algorithms

MDPI

Article

A Discrete-Continuous Algorithm for Free Flight Planning

Ralf Borndörfer †, **Fabian Danecker** *,† and **Martin Weiser** †

Zuse Institute Berlin, Takustraße 7, 14195 Berlin, Germany; borndoerfer@zib.de (R.B.); weiser@zib.de (M.W.)
* Correspondence: danecker@zib.de
† These authors contributed equally to this work.

Abstract: We propose a hybrid discrete-continuous algorithm for flight planning in free flight airspaces. In a first step, our discrete-continuous optimization for enhanced resolution (DisCOptER) method computes a globally optimal approximate flight path on a discretization of the problem using the A^* method. This route initializes a Newton method that converges rapidly to the smooth optimum in a second step. The correctness, accuracy, and complexity of the method are governed by the choice of the crossover point that determines the coarseness of the discretization. We analyze the optimal choice of the crossover point and demonstrate the asymtotic superority of DisCOptER over a purely discrete approach.

Keywords: shortest path; flight planning; free flight; discrete-continuous algorithm; optimal control; discrete optimization

MSC: 90C35; 49M37; 65K10; 65L10; 90C27

1. Introduction

Flight planning is concerned with the computation of time and fuel efficient flight paths with respect to the weather, see [1] for a comprehensive survey. In particular, wind conditions make a big difference: flying with a headwind of 60 kts increases flight time and fuel consumption of an Airbus A321 by as much as 20% over a tailwind of 60 kts [2]. To exploit this potential, and to mitigate airspace congestion, free flight aircraft routing has been suggested since 1995 [3], and projects to complement, enhance, and finally replace the airway network that is currently used to organize all air traffic are now under way all over the world. Europe is introducing so-called free route airspaces (FRAs), in which one can fly on arbitrary straight lines between defined entry and exit points, and between more and more intermediate points, moving ever closer towards free flight. According to EUROCONTROL, FRA projects are now in place in three quarters of all European airspaces, and, once fully implemented, will save total fuel burn, CO_2, and H_2O emissions by 1.6–2.3%, which amounts to 3000 tonnes of fuel/day, 10,000 tonnes of CO_2/day, €3 million in fuel costs/day, and 500,000 nautical miles/day [4].

The flight planning problem can be seen as a special type of time-dependent shortest path problem. A large number of algorithms has been developed in this general context, including contraction hierarchies, hub labeling, and arc flags for route planing in road networks, see [5,6] for surveys, RAPTOR, transfer patterns, and connection scan for journey planning in public transport networks [6], the isochrones method and dynamic programming for ship weather routing [7], and sampling-based algorithms like rapidly exploring random graphs and trees (RRTs), probabilistic road maps (PRMs), artificial potential fields, as well as graph-based algorithms such as A^*, D^*, theta*, etc. for robot path planning [8]. The variety of these methods reflects the different characteristics of the respective problems.

The best flight planning methods are currently super-fast A^*/Dijkstra algorithms that employ efficient problem specific speed-up techniques such as cost projection [9], super-optimal wind [10], and active constraint propagation [11]. They can find globally

Citation: Borndiörfer, R.; Danecker, F.; Weiser, M. A Discrete-Continuous Algorithm for Free Flight Planning. *Algorithms* **2021**, *14*, 4. https://dx.doi.org/10.3390/a14010004

Received: 30 November 2020
Accepted: 20 December 2020
Published: 25 December 2020

optimal solutions of basic problem variants on the world wide airway network with its 100,000 nodes and 600,000–700,000 edges within milliseconds. A^* methods can in principle be extended to deal with FRAs or free flight by excessive graph augmentation, but only up to a certain point, when the graphs become too large and dense.

On the other hand, numerical methods of optimal control are able to compute optimal free flight trajectories to high precision with great efficiency, either with indirect methods based on Pontryagin's maximum principle [12–15], or by direct methods using a collocation discretization to reformulate the problem as a nonlinear program (NLP) [16,17]. These methods compute a smooth trajectory independently of any a priori network discretization, i.e., the desired free flight path. The computational complexity of solving the optimality systems by Newton's method is asymptotically much smaller than for graph discretizations. If measured in terms of accuracy, higher order discretizations of the underlying ordinary differential equations exhibit even larger asymptotic gains. The drawback is that continuous optimal control methods converge only locally, and towards any local optimum, without providing any guarantee of global optimality. Approaches to compute globally optimal solutions to optimal control problems include global optimal control [18], mixed-integer optimal control [19], and various heuristics [20]. Applications to flight planning exist, but consider only very small networks [21] or vertical profiles [22].

We propose in this paper the novel hybrid algorithm discrete-continuous optimization for enhanced resolution (DisCOptER) that combines the strengths of discrete and continuous approaches to flight planning, and provide a numerical study of its efficiency and accuracy. The discrete component of our method provides global optimality, the continuous component high accuracy and asymptotic efficiency. The idea of the method is to do a discrete search for a global optimum on a coarse, approximate, artificial network, and to use the resulting approximate solution to initialize a Newton method for the solution of a continuous optimal control problem. The correctness and effectiveness of this strategy depends on the choice of the crossover point between the discrete and the continuous part of the algorithm. The network for the discrete part must be coarse enough to be searched efficiently, and fine enough to guarantee sufficient proximity to the continuous optimum, such that a subsequent Newton iteration will converge to the latter. Clearly, the convergence radius of the continuous method strongly depends on the gradients of the wind field and affects, together with the approximation error associated with the graph, the computational complexity of both methods. We shall show that this idea is ideally suited for free flight settings.

Our aim in this paper is to demonstrate the potential of combining discrete and continuous optimization methods for the solution of problems that involve space or time discretizations. The goal is to achieve global optimality and rapid convergence at the same time. The model that we consider is simple, and the special characteristics of flight planning have left their imprint. The basic idea, however, should, with suitable modifications, be applicable to various problems of similar nature. In this vein, our paper intends to give a first indication of the usefulness of such approaches.

The paper is structured as follows. Sections 2.1–2.3 describe the free flight problem that we consider. The DisCOptER algorithm is introduced in Section 2.4 including error and complexity estimates. Finally, Section 3 provides a computational study and analyzes the choice of the switch over point.

2. Materials and Methods

2.1. Free Flight Planning

We consider in this paper an idealized version of the flight planning problem in 2D Euclidean space subject to a stationary wind field. We want to compute a flight path $x : [0, T] \to \mathbb{R}^2, \tau \mapsto x(\tau)$ that connects an origin and a destination $x_O, x_D \in \mathbb{R}^2$; the path parameter τ measures the flight time. The path is influenced by a smooth field of stationary wind $w : \mathbb{R}^2 \to \mathbb{R}^2$ of bounded magnitude $\|w\| \leq \bar{w}$. Flying at an airspeed $v : [0, T] \to \mathbb{R}^2, \tau \mapsto v(\tau)$ of constant magnitude $\|v\| = \bar{v} > \bar{w}$, the aircraft arrives at time

$T \in \mathbb{R}_{\geq 0}$, which we seek to minimize. Our setting is chosen for ease of exposition, but our method carries over to more complex 3D and/or time dependent versions, or other objectives, in particular, minimization of fuel consumption.

2.2. Continuous Approach: Optimal Control

In free flight, the flight path is not restricted to a predefined airway network of waypoints and segments. Instead, any Lipschitz-continuous path $x : [0, T] \to \mathbb{R}^2$, with $\|x_t - w\| = \bar{v}$ almost everywhere, connecting origin x_O and destination x_D, is a valid trajectory. Among those, we shall find one of minimal flight duration T. This classic of optimal control is known as Zermelo's navigation problem [23].

In order to formulate the problem over a fixed interval $[0, 1]$ independent of the actual flight duration, we scale time by T^{-1} as usual in free end time problems and arrive at the following optimal control problem for the flight duration $T \in \mathbb{R}$, the flight path $x \in H^1([0, 1])^2$, and the airspeed $v \in L^2([0, 1])^2$:

$$\min_{T,x,v} T \quad \text{s.t.} \quad c(T, x, v) = \begin{bmatrix} x(0) - x_O \\ x(1) - x_D \\ \dot{x}(\tau) - T\,(v(\tau) + w(x(\tau))) \\ v(\tau)^T v(\tau) - \bar{v}^2 \end{bmatrix} = 0\,. \tag{1}$$

Here, the constraint $c : Z \to \Lambda$ maps from the primal domain $Z := \mathbb{R} \times H^1([0, 1])^2 \times L^2([0, 1])^2$ to the image space $\Lambda := \mathbb{R}^2 \times \mathbb{R}^2 \times L^2([0, 1])^2 \times L^2([0, 1])$.

