming approach for on-line aircraft 4d-trajectory optimization, European Journal of Operational Research 107 (1998) 87–95.

[26] G. B. Dantzig, On the significance of solving linear programming problems with some integer variables, Econometrica 28 (1) (1960) 30 – 44.

[27] H. M. Markowitz, A. S. Manne, On the solution of discrete programming problems, Econometrica 25 (1) (1957) 84 – 110.

[28] D. Wilson, Polyhedral methods for piecewise-linear functions, Ph.D. thesis, University of Kentucky (1998).

[29] T. Achterberg, SCIP: Solving Constraint Integer Programs, Mathematical Programming Computation 1 (1) (2009) 1–41.

[30] Y. Shinano, T. Achterberg, T. Berthold, S. Heinz, T. Koch, ParaSCIP – a parallel extension of SCIP, in: C. Bischof, et al. (Eds.), Competence in High Performance Computing 2010, Springer, 2012, pp. 135–148.

[31] M. Padberg, Approximating separable nonlinear functions via mixed zero-one programs, Operations Research Letters 27 (1) (2000) 1–5.