Accepted Manuscript

Trajectory Planning for Autonomous Underwater Vehicles in the Presence of Obstacles and a Nonlinear Flow Field using Mixed Integer Nonlinear Programming

Tong Wang, Ricardo M. Lima, Loïc Giraldi, Omar M. Knio

PII: S0305-0548(18)30227-2
DOI: <https://doi.org/10.1016/j.cor.2018.08.008>
Reference: CAOR 4544

To appear in: *Computers and Operations Research*

Received date: 13 August 2017
Revised date: 18 March 2018
Accepted date: 17 August 2018

Please cite this article as: Tong Wang, Ricardo M. Lima, Loïc Giraldi, Omar M. Knio, Trajectory Planning for Autonomous Underwater Vehicles in the Presence of Obstacles and a Nonlinear Flow Field using Mixed Integer Nonlinear Programming, *Computers and Operations Research* (2018), doi: <https://doi.org/10.1016/j.cor.2018.08.008>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Highlights

- The time-optimal trajectory planning for autonomous underwater vehicles is considered.
- Detailed and simplified MINLP models considering obstacles and a flow field are proposed.
- A solution approach with an enhanced initialization and MILP models is implemented.
- A detailed analysis of the MILP models and performance of the MILP solver is made.
- The performance of the MINLP methodology and quality of the trajectories are analyzed.

Trajectory Planning for Autonomous Underwater Vehicles in the Presence of Obstacles and a Nonlinear Flow Field using Mixed Integer Nonlinear Programming

Tong Wang^{a,b,1}, Ricardo M. Lima^{a,*}, Loïc Giraldi^a, Omar M. Knio^a

^aComputer, Electrical and Mathematical Sciences & Engineering Division, King Abdullah University of Science and Technology, (KAUST), Thuwal 23955-6900, Saudi Arabia

^bCollege of Information Science and Technology, Donghua University, Shanghai, China.

Abstract

This paper addresses the time-optimal trajectory planning for autonomous underwater vehicles. A detailed mixed integer nonlinear programming (MINLP) model is presented, explicitly taking into account vehicle kinematic constraints, obstacle avoidance, and a nonlinear flow field to represent the ocean current. MINLP problems pose great challenges because of the combinatorial complexity and nonconvexities introduced by the nature of the flow field. A novel solution approach in an optimization framework is developed to address associated difficulties. The main benefit of the proposed methodology is the ability to find multiple local minima. The contribution of the paper is fourfold: 1) a novel approach to integrate the flow field into the MINLP model; 2) a diversified initialization strategy using multiple waypoints, different solvers and approximated models, namely, a mixed integer linear programming model and the MINLP model with and without the flow field; 3) an algorithm that forces the solver to seek improved solutions; and 4) a parallel computing approach capitalizing on diversified initialization. The performance of the resulting methodology is illustrated on idealized case studies, and the results are used to gain insight into trajectory planning in the presence of flow fields.

Keywords: Optimal trajectory planning, MINLP, MILP

1. Introduction

We address the trajectory planning for an Autonomous Underwater Vehicle (AUV) moving in a flow field while avoiding obstacles. Trajectory planning involves the determination of a geometric path and the motion profile along the path (Carbone and Gomez-Bravo, 2015; LaValle, 2006). In this work, the path planning and trajectory planning are merged into one problem and are determined simultaneously. In an underwater scenario, this is a crucial task because the vehicle dynamics are substantially affected by the vehicle's velocity and acceleration

*Corresponding author.

Email addresses: tong.wang@kaust.edu.sa (Tong Wang), ricardo.lima@kaust.edu.sa (Ricardo M. Lima), loic.giraldi@kaust.edu.sa (Loïc Giraldi), omar.knio@kaust.edu.sa (Omar M. Knio)

¹T. Wang was with the Division of Computer, Electrical and Mathematical Sciences and Engineering, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. She is now with the College of Information Science and Technology, Donghua University, Shanghai, China.

as well as the current velocity along the trajectory. Specifically, it is possible that even though it may be geometrically feasible, the trajectory can be dynamically infeasible in strong current conditions. Consequently, it is essential to determine the velocity and acceleration profiles of the vehicle to ensure a proper trajectory that also avoids obstacles.