2.2.1. Optimality Conditions

Let us briefly recall the necessary and sufficient optimality conditions for the optimal control problem (1). With $z = (T, x, v) \in Z$ and $\lambda \in \Lambda$, the Lagrangian is defined as

$$L(z, \lambda) = T - \langle \lambda, c(z) \rangle_{\Lambda,\Lambda}. \tag{2}$$

If z is a (local) minimizer, the necessary first order or Karush–Kuhn–Tucker (KKT) condition

$$L'(z, \lambda) = 0 \tag{3}$$

and the necessary second order condition

$$\langle L_{zz}\zeta, \zeta \rangle_{Z^*,Z} \geq 0 \quad \forall \zeta \in \ker c'(z)$$

hold [24], since $c'(z)$ is surjective due to $\|w\|_{L^\infty([0,1])} < \bar{v}$. Moreover, if the KKT conditions (3) are satisfied at z for some λ and the sufficient second order condition (SSC)

$$\langle L_{zz}\zeta, \zeta \rangle_{Z^*,Z} \geq \alpha \|z\|_Z^2 \quad \forall \zeta \in \ker c'(z) \tag{4}$$

holds for some $\alpha > 0$, then z is a locally unique solution of (1) and stable under perturbations of the problem, e.g., due to sufficiently fine discretization. Let us point out that for wind fields with non-vanishing second derivative, in general there is no closed form solution of the necessary conditions (3), such that solutions must be approximated numerically.

Approaches to computing solutions of (1) generally fall into two classes: indirect methods relying on Pontryagin's maximum principle [15,25] and direct methods based on discretization of the minimization problem (1) [26,27]. Indirect methods lead to a boundary value problem for state x and adjoint state λ together with a pointwise optimality condition for the control v, which can be solved by shooting type methods, collocation, or spectral discretization approaches [28,29]. The discretization of direct methods, usually by collocation or spectral methods, translates the optimal control problem in a finite dimensional nonlinear program to be solved by corresponding optimization problems [16,17,30]. While indirect methods using multiple shooting lead to smaller problem sizes than direct methods based on collocation in particular for high accuracy requirements, the latter

are widely seen as being easier to implement and use, in particular in the presence of state constraints.

In both approaches, Newton-type methods for solving either discretized boundary value problems or nonlinear programming problems converge in general only locally towards a close-by minimizer. Thus, sufficiently good initial iterates need to be provided. The domain of convergence can be extended using line search methods or trust region methods, but without guarantee of global optimality. Special care has to be taken in the case of non-convex problems, since the Newton direction need not be a descent direction if far away from a minimizer satisfying the sufficient second order conditions. Convexification, truncation of iterative solvers [31–33], or solvers for non-convex quadratic programs can be used to address this.

2.2.2. Collocation Discretization

Exemplarily, we consider a discretize-then-optimize approach based on direct collocation with the midpoint rule. Let $0 = \tau_0 < \cdots < \tau_n = 1$ be a time grid, $X_h = \{x \in H^1([0,1]) \mid x|_{[\tau_i,\tau_{i+1}]} \in \mathbb{P}_1, i = 0,\dots,n-1\}$ and $V_h = \{v \in L^2([0,1]) \mid v|_{]\tau_i,\tau_{i+1}[} \in \mathbb{P}_0, i = 0,\dots,n-1\}$ the piecewise linear and piecewise constant ansatz spaces for positions and velocities, respectively. We discretize (1) by looking for solutions $x_h \in X_h^2$ and $v_h \in V_h^2$ and require the state equation $\dot{x}(\tau) - T(v(\tau) + w(x(\tau))) = 0$ to be satisfied only at the interval midpoints $\tau_{i+1/2} := (\tau_i + \tau_{i+1})/2$ for $i = 0,\dots,n-1$. Representing x_h by its nodal values $x_i = x_h(\tau_i)$ and v_h by its midpoint values $v_i = v_h(\tau_{i+1/2})$, we obtain the large nonlinear program

$$\min_{T,x_h,v_h} T \quad \text{s.t.} \quad c_h(T,x_h,v_h) = \begin{bmatrix} x_0 - x_O \\ x_n - x_D \\ x_1 - x_0 - (\tau_1 - \tau_0)T\,(v_0 + w((x_0 + x_1)/2)) \\ \vdots \\ x_n - x_{n-1} - (\tau_n - \tau_{n-1})T\,(v_{n-1} + w((x_{n-1} + x_n)/2)) \\ v_0^T v_0 - \bar{v}^2 \\ \vdots \\ v_{n-1}^T v_{n-1} - \bar{v}^2 \end{bmatrix} = 0. \tag{5}$$

2.2.3. Discretization Error

The discretization error introduced by the midpoint rule is well-known to be of second order. For given $w \in C^1([0,1])^2$ and $T > 0$ there is some constant $C(T, \|w_x\|_{L^\infty(\mathbb{R}^2)})$ independent of v such that

$$\|x_h - x\|_{L^\infty([0,1])} \le C\delta\tau^2, \tag{6}$$

where $\delta\tau = \max_{i=0,\dots,n-1} \tau_{i+1} - \tau_i$ is the mesh width, see, e.g., [34]. This translates into a corresponding error in the objective, i.e., the flight time T. Different goal-oriented error concepts have been investigated [35,36] for equality constrained optimal control problems. The excess in actual flight time when the computed path is implemented depends quadratically on the path deviation $\|x_h - x\|_{L^\infty([0,1])}$, and is therefore of order $\mathcal{O}(\delta\tau^4)$. In any case, the error depends directly on the mesh width $\delta\tau$, which we therefore use as a coarse but simple qualitative measure of accuracy.

2.2.4. Newton-KKT Solver

Analogous to the continuous Lagrangian (2), we may formulate its discretized counterpart

$$L_h(z_h, \lambda_h) = T - \lambda_h^T c_h(z_h)$$

and the corresponding necessary first order optimality condition

$$L_h'(z_h, \lambda_h) = 0\,. \tag{7}$$

Writing $\chi = [z_h, \lambda_h]^T$, this can be solved using Newton's method by computing

$$L_h''(\chi_k)\delta\chi_k = -L_h'(\chi_k), \quad \chi_{k+1} = \chi_k + \delta\chi_k. \tag{8}$$

For smooth wind fields, there is some $\omega < \infty$ related to an affine invariant Lipschitz constant of L_h'', such that

$$\|\chi_{k+1} - \chi_*\| \leq \omega\|\chi_k - \chi_*\|^2 \tag{9}$$

holds [37]. Thus, Newton's method converges quadratically if started sufficiently close to a locally unique solution point χ_*, i.e., if $\|\chi_0 - \chi_*\| < \omega^{-1}$. This convergence radius is in general mesh-independent [37,38], and does not depend on the final accuracy in terms of mesh width $\delta\tau$, but only on problem parameters such as derivatives of the wind w.

2.2.5. Time Complexity

The run time of the Newton-KKT solver is determined by the number of steps and the cost of each step. The computational effort of a Newton step is dominated by the cost of solving the linear equation (8). Due to the ODE structure, $L_h''(\chi)$ is an arrow-shaped matrix with band width independent of $\delta\tau$. Assuming quasi-uniform meshes, i.e., there is a generic constant $C > 0$ such that $\tau_{i+1} - \tau_i \geq C\delta\tau$, this structure allows for an efficient solution in $\mathcal{O}(\delta\tau^{-1})$ time using direct band solvers.

Starting sufficiently close to the solution, say $\omega\|\chi_0 - \chi_*\| < 1$, allows to bound the truncation error by linear convergence as

$$\|\chi_k - \chi_*\| \leq (\omega\|\chi_0 - \chi_*\|)^k\|\chi_0 - \chi_*\|,$$

which is of course rather pessimistic. A tolerance of $\|\chi_k - \chi_*\| \leq \mathcal{O}(\delta\tau^2)$ to match the discretization error is therefore reached after at most $\mathcal{O}(\log\delta\tau / \log(\omega\|\chi_0 - \chi_*\|))$ iterations. Thus, the overall complexity in terms of mesh width $\delta\tau$ is

$$R_C = \mathcal{O}\left(\delta\tau^{-1}\frac{\log\delta\tau}{\log(\omega\|\chi_0 - \chi_*\|)}\right). \tag{10}$$

Remark 1. *If an inexact Newton method based on a geometrically refined sequence of meshes with mesh width $\delta\tau_k = \beta^k l$ for some $l \gg \delta\tau$ and $\beta < 1$ is used, the number of iterations is determined by the linear convergence of the collocation discretization, while the truncation error is quickly diminished by the quadratic convergence of Newton's method [37]. Since the effort per Newton step grows geometrically to its final value, the complexity reduces to*

$$R_C = \mathcal{O}(\delta\tau^{-1}) \tag{11}$$

provided the initial error is sufficiently small, i.e., $\omega\|\chi_0 - \chi_*\| \leq \beta^2$.

2.3. Discrete Approach: Shortest Paths in Airway Networks

If flight paths are restricted to a predefined airway network of waypoints and segments, flight planning becomes a special kind of shortest path problem on a digraph. Let $V \subset \mathbb{R}^2$ be a finite set of waypoints including x_O and x_D, and $A \subset V \times V$ a set of arcs such that $G = (V, A)$ is a strongly connected directed graph. A discrete flight path is a finite sequence $(x_i)_{0\leq i\leq n}$ of waypoints with $(x_i, x_{i+1}) \in A$ for $i = 0, \ldots, n-1$, connecting $x_0 = x_O$ with $x_n = x_D$. We denote the set of all flight paths by P. The total flight duration $T(p)$ for a path $p = (x_0, \ldots, x_n) \in P$ is given in terms of the flight duration $T(e_i)$ for an arc $e_i = (x_i, x_{i+1})$ by

$$T(p) = \sum_{i=0}^{n-1} T(e_i).$$

The travel time on the arc e_i can be computed from the local ground speed

$$s = v + w = \|s\| \frac{x_{i+1} - x_i}{\|x_{i+1} - x_i\|} =: \|s\| \bar{e}_i,$$

at $x = (1-\tau)x_i + \tau x_{i+1}$ and the constant airspeed $\|v\| = \bar{v}$ by solving the quadratic equation

$$(\|s\|\bar{e}_i - w)^T (\|s\|\bar{e}_i - w) = \bar{v}^2.$$

This yields

$$\|s\| = \bar{e}_i^T w + \sqrt{(\bar{e}_i^T w)^2 + \bar{v}^2 - \|w\|^2}$$

and thus

$$T(e_i) = \|e_i\| \int_{\tau=0}^{1} \|s\|^{-1} \, \delta\tau. \tag{12}$$

The discrete optimization problem to be solved is now

$$\min_{p \in P} T(p).$$

2.3.1. Graph Construction

Discrete approaches to (flight) path planning fall into two classes. The first class are sampling-based algorithms. They construct the search graph by some kind of sampling during the execution of the shortest path algorithm, which is usually an A^*-method. This class includes rapidly exploring random trees (RRT) and graph (RRG) algorithms, that are often used in robot path planning [8]. There are versions that guarantee convergence to a global optimum with probability one [39], however, although undoubtedly often extremely efficient in practice, these methods do not provide a priori error bounds or complexity estimates.

The second class are graph-based algorithms, that take the search graph as an input. Impressive super-fast performance in practice, and also theoretically for special classes of graphs, has been achieved by making use of preprocessing as well as sophisticated data structures. However, "proving better running time bounds than those of Dijkstra's algorithm is unlikely for general graphs" [6]. In many applications, including traditional flight plannning on airway networks, the search graph is canonical. In applications involving space and time discretization, like in free flight, graph construction is a degree of freedom. Using an appropriate discretization, a priori bounds on the runtime and the accuracy of the solutions can be derived. This is the approach that we take.

We cover free flight zones with "locally densely" connected digraphs, i.e., digraphs with a certain density of vertices and arcs.

Definition 1. *A digraph $G = (V, A)$ is said to be (h, l)-dense in a convex set $\Omega \subset \mathbb{R}^2$ for some $h, l > 0$, if it satisfies the following conditions:*

1. containment: $V \subset \Omega$
2. vertex density: $\forall x \in \Omega : \exists v \in V : \|x - v\| \leq h$
3. arc density: $\forall x, y \in V, \|x - y\| \leq 2h + l : (x, y) \in A$.

We call h the vertex density and $2h + l$ the connectivity radius of an (h, l)-dense graph.

With this definition, $|V| \in \mathcal{O}(h^{-2})$ and $|E| \in \mathcal{O}((2h + l)^2 h^{-4})$ hold. Furthermore, it is easy to see that this graph structure implies that G is strongly connected and therefore P is nonempty, i.e., a path from x_O to x_D exists.

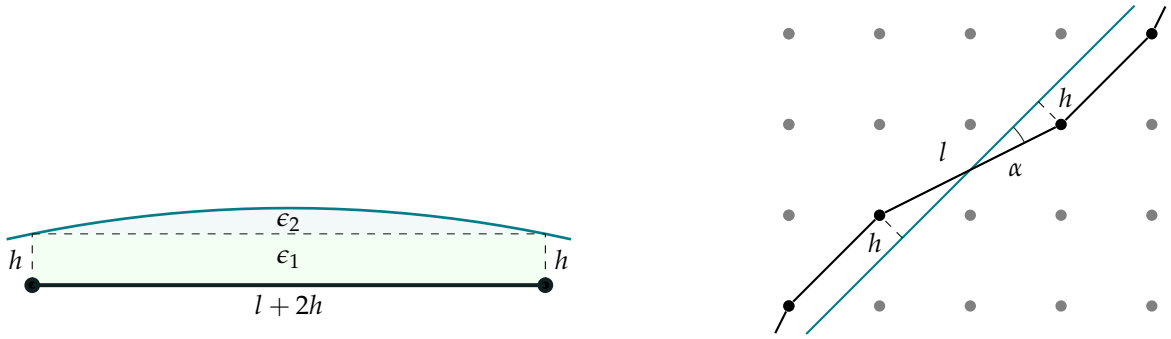
2.3.2. Discretization Error

Similar to the collocation error (6), the error due to graph discretization depends on the vertex density h and the connectivity radius $2h + l$. We restrict our exposition in this paper to a plausible argument for the error order in terms of h and l and refer the reader to [40] for rigorous error bounds of the same orders.

Spatial deviation due to vertex spacing is bounded by $\epsilon_1 = \mathcal{O}(h)$, while the linear interpolation error depending on the curvature of the continuous optimal path is bounded by $\epsilon_2 = \mathcal{O}((2h + l)^2)$, as illustrated in Figure 1a. Together, the deviation measured as pointwise normal distance is bounded by $\mathcal{O}(l^2 + h)$. As mentioned in Section 2.2 above, this translates quadratically into a flight time error of order $\mathcal{O}(l^4 + h^2)$.

Assuming that the path actually contains arcs of length around $l + 2h$, this order of error is not only an upper bound, but also the expected error. The existence of arcs of length $2h + l$ in the optimal discrete path is likely, as the following argument based on Figure 1b shows. With quasi-uniform vertex spacing, the number of adjacent vertices within a distance $2h + \hat{l} \leq 2h + l$ is of order $\mathcal{O}((2h + \hat{l})^2/h^2)$, such that the average angle between arcs of length up to $2h + \hat{l}$, and thus the expected angular deviation α between the discrete path and the optimal continuous path is of order $\mathcal{O}((2 + \hat{l}/h)^{-2})$. The geometric length difference between the discrete path and the continuous path is of order $\mathcal{O}(1 - \cos\alpha) = \mathcal{O}(\alpha^2) = \mathcal{O}((2 + \hat{l}/h)^{-4})$. The expected length and therefore travel time error induced by angular deviation is smallest if $\hat{l}$ is largest, i.e., arcs of maximum length $2h + l$ are to be preferred.

This analysis also implies that the total flight time error is of order $\mathcal{O}(l^4 + h^2 + (h/l)^4)$ and that $h = o(l^2)$ ensures convergence for $l \to 0$.



(**a**) Distance error (**b**) Angular error

Figure 1. Geometrical error bounds. Blue line: continuous optimal solution. Black line: discrete optimal solution. Gray dots: graph nodes.

2.3.3. A^* Shortest Path Algorithm

The state-of-the-art for finding shortest paths in airway networks is the A^* algorithm [10], which extends Dijkstra's algorithm by a (heuristic) lower bound for the distance of some vertex x_i to the destination x_D in order to prioritize the search. Depending on the tightness of the heuristic bound, A^* can discard a substantial part of the graph and reduce the run time considerably. As a particularly fast and simple heuristic we employ the travel time along a straight line between x_i and x_D, assuming maximum tail wind, i.e.,

$$d(x_i, x_D) = \frac{\|x_i - x_D\|}{\bar{v} + \bar{w}}.$$

Tighter and more complex heuristics exploiting local bounds on the wind have been proposed for flight planning, also for time-dependent wind fields [10].