The performance of trajectory planning is characterized in terms of a cost for the vehicle. An efficient trajectory minimizes the travel time to complete the task, the fuel/energy expenditure, or an appropriate combination thereof. In this paper, we focus on a time-optimal objective and accordingly formulate trajectory planning as an optimization problem accounting for the predicted flow field, and the constraints imposed by obstacles and by the vehicle dynamics.

Path planning methods were motivated by the determination of a geometric path between a starting and an ending point within a space with obstacles to be followed by robots. The main methods to address this problem are based on of the following methods (Latombe, 1991; Goerzen et al., 2010): a) roadmap methods; b) exact cell decomposition; c) approximate cell decomposition; d) potential field methods; e) probabilistic approaches; f) weighted region problem.

Trajectory planning methods can be broadly divided into three main categories, each one with several methods and variants: a) sampling-based methods; b) grid-based methods; c) optimization using random search: simulated annealing, and evolutionary algorithms; and d) optimization via mathematical programming. Rapidly-exploring Random Trees (RRT)(LaValle, 1998) is a popular sampling-based method, which randomly builds a space-filling tree to represent the collision-free areas that can generate feasible solutions quickly and sample high dimensional spaces, but it may lead to suboptimal solutions; see an additional discussion in Lolla (2012). In grid-based methods such as the A* algorithm (Hart et al., 1968), level set methods (Osher and Sethian, 1988), Fast Marching Method (FMM) (Sethian, 1999), a regular grid is imposed over the physical search area and used to constrain the motion of the vehicle, namely by limiting directions. Random search-based optimization methods are derivative-free and use random strategies and evolutionary concepts to explore the search space and escape local optima; see Carbone and Gomez-Bravo (2015) for references on the application of these algorithms and Luo and Hauser (2014) for a comparison on the performance of these methods.

In trajectory planning, physical obstacles or forbidden areas may be present in the region of interest, which create holes in the collision-free space, leading to a discontinuous search space. Therefore, computational difficulties arise in mathematical programming, and suitable approaches are needed to address these difficulties. The optimal trajectory planning containing collision-free constraints and the linear kinematic equations governing the continuous motion of the vehicle can be posed as a Mixed Integer Linear Programming (MILP) problem. MILP problems have been proposed for unmanned aerial vehicles by Schouwenaars et al. (2001), Richards and How (2002), and Richards et al. (2002). After these initial works, several other authors have used MILP models for a) unmanned aerial vehicles (Schouwenaars et al., 2004b; Borrelli et al., 2006; Schouwenaars et al., 2006; Vitus et al., 2008; Kuwata and How, 2011; Richards and Turnbull, 2015; Turnbull et al., 2016; Grøtli and Johansen, 2016; Radmanesh and Kumar, 2016; Richards and How, 2006); and b) wheeled robots (Earl and D'Andrea, 2005).

From the works above, there are three solution approaches that were developed to handle the size and complexity of the MILP models: a) receding horizon algorithms that break the original problem into a solution sequence of smaller problems (Schouwenaars et al., 2004a); b) iterative solution of MILP problems, where iteratively variable lengths of the time intervals are determined, so that a minimum number of time intervals is used, and therefore, smaller problems are solved (Earl and D’Andrea, 2005); and c) convex decompositions of the space based on the distribution of the obstacles, which leads to problems with fewer binary variables (Vitus et al., 2008; Park et al., 2015). In Vitus et al. (2008), it is also proposed the utilization of an A* algorithm and an FMM to determine an initial path to be considered by the MILP problem. The path optimization of unmanned aerial vehicles has also been extended to deal with routing problems involving task assignments by Edison and Shima (2011), Mufalli et al. (2012), and Shetty et al. (2008). Compared with MILP, MINLP has received less attention. Fukushima et al. (2013) reduced an optimal control problem to mixed-integer quadratic programming (MIQP) problem for multiple vehicles formation. For a review of mathematical programming techniques applied to motion planning problems see Abichandani et al. (2012).