The travel time for each arc is again calculated via numerical integration of (12) using the same method as for the collocation with a fixed discretization. In order to draw up a

fair comparison, we process arcs on the fly and thus eliminate any major preprocessing. This also makes the approach extendible to the time dependent case.

2.3.4. Time Complexity

Using a Fibonacci heap for the priority queue, A^* can be implemented to run in time

$$R_D \in \mathcal{O}(|A| + |V| \log |V|), \tag{13}$$

where $|A|$ and $|V|$ are the numbers of arcs and vertices, respectively. As outlined above, we may assume $l \gg h$. If we—conservatively—assume $l^2 \geq h^2 \log h^{-1}$, the total complexity becomes

$$R_D = \mathcal{O}(l^2 h^{-4}).$$

2.3.5. Graph Structure

Given the above considerations, we now address the question of which graph structure in terms of h and l is the most efficient, i.e., which structure achieves a minimal error for a given computational budget b or, equivalently, a desired accuracy with minimal effort. Models for computational work and accuracy are

$$W(h,l) = l^2 h^{-4} \quad \text{and} \quad \epsilon(h,l) = l^4 + h^2 + h^4 l^{-4}.$$

We are hence interested in solving the optimization problem

$$\min \epsilon(h,l) \quad \text{s.t.} \quad W(h,l) = b.$$

The constraint $W(h,l) = b$ yields $l^2 = h^4 b$, which, when inserted into the objective, leads to the unconstrained problem

$$\min_h b^2 h^8 + h^2 + b^{-2} h^{-4} \quad \Leftrightarrow \quad \min_H BH^4 + H + B^{-1} H^{-2}.$$

The necessary optimality condition $4BH^3 + 1 - 2B^{-1}H^{-3} = 0$ is a quadratic equation in H^3 with solution

$$h^6 = \frac{-1 + \sqrt{33}}{8b^2} \quad \Leftrightarrow \quad b = \sqrt{\frac{-1 + \sqrt{33}}{8h^6}}.$$

Inserting this into the constraint yields $l^2 = h\sqrt{\frac{-1+\sqrt{33}}{8}}$. We conclude that graphs of optimal efficiency should follow the law

$$h = \mathcal{O}(l^2), \tag{14}$$

such that the computational complexity of the discrete optimization can be expressed as

$$R_D \in \mathcal{O}(h^{-3}) \Leftrightarrow R_D \in \mathcal{O}(l^{-6}) \tag{15}$$

with an associated error of $\mathcal{O}(l^4)$ in flight time and of $\mathcal{O}(l^2)$ in path approximation.

2.4. DisCOptER Algorithm

Due to their superior angular resolution, arcs of length l will occur in the optimal discrete path, while the length of flight path segments in the collocation solution is around $\delta\tau T\bar{v}$. Hence, we expect the accuracy of continuous and discrete optimization approaches to be comparable if $l = \delta\tau T\bar{v}$. Obviously, for increasing accuracy $l \to 0$, the collocation effort of order $\mathcal{O}(\delta\tau^{-1} \log \delta\tau / \log(\omega \|\chi_0 - \chi_*\|))$ grows much slower than the A^* effort of complexity $\mathcal{O}(l^{-6})$. On the other hand, the collocation approach converges only locally.

We therefore propose a hybrid algorithm that combines the strengths of discrete optimization and optimal control, see Algorithm 1. First, a discrete shortest path of low

accuracy is calculated using the A^* algorithm as described in Section 2.3. This is used as initial iterate for solving the KKT system (5) to the desired final accuracy using the ordinary Newton method. Employing linesearch or trust region globalization would enlarge the convergence domain of Newton's method and allow for coarser graphs to be used, but at the cost of less robust and efficient convergence. In favour of robustness and simplicity, we restrict the attention to the ordinary Newton method.

Algorithm 1: DisCOptER algorithm

Input : $x_O, x_D \in \mathbb{R}^2$, $w \in C^2(\mathbb{R}^2)^2$, $\bar{v} > 0$, TOL > 0
Output: approximate solution $(T, x, v) \in \mathbb{R} \times C^0([0,1])^2 \times L^2([0,1])^2$ of (1) with error of order TOL^2, TOL, TOL, respectively

1 Choose $\delta\tau = \mathcal{O}(\sqrt{\mathrm{TOL}})$
2 Define (h, l)-dense graph G with $l > \delta\tau\|x_D - x_O\|$, but sufficiently small, and $h = \mathcal{O}(l^2)$
3 Calculate shortest path on G using A^* algorithm (see Section 2.3)
4 Interpolate path onto collocation discretization (see Section 2.4.1)
5 Calculate a continuous solution by solving the nonlinear problem (5), via direct collocation and Newton's method (see Section 2.2)

Algorithms that combine methods from discrete and continuous optimization have been proposed before. A reverse deterministic combination going from continuous to discrete has been proposed in [41] based on dynamic programming principles. Other combinations involve a stochastic discrete stage, like rapidly-exploring random trees (RRT, RRT*) (see, e.g., [42]) or Probabilistic Roadmaps (PRMs) [43] in combination with a second NLP stage. Another approach was taken in [44], where the authors use a combination of A^* and RRT* to find an optimal trajectory. These algorithms reveal remarkable performance when it comes to obstacle avoidance. Since our goal is, however, to develop an algorithm with a priori error estimates and bounded runtime, we do not make use of any stochastic approaches in the DisCOptER algorithm.

2.4.1. Initialization

Let $(x_0, \ldots, x_n) \in P$ be a shortest path from x_O to x_D in the graph G. Let t_i denote the time at which waypoint x_i is passed, and define the relative passage time $\tau_i = t_i / T(p) \in [0,1]$. We define a mapping $\Xi : P \to C^{0,1}([0,1])^2$ of discrete flight paths $(x_0, \ldots, x_n) \in P$ to continuous paths $x \in C^{0,1}([0,1])^2$ by piecewise linear interpolation:

$$x(\tau) = x_i + \frac{T(p)\tau - t_i}{t_{i+1} - t_i}(x_{i+1} - x_i) \quad \text{if } t_i \le T(p)\tau \le t_{i+1}. \tag{16}$$

The initial collocation iterate is then obtained by evaluating $x(\tau_i)$ on the collocation grid $0{=}\tau_0, \ldots, \tau_n{=}1$ and defining the airspeeds

$$v_i = \frac{x(\tau_{i+1}) - x(\tau_i)}{T(p)(\tau_{i+1} - \tau_i)} - w((x(\tau_{i+1}) + x(\tau_i))/2)$$

at the interval midpoints $\tau_{i+1/2}$.

2.4.2. Complexity

The runtime of this hybrid algorithm is comprised of the runtime of the A^* algorithm (13) and the runtime (10) of the KKT-Newton solver for the collocation system. Provided that the initial iterate based on the discrete solution p is sufficiently close to the continuous optimum, i.e., $\omega\|\chi_0 - \chi_*\| < 1$, Newton's method converges locally as

described by (10). For the path approximation error, we expect $\|\chi_0 - \chi_*\| \approx Cl^2$, which leads to an overall complexity of

$$R_H = R_D + R_C \in \mathcal{O}\left(l^{-6} + \delta\tau^{-1}\frac{\log \delta\tau}{\log(\omega C l^2)}\right), \tag{17}$$

or $R_H = \mathcal{O}(l^{-6} + \delta\tau^{-1})$ subject to $\omega C l^2 < 1$ if an inexact Newton method is used. Both complexity bounds essentially suggest to choose a graph as sparse as possible ($l \to \infty$), only restricted from above by the accuracy necessary for the Newton-KKT solver to converge locally, i.e., $\omega C l^2 < 1$, which is independent of the final accuracy $\delta\tau$.

3. Results

We validate in this section the effectiveness and the efficiency of our algorithm on a test set of four problems of increasing difficulty. Our aim is to demonstrate that our hybrid approach is asymptotically superior to a purely graph based alternative. We discuss the convergence properties of the method in relation to the choice of the crossover point and its dependence on the minimum required graph density. Based on these results, we evaluated the DisCOptER algorithm for varying graph densities, using the theoretically optimal graph structure of $h = l^2$, cf. (14). We will see that—as expected—the best performance was achieved on very sparse graphs. The chapter concludes with a computational comparison of the hybrid algorithm with a purely discrete approach.

3.1. Test Problems

We tested our algorithm on a set of simple, but representative examples that were well suited to demonstrate our method, and not far from real world situations or arguably even more difficult. The instances lived in a square $[0,1]^2$ or $[0,1] \times [-1,1]$, and the origin and the destination were $x_O = [0,0]^T$ and $x_D = [1,0]^T$, respectively. All values were chosen dimensionless. The wind fields were constructed in a such a way that the straight line from the origin to the destination was particularly unfavorable. The wind speed was limited to $\bar{w} = \frac{1}{2}\bar{v}$, which was rather strong, but not unrealistic. Even though not formally required, the graph nodes were positioned on a uniform Cartesian grid, such that the diagonal distance of two adjacent nodes was $2h$. For the sake of simplicity the graph structure will be described based on the node spacing in x-direction $h_x = \sqrt{2}h$. This directly defined $l = \sqrt{h}$ by (14).

In the first test problem (a), the wind field was a laminar flow of opposing parallel currents, namely,

$$w(x) = \begin{bmatrix} \bar{w}\min(\max(2\frac{x_2}{H} - 1, -1), 1) \\ 0 \end{bmatrix},$$

with $H = 0.5$, see Figure 2; a similar problem is discussed in [45]. Due to its simplicity, this problem had only one distinct minimum, a property that we make use of in the next section.

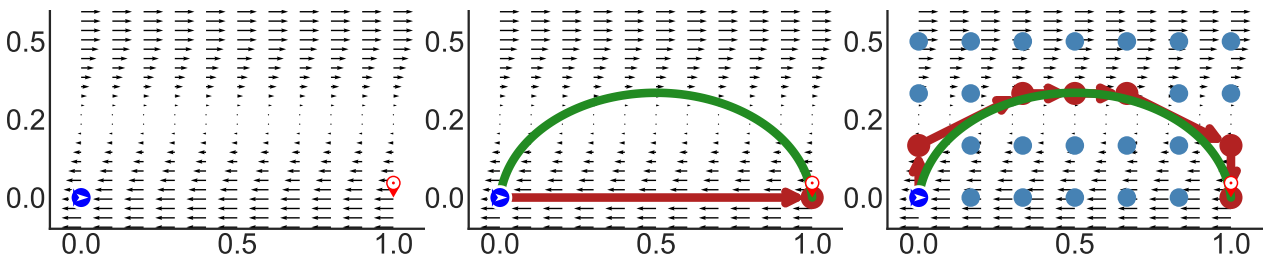


Figure 2. Test problem (a) with $H = 0.5$, $x_O = (0,0)$, $x_D = (1,0)$, $\bar{w} = \frac{1}{2}\bar{v}$. Blue dots: network-graph with $h = l^2$ and $1/h_x = 1$ and 6, respectively, red: discrete optimal trajectory, green: continuous optimal trajectory.

In problems (b)–(d), the wind w was the sum of an increasing number of vortices w_i, each of which was described by

$$w_i(x) = \begin{bmatrix} -s\tilde{w}(r)\sin(\alpha) \\ s\tilde{w}(r)\cos(\alpha) \end{bmatrix},$$

where s is the spin of the vortex (s=+1: counter-clockwise, s=−1: clockwise), $r = \|x - z\|_2$ is the distance from the vortex center z_i, α is the angle with respect to the center and the x-axis with $\tan(\alpha) = \frac{(x-z_i)_2}{(x-z_i)_1}$ and the absolute vortex wind speed $\tilde{w}$ is a function of r and the vortex radius R:

$$\tilde{w}(r) = \begin{bmatrix} \bar{w}\exp\left(\frac{r^2}{r^2-R^2}\right) & \text{if } r \leq R \\ 0 & \text{otherwise} \end{bmatrix}. \tag{18}$$

Problem (b) involved one large vortex with $R = 1/2$ at $z = [0.5, -0.1]^T$, see Figure 3. This caused Newton's method to converge to a suboptimal trajectory (above the vortex) if initialized with the straight trajectory. That was the case if the DisCOptER algorithm was used with a minimally sparse graph with $h_x = 1$ (see Figure 3, middle). We observed that the discrete shortest path passed below the vortex for any $h_x > 1$. From there the globally optimal solution shown in the Figure 3 (left 3) was found for $h_x = 1/6$.

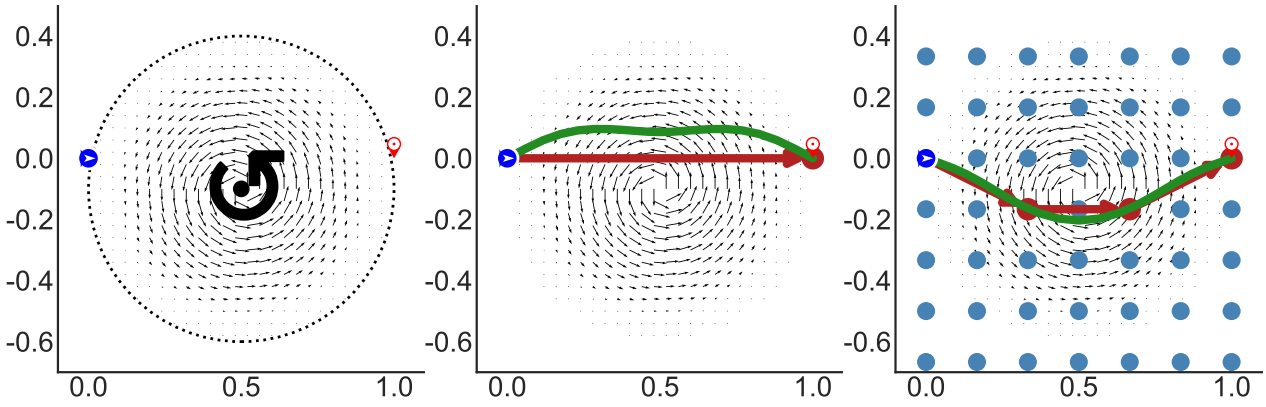


Figure 3. Test problem (b) with counterclockwise spinning vortex centered at $z = [0.5, -0.1]^T$, $R = 0.5$. $x_O = [0, 0]^T$, $x_D = [1, 0]^T$, $\bar{w} = \frac{1}{2}\bar{v}$. Blue dots: network graph with $h = l^2$ and $1/h_x = 1$ and 6, respectively, red: discrete optimal trajectory, green: continuous optimal trajectory.

Problem (c) involved 15 vortices with $R = 1/8$. One was centered at $z = [0.5, -R/2]^T$, the others were regularly aligned as seen in Figure 4. Vortices with positive spin (clockwise) were colored green, vortices with negative spin (anti-clockwise) were colored red. Due to the turbulence of this wind field, a plain application of Newton's method was not guaranteed to converge. As an example, (Figure 4, left 2) shows again the result with the trivial initialization. Further there are several local minima, Figure 4 (left 3 and 4) show two of them. A relatively high graph density ($h_x \leq 1/17$) was required to find the globally optimal trajectory (Figure 4, left 5).

Problem (d) involved 50 regularly aligned vortices of radius $R = 1/16$ (Figure 5). This was clearly an exaggeration, and no commercial plane would ever try to traverse a wind field like this. We used the instance to show that the proposed algorithm outperformed existing methods even under the most adverse conditions. In fact, the high level of non-convexity exacerbated the situation regarding the convergence of Newton's method. Note that the magnitude of derivatives of (18) was coupled directly to the vortex radius. With $R = 1/16$ the vortices here were half as large as in the previous example. In turn, the first and second order derivatives were 2 and 4 times larger, respectively. Consequently, a quite dense graph with $h_x \leq 1/60$ was required to make Newton's method converge

reliably. Note that this wind field was point-symmetric with respect to $[0.5, 0]^T$, which allowed for two equivalent global optima (see Figure 5, left 2 and 3). It was a priori not obvious which one would be found. This depended on the discrete optimum.