The trajectory planning methods for AUVs are extensions of trajectory planning methods developed for wheeled robots or unmanned aerial vehicles, which include specific approaches to handle the flow fields. Rao and Williams (2009) proposed an RRT method for the path planning of underwater gliders taking into account the ocean current and they compare it with a grid-based method. They conclude that the solutions from the grid-based method are limited by the grid discretization, but in the cases where strong currents are considered the RRT methods have difficulties to find paths. Grid-based methods are widely used in path planning for AUVs. Pêtrès et al. (2007) proposed a new FMM for trajectory planning for AUVs that generates a continuous path and considers directional constraints. Lolla et al. (2014b) present a level set approach to minimize the travel time and determine the optimal trajectories, and apply the method in (Lolla et al., 2014a) to the trajectory planning for swarms of AUVs in the ocean with multiple obstacles. Eichhorn (2015) developed an enhanced A* algorithm to handle the time-varying environment in trajectory planning for AUVs. Note that in general, the optimality of the path is compromised by the grid space and time discretization. Another line of work has used model predictive control and optimal control (Morgan et al., 2014; Baumgartner et al., 2009; Zeng et al., 2014), and evolutionary algorithms (Alvarez et al., 2004). Risk-aware path planning methodologies have also been developed by considering the uncertainty in the ocean current predictions (Pereira et al., 2013; Wang et al., 2016). The trajectory planning of AUVs via mathematical programming methods has been less reported. Yilmaz et al. (2008) proposed a path planning method for AUVs based on an MILP formulation with a fixed grid for the space and limited angles of movement, but their model did not consider the impact of the ocean current on the vehicle. Abichandani et al. (2015) developed an MINLP model and a receding horizon for the multi-vehicle motion planning with acoustic connectivity constraints. However, these authors did not consider the ocean current in their model.

In this work, we propose a novel MINLP model for AUVs that considers obstacles avoidance and a nonlinear flow field to represent the ocean current. The incorporation of the nonlinear

flow in the MINLP model is a substantial improvement over other mathematical programming models for trajectory planning for AUVs. Overall, the MINLP model provides a versatile approach, which accommodates discrete binary variables and continuous functions accounting for nonlinear dynamics, motion constraints, and advection by the current. However, difficulties may arise due to nonconvexities and combinatorial complexity. A compromise may thus be sought between the details captured by MINLP models, the resulting complexity, and the capability to solve them.

Several algorithms, commercial and open-source solvers are available for the solution of MINLP problems. These solvers include 1) deterministic global optimization solvers; 2) random-search based solvers; and 3) deterministic local solvers. The first solver type is relatively recent compared with the other two; examples include α -BB (Androulakis et al., 1995), ANTIGONE (Misener and Floudas, 2014), BARON (Sahinidis, 2015; Tawarmalani and Sahinidis, 2005), LINDOGLOBAL (Lindo Systems, 2017), and Couenne (Belotti et al., 2009). The performance and capabilities of these solvers have increased significantly in the last years, due to advances in global optimization theory, for example in building tighter relaxations. These deterministic global solvers can solve to global optimality several small MINLP problems, but for larger problems, they require some computational time to explore the spatial branch and bound tree to prove global optimality. Furthermore, these solvers may be limited to situations in which rigorous relaxation problems can be built. Random search based solvers, such as genetic algorithms (Mitchell, 1996) or simulated annealing (Cardoso et al., 1997), can handle discontinuous and non-differential functions, and these solvers easily handle black box models. These solvers can be efficient to find good solutions mainly when the simulation step is relatively fast, or when specific approaches are used to optimize the solution procedure to handle the constraints (Lima et al., 2006). This type of solver does not provide a global optimality certificate. In the third type, solvers are based on MINLP algorithms (Leyffer, 2001; Duran and Grossmann, 1986; Westerlund and Porn, 2002; GAMS Development Corporation, 2017) which can guarantee global optimality for convex MINLP problems, but not for nonconvex problems. Examples of these solvers are DICOPT (Viswanathan and Grossmann, 1990), α -ECP (Westerlund and Porn, 2002; Westerlund and Lastusilta, 2017), SBB (GAMS Development Corporation, 2017), MINOPT (Schweiger and Floudas, 1998), and BONMIN (Bonami et al., 2008). However, in general, these solvers are relatively faster than the deterministic global optimization solvers, and they do not have the limitations that deterministic global optimization solvers have in handling particular nonlinear functions. Local MINLP solvers may depend on others solvers to tackle NLP and MILP subproblems, and therefore may be affected by the performance of these auxiliary solvers. We use the local solver DICOPT to solve the MINLP problems, and we implement a specific solution approach to overcome initialization difficulties and convergence to local solutions.