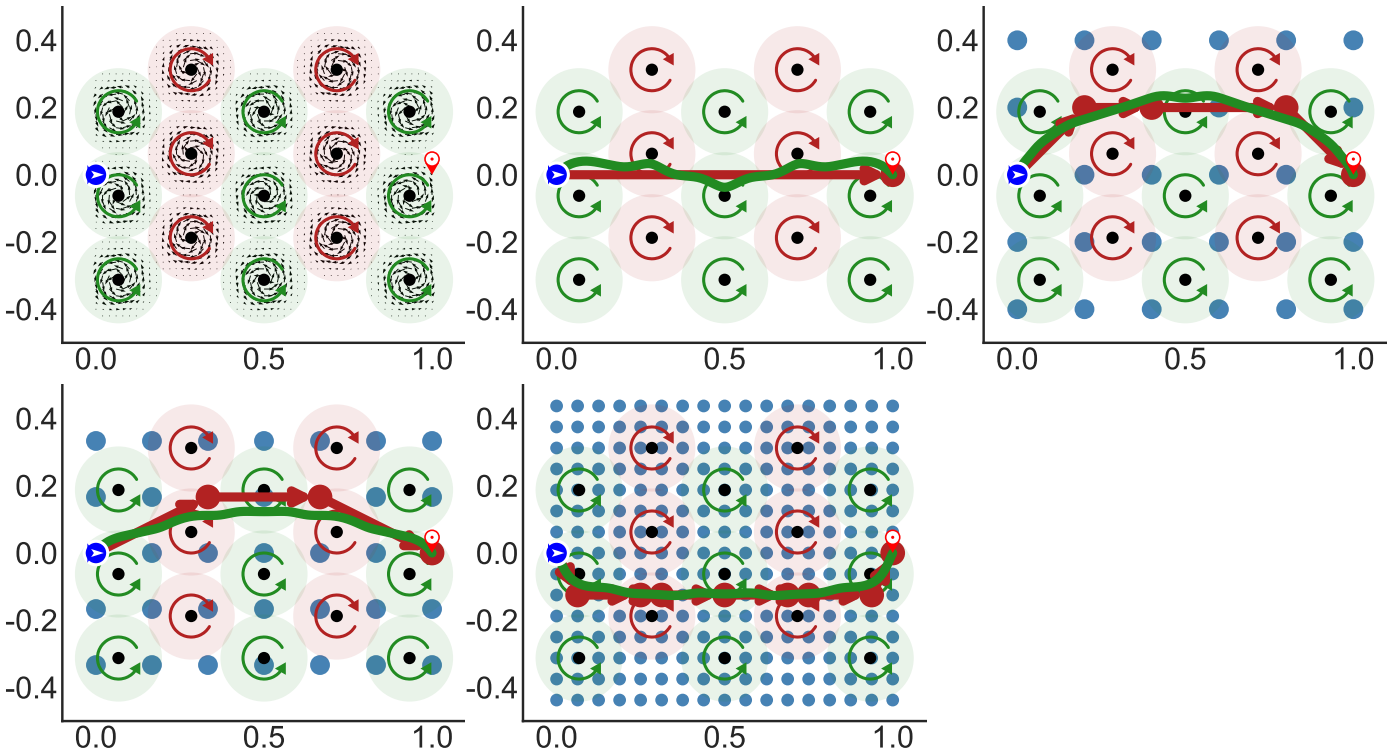


Figure 4. Test problem (c) with 15 vortices, $R = 1/8$. One is centered at $z = [0.5, -R/2]$, the others are regularly positioned as seen above. $x_O = [0, 0]^T$, $x_D = [1, 0]^T$, $\bar{w} = \frac{1}{2}\bar{v}$. Blue dots: graph with $h = l^2$ and $1/h_x = 1, 5, 6$, and 16, respectively, red: discrete optimal trajectory, green: continuous optimal trajectory. Note that the straight trajectory is particularly unfavorable.

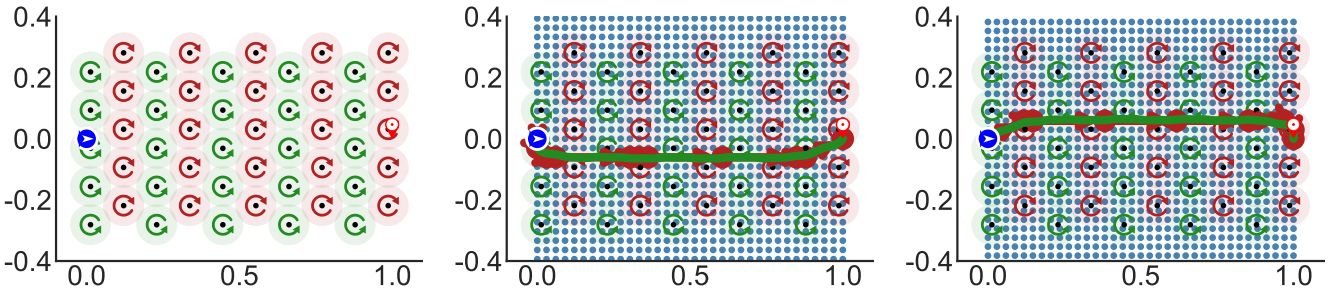


Figure 5. Test problem (d) with 50 vortices (10 columns of 5 vortices each), $R = 1/16$, regularly positioned. $x_O = [0, 0]^T$, $x_D = [1, 0]^T$, $\bar{w} = 0.5\bar{v}$. Blue dots: graph with $h = l^2$ and $1/h_x = 33$ and 34, respectively, red: discrete optimal trajectory, green: continuous optimal trajectory. Note that, again, the straight trajectory is particularly unfavorable. This problem is point-symmetric w.r.t. $[0.5, 0]^T$. The two shown solutions are equivalent.

3.2. Computational Complexity

Before discussing the DisCOptER algorithm we validated that the optimal control methods were asymptotically more efficient than a purely graph-based approach. This could be investigated only for the first test problem. Due to the simplicity of this wind field, Newton's method converged from a trivial initialization, i.e., the straight line from x_O to x_D. On the other hand, discrete flight paths were calculated with varying accuracy, that is, with varying l, which is, according to Section 2.3.2, a measure for the accuracy of the calculated path. For the sake of consistency, we indicate the accuracy of the continuous solution by $l_C := \delta\tau L$, where L is the path length. Figure 6 clearly confirms the estimated

time complexities of $\mathcal{O}(l^{-6})$ for the discrete approach (see Equation (13)) and $\mathcal{O}(\delta\tau^{-1})$ for the continuous approach (see Equation (10)).

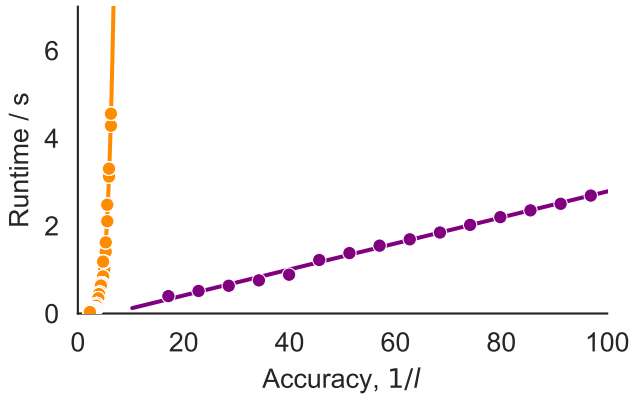


Figure 6. Test case (a). Orange: Discrete-only approach, polynomial trend line of order 6. Purple: Newton-Karush–Kuhn–Tucker (KKT) initialized with straight line, $l \hat{=} l_C = \delta\tau L$, linear trend line.

3.3. Minimum Graph Requirements

Figure 7 shows the runtime of the DisCOptER algorithm for various graph parameters (h, l). Some key observations could be made for all four test problems. Towards the top right of each figure the graph density increased, which came with an increased runtime dictated by the graph searching part of the algorithm. The black dashed line represents the theoretically optimal graph structure $h = l^2$, cf. (14); it is shown for orientation as the results in the following sections were computed with this setting.

From left to right the wind fields became increasingly more non-convex. As these plots reveal, this came with generally higher runtimes, but more importantly with a region of low graph densities where the algorithm converged either to a local minimum or not at all (gray areas). As discussed before, this was not an issue in case (a) since this problem was convex and Newton's method converged even from a trivial initialization. Even in case (b) we found that a graph with $h_x < 1$ was good enough such that we found the optimal flight path (only one additional column of nodes between start and destination was required). This outcome is not visible here due to the limited resolution of the plot. The convergence problem became all the more apparent with instances (c) and (d). In both cases a good number of local minima existed, each of which had a rather small radius of convergence such that Newton's method failed if not initialized with sufficient accuracy. Especially in case (d) we saw a patchwork of runs that converged presumably by chance in an unpredictable way. Some of this might be compensated by globalization techniques and an explicit treatment of non-convexity, which, however, would affect the efficiency of Newton's method. In order to provide reliable results, we need to use a graph density of at least $l < 0.15$ and 0.11 for case (c) and (d), respectively.

Interestingly, the node distance h appeared to have a much stronger effect on the convergence than the connectedness width l. This might be explained to some extent in terms of the distance and angular error (cf. Section 2.3.2). Low l decreased the angular resolution and thus induced an increased angular error. The discrete flight path then tended to zig-zag along the optimum. This could easily be smoothed by Newton's method. However, if the discrete flight path was parallely offset from the optimum (distance error due to large node distance h), it might quickly leave the convergence radius of Newton's method.

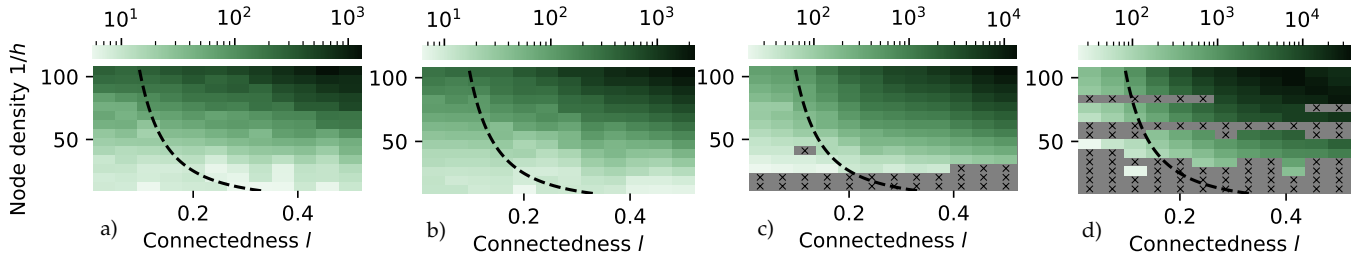


Figure 7. Runtime of the DisCOptER algorithm in seconds. Top right corner: dense graph. Bottom left corner: sparse graph. Gray areas: not converged or converged to local minimum. Black dashed line: $h = l^2$. Subfigures refer to the corresponding test problems.

3.4. Optimal Crossover Point

In Section 2.3.2 we derived $h = l^2$ as the optimal graph structure, cf. (14). Using this setting and sampling over various graph densities for the test problems led to the results presented in Figure 8. Obviously, an increased graph density came with a computationally more expensive graph searching task (pink, top, cf. (15)). In turn the NLP part of the algorithm (green, bottom) becomes cheaper following (10), since the initial guess gets closer to the optimum. From test problem a) it can be seen that the best performance was achieved—independently of the overall accuracy $\delta\tau$—with a relatively low graph density of $1/l \approx 5$, where the graph search was still negligibly cheap but the graph was already dense enough for Newton's method to converge with only one iteration. The exact numbers depended strongly on implementation details. We do not claim to have an optimal realization of either part of the algorithm. The trend towards low graph densities, however, was confirmed by the following examples.

As it turned out, the lower bound on the required graph density imposed by the non-convexity of the wind field was the more important criterion. Looking at examples (c) and (d), where we excluded the area that we identified as not trustworthy in the previous section, we concluded that we want the graph to be as sparse as possible and only as dense as necessary.

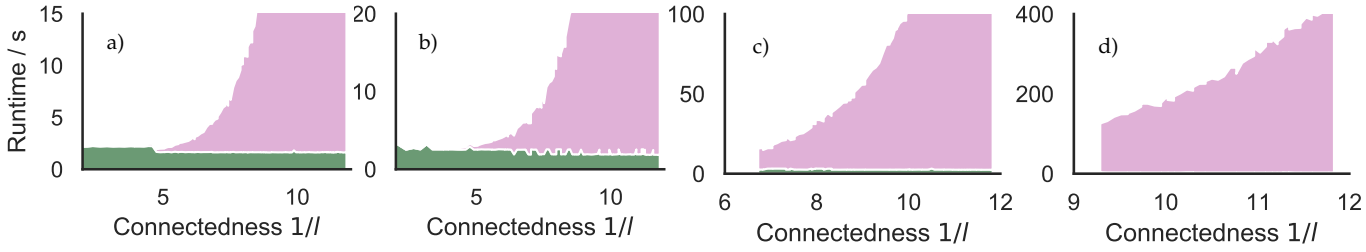


Figure 8. Runtimes of the DisCOptER algorithm in seconds, split into the discrete part (pink, top) and the continuous part (green, bottom), sampled with $h = l^2$. Constant accuracy $\delta\tau = 1/300$. Subfigures refer to the corresponding test problems.

3.5. Computational Complexity

We finally showed that globally optimal shortest paths could be calculated more efficiently using the proposed DisCOptER algorithm than with a purely graph based approach, see Figure 9. In the previous section we showed that the algorithm performed best if the graph was chosen rather sparse while respecting the problem-specific minimum density. Consequently, the calculation of the discrete solution was relatively cheap and we could benefit from the computational efficiency of Newton's algorithm. We can also confirm that the time complexity of the DisCOptER Algorithm is $\mathcal{O}(\delta\tau^{-1})$, see Equation (17), and that the time complexity of the purely graph-based approach is $\mathcal{O}(l^{-6})$, see (13).

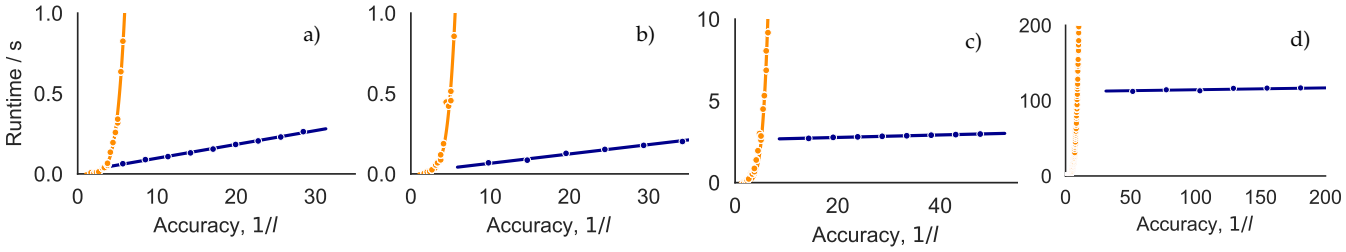


Figure 9. Minimum runtime in seconds taken experiments similar to Figure 8. Blue: DisCOptER algorithm with $l_c := \delta\tau L$, linear trend line. Orange: Purely discrete, polynomial trend line of order 6. Subfigures refer to the corresponding test problems.

4. Conclusions

In this paper we presented the novel DisCOptER algorithm to calculate flight paths in free flight airspaces utilizing a combination of discrete and continuous optimization. We demonstrated that the achieved efficiency is asymptotically much better than the conventional purely discrete alternative. Even though the algorithm was described for the static two-dimensional case it is also strongly promising for more complex cases, to which it can directly be transferred.

Our study also reveals a need for more theoretical analysis of the problem. In order to design a one-shot algorithm with theoretical efficiency guarantees, a priori error estimates allowing the determination of a minimum required crossover graph density is needed. This will of course depend mainly on the characteristics of the wind field including first and second order derivatives. On the other hand, a posteriori error estimates and adaptive coarse-to-fine graph refinement strategies will be necessary for robustness and efficiency in practice. We investigate some of these questions in [40].

Author Contributions: Conceptualization, R.B. and M.W.; methodology, M.W.; software, F.D.; validation, F.D.; formal analysis, M.W.; investigation, F.D. and M.W.; resources, R.B., F.D. and M.W.; data curation, F.D.; writing—original draft preparation, F.D. and M.W.; writing—review and editing, R.B.; visualization, F.D.; supervision, R.B.; project administration, R.B. and M.W.; funding acquisition, R.B. and M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the DFG Research Center of Excellence MATH$^+$ – Berlin Mathematics Research Center, Project AA3-3.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stojković, S.E.K.S.S.A.G.; Stojković, M. *Quantitative Problem Solving Methods in the Airline Industry*; Springer: Berlin/Heidelberg, Germany, 2011; Chapter 6—Operations, pp. 283–383.
2. Airbus Industries. Getting to Grips with Fuel Economy. Issue 4, 2004. Available online: https://ansperformance.eu/library/airbus-fuel-economy.pdf (accessed on 24 December 2020).
3. Radio Technical Commission for Aeronautics. *Final Report of RTCA Task Force 3 Free Flight Implementation*; RTCA: Washington, DC, USA, 1995.
4. Jelinek, F.; Carlier, S.; Smith, J.; Quesne, A. *The EUR RVSM Implementation ProjectEnvironmental Benefit AnalysisEEC/ENV/2002/008*. Technical Report; 2003. Available online: https://www.eurocontrol.int/eec/gallery/content/public/document/eec/report/2002/023_RVSM_Implementation_Project.pdf (accessed on 24 December 2020).
5. Delling, D.; Wagner, D. Time-Dependent Route Planning. In *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*; Ahuja, R.K., Möhring, R.H., Zaroliagis, C.D., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 207–230. [CrossRef]
6. Bast, H.; Delling, D.; Goldberg, A.; Müller-Hannemann, M.; Pajor, T.; Sanders, P.; Wagner, D.; Werneck, R.F. Route Planning in Transportation Networks. 2015. Available online: https://arxiv.org/pdf/1504.05140.pdf (accessed on 24 December 2020).
7. Zis, T.P.; Psaraftis, H.N.; Ding, L. Ship weather routing: A taxonomy and survey. *Ocean Eng.* **2020**, *213*, 107697. [CrossRef]
8. Yang, L.; Qi, J.; Song, D.; Xiao, J.; Han, J.; Xia, Y. Survey of Robot 3D Path Planning Algorithms. *J. Control Sci. Eng.* **2016**, *2016*, 1687–5249. [CrossRef]
9. Blanco, M.; Borndörfer, R.; Hoang, N.D.; De las Casas, A.K.P.M.; Schlechte, T.; Schlobach, S. Cost projection methods for the shortest path problem with crossing costs. In Proceedings of the 17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2017), Vienna, Austria, 7–8 September 2017; Volume 59.