Specifically, we explore the problem of minimum-time trajectory planning for AUVs defined by MINLP models, considering the vehicle dynamics, collision avoidance, kinematic constraints and nonlinear flow fields.

The problem statement for minimum-time trajectory planning of AUV in the flow field is

specified in Section 2, which also formulates the MINLP and MILP models. The discussion provides a detailed outline of the MINLP problem, including AUV equations of motion, kinematic and obstacle avoidance constraints, and destination arrival check. Section 3 introduces a novel solution strategy, combining diversified initialization using waypoints, different solution methods, and parallel strategies. In Section 4, the computational efficiency of the solution approach is demonstrated through idealized case studies. Section 5 summarizes the major conclusions.

2. Formulation

We consider the optimal motion planning of an AUV represented by a point mass vehicle (Goerzen et al., 2010; Latombe, 1991). Starting from fixed initial position, velocity, and acceleration, the objective is to minimize the time to reach a fixed destination, subject to: a) limits on the speed and acceleration of the vehicle; b) transport by the current; and c) avoidance of obstacles. We assume that the velocity field of the current is known and steady, and that the position and size of the obstacles are stationary and given.

The optimal motion planning problem is accordingly expressed as

$$\begin{aligned}
 \min \quad & \int_{t_0}^{t_f} 1 \, dt = t_f - t_0 & (1) \\
 \text{subject to:} \quad & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{v}(t), & (2) \\
 & \ddot{\mathbf{x}}(t) = \nabla \mathbf{f}(\mathbf{x}(t)) \cdot \dot{\mathbf{x}}(t) + \mathbf{a}(t), & (3) \\
 & \mathbf{x}(t_0) = \mathbf{x}_0, & (4) \\
 & \mathbf{x}(t_f) = \mathbf{x}_f, & (5) \\
 & 0 \leq \|\mathbf{v}(t)\| \leq \bar{v}, & (6) \\
 & 0 \leq \|\mathbf{a}(t)\| \leq \bar{a}, & (7) \\
 & \mathbf{x}(t) \in \mathcal{X}, & (8) \\
 & \mathbf{x}(t), \mathbf{v}(t), \mathbf{a}(t) \in \mathbb{R}^2, & (9)
 \end{aligned}$$

where $\mathbf{x}(t)$ defines the position of the vehicle in the two-dimension Euclidean space, $\mathbf{f}(\mathbf{x}(t))$ is the velocity of the current at $\mathbf{x}(t)$, $\mathbf{v}(t)$ and $\mathbf{a}(t)$ are the velocity and acceleration of the vehicle relative to the current, respectively, t_f is the time to travel from the initial position $\mathbf{x}_0$ to the destination $\mathbf{x}_f$, whereas $\bar{v}$ and $\bar{a}$ are the maximum velocity and acceleration of the vehicle. In component form, we denote $\mathbf{x} = (x, y)$, $\mathbf{v} = (u, w)$, and $\mathbf{a} = (\dot{u}, \dot{w})$.

The two first constraints are the kinematic equations that describe the motion of the vehicle, and the constraint $\mathbf{x}(t) \in \mathcal{X}$ enforces that the trajectory does not intersect any obstacle. The remaining constraints define the initial position of the vehicle at the initial time, t_0 , and the final position of the vehicle at time t_f , and the bounds on the velocity and acceleration.

Below, we reformulate the problem (1) to (9) as an MINLP model that we denote by detailed MINLP problem. However, some preliminary experiments revealed that the complexity of this problem poses some difficulties to solvers to obtain feasible solutions. Therefore, we propose a solution method that relies on the solution of a set of approximated models of decreasing

complexity to initialize the detailed MINLP problem. We start by describing the specific features of the models, and then discuss the solution approach.