10. Blanco, M.; Borndörfer, R.; Hoang, N.D.; Kaier, A.; Schienle, A.; Schlechte, T.; Schlobach, S. Solving time dependent shortest path problems on airway networks using super-optimal wind. In Proceedings of the 16th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2016), Aarhus, Denmark, 25 August 2016. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
11. Larsen, K.; Knudsen, A.; Chiarandini, M. Constraint handling in flight planning. In *Principles and Practice of Constraint Programming*; Lecture Notes in Computer Science; Beck, J., Ed.; Springer, Cham, Switzerland, 2017; Volume 10416, pp. 354–369.
12. Marchidan, A.; Bakolas, E. Numerical Techniques for Minimum-Time Routing on a Sphere with Realistic Winds. *Am. Inst. Aeronaut. Astronaut.* **2016**, *39*, 188–193. [CrossRef]
13. Jardin, M.R.; Bryson, A.E. Methods for computing minimum-time paths in strong winds. *J. Guid. Control Dyn.* **2012**, *35*, 165–171. [CrossRef]
14. McDonald, J.A.; Zhao, Y. Time benefits of free-flight for a commercial aircraft. In Proceedings of the AIAA Guidance Navigation & Control Conference, Dever, CO, USA, 14–17 August 2000.
15. von Stryk, O.; Bulirsch, R. Direct and indirect methods for trajectory optimization. *Ann. Oper. Res.* **1992**, *37*, 357–373. [CrossRef]
16. Betts, J.; Cramer, E. Application of direct transcription to commercial aircraft trajectory optimization. *J. Guid. Contr. Dyn.* **1995**, *18*, 151–159. [CrossRef]
17. García-Heras, J.; Soler, M.; Sáez, F.J. A Comparison of Optimal Control Methods for Minimum Fuel Cruise at Constant Altitude and Course with Fixed Arrival Time. *Procedia Eng.* **2014**, *80*, 231–244. [CrossRef]
18. Diedam, H.; Sager, S. Global optimal control with the direct multiple shooting method. *Optim. Control Appl. Meth.* **2017**, *39*, 1–22. [CrossRef]
19. Sager, S.; Reinelt, G.; Bock, H. Direct Methods with Maximal Lower Bound for Mixed-Integer Optimal Control Problems. *Math. Program.* **2009**, *118*, 109–149. [CrossRef]
20. Rao, A. A Survey of Numerical Methods for Optimal Control. *Adv. Astronaut. Sci.* **2010**, *135*, 497–528.
21. Bonami, P.; Olivares, A.; Soler, M.; Staffetti, E. Multiphase mixed-integer optimal control approach to aircraft trajectory optimization. *J. Guid. Control Dyn.* **2013**, *36*, 1267–1277. [CrossRef]
22. Moreno, L.; Fügenschuh, A.; Kaier, A.; Schlobach, S. A Nonlinear Model for Vertical Free-Flight Trajectory Planning. In *Operations Research Proceedings*; Springer: Cham, Switzerland, 2018; pp. 445–450. [CrossRef]
23. Zermelo, E. Über das Navigationsproblem bei ruhender oder veränderlicher Windverteilung. *ZAMM Z. Angew. Math. Mech.* **1931**, *11*, 114–124. [CrossRef]
24. Maurer, H.; Zowe, J. First and second-order necessary and sufficient optimality conditions for infinite-dimensional programming problems. *Math. Program.* **1979**, *16*, 98–110. [CrossRef]
25. Pontrjagin, L.; Boltyansky, V.; Gamkrelidze, V.; Mischenko, E. *Mathematical Theory of Optimal Processes*; Wiley-Interscience: New York, NY, USA, 1962.
26. Betts, J.T. Survey of Numerical Methods for Trajectory Optimization. *J. Guid. Control. Dyn.* **1998**, *21*, 193–207. [CrossRef]
27. Hargraves, C.; Paris, S. Direct trajectory optimization using nonlinear programming and collocation. *J. Guid. Control Dyn.* **1987**, *10*, 338–342. [CrossRef]
28. Fahroo, F.; Ross, I.M. Direct Trajectory Optimization by a Chebyshev Pseudospectral Method. *J. Guid. Control Dyn.* **2002**, *25*, 160–166. [CrossRef]
29. Ascher, U.; Mattheij, R.; Russell, R. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*; Prentice Hall: Upper Saddle River, NJ, USA, 1988.
30. Hagelauer, P.; Mora-Camino, F. A soft dynamic programming approach for on-line aircraft 4D-trajectory optimization. *Eur. J. Oper. Res.* **1998**, *107*, 87–95. [CrossRef]
31. Steihaug, T. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.* **1983**, *20*, 626–637. [CrossRef]
32. Toint, P. Towards an efficient sparsity exploiting Newton method for minimization. In *Sparse Matrices and Their Use*; Duff, I., Ed.; Academic Press: Cambridge, MA, USA, 1981; pp. 57–88.
33. Weiser, M.; Deuflhard, P.; Erdmann, B. Affine conjugate adaptive Newton methods for nonlinear elastomechanics. *Optim. Meth. Softw.* **2007**, *22*, 413–431. [CrossRef]
34. Deuflhard, P.; Bornemann, F. *Scientific Computing with Ordinary Differential Equations*, 2nd ed.; Texts in Applied Mathematics; Springer: New York, NY, USA, 2002; Volume 42.
35. Becker, R.; Kapp, H.; Rannacher, R. Adaptive finite element methods for optimal control of partial differential equations: basic concepts. *SIAM J. Control Optim.* **2000**, *39*, 113–132. [CrossRef]
36. Weiser, M. On goal-oriented adaptivity for elliptic optimal control problems. *Optim. Meth. Softw.* **2013**, *28*, 969–992. [CrossRef]
37. Deuflhard, P. *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*; Computational Mathematics; Springer: New York, NY, USA, 2004; Volume 35.
38. Weiser, M.; Schiela, A.; Deuflhard, P. Asymptotic Mesh Independence of Newton's Method Revisited. *SIAM J. Numer. Anal.* **2005**, *42*, 1830–1845. [CrossRef]
39. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [CrossRef]
40. Borndörfer, R.; Danecker, F.; Weiser, M. *Error Bounds for Free Flight Planning with Locally Connected Airway Networks*; Technical Report; Zuse Institute Berlin: Berlin, Germany. in preparation.

41. Junge, O.; Osinga, H. A set oriented approach to global optimal control. *ESAIM: Contr. Opt. Calc. Var.* **2004**, *10*, 259–270. [CrossRef]
42. Karatas, T.; Bullo, F. Randomized searches and nonlinear programming in trajectory planning. In Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228), Orlando, FL, USA, 4–7 December 2001; Volume 5, pp. 5032–5037. [CrossRef]
43. Bouffard, P.; Waslander, S. A Hybrid Randomized/Nonlinear Programming Technique For Small Aerial Vehicle Trajectory Planning in 3D. *Plan. Percept. Navig. Intell. Veh. (PPNIV)* **2009**, *63*, 2009.
44. Brunner, M.; Brüggemann, B.; Schulz, D. Hierarchical Rough Terrain Motion Planning using an Optimal Sampling-Based Method. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; [CrossRef]
45. Techy, L. Optimal navigation in planar time-varying flow: Zermelo's problem revisited. *Intell. Serv. Robot.* **2011**, *4*, 271–283. [CrossRef]