2.1. Detailed MINLP model

In this section, we describe the derivation of the detailed MINLP model. Let $I = \{0, \dots, N_t - 1\}$ be a set of indices. We consider a discretization of the time interval $T = \{t_i; i \in I\}$ ordered such that $t_0 \leq t_1 \leq \dots \leq t_{N_t-1}$. For all time dependent variables $\gamma(t)$, we denote by γ_i the quantity $\gamma(t_i)$ for all $i \in I$. We also define the set of indices $J = \{0, 1, \dots, N_t - 2\}$ and the time steps $\Delta t_i = t_{i+1} - t_i$ for $i \in J$.

Given a sufficiently small time step Δt_i , the trajectory $\mathbf{x}(t)$ is approximated using a second-order Taylor series at t_i yielding

$$\begin{aligned} \mathbf{x}(t_{i+1}) &\approx \mathbf{x}(t_i) + \dot{\mathbf{x}}(t_i)\Delta t_i + \ddot{\mathbf{x}}(t_i)\Delta t_i^2/2, \\ &= \mathbf{x}(t_i) + (\mathbf{f}(\mathbf{x}(t_i)) + \mathbf{v}(t_i))\Delta t_i + (\nabla \mathbf{f}(\mathbf{x}(t_i)) \cdot (\mathbf{f}(\mathbf{x}(t_i)) + \mathbf{v}(t_i)) + \mathbf{a}(t_i))\Delta t_i^2/2, \end{aligned} \quad (10)$$

$$\mathbf{v}(t_{i+1}) \approx \mathbf{v}(t_i) + \mathbf{a}(t_i)\Delta t_i. \quad (11)$$

In the rest of this work, we assume that the time step is constant, i.e. $\Delta t_0 = \dots = \Delta t_{N_t-1} = \Delta t$, this discretization results in the following set of equations

$$\mathbf{x}_{i+1} = \mathbf{x}_i + (\mathbf{f}(\mathbf{x}_i) + \mathbf{v}_i)\Delta t + (\nabla \mathbf{f}(\mathbf{x}_i) \cdot (\mathbf{f}(\mathbf{x}_i) + \mathbf{v}_i) + \mathbf{a}_i)\Delta t^2/2, \quad \forall i \in J, \quad (12)$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \mathbf{a}_i\Delta t, \quad \forall i \in J, \quad (13)$$

where $\mathbf{x}_i$, $\mathbf{v}_i$, $\mathbf{a}_i$ respectively denote the position, relative velocity and relative acceleration of the vehicle for $i \in I$.

The objective function minimizes the time that the vehicle needs to arrive to the destination. As in (1), the objective function is defined as

$$\min t_f. \quad (14)$$

However, because the time is discretized, the final time is one of the possible discrete values:

$$[t_f = \Delta t_0] \vee [t_f = \Delta t_0 + \Delta t_1] \vee \dots \vee \left[t_f = \sum_{i \in J} \Delta t_i \right]. \quad (15)$$

The proposition in (15) represents a disjunctive set (Vecchiotti et al., 2003) with a set of equality constraints separated by the operator OR ($\vee$). In the following derivation of the models, we also use the additional notation: $\wedge$ for AND, $\neg$ for NOT, and $\Rightarrow$ for implication. We define the arrival of the vehicle to the destination when the vehicle is inside of a box with coordinates $x_{\min}^F, x_{\max}^F, y_{\min}^F, y_{\max}^F$:

$$\begin{aligned} x_{\min}^F \leq x_i \leq x_{\max}^F \text{ and } y_{\min}^F \leq y_i \leq y_{\max}^F \\ \Rightarrow \text{the vehicle arrived to the destination, } \forall i \in I. \end{aligned} \quad (16)$$

The propositions in (15) and (16) are modeled using the disjunctions in (17) and (18), respectively, using a Boolean variable B_i , which is True if the vehicle is within the region defined in (16).

$$\bigvee_{j \in J} \left[t_f = \sum_{i=0}^j \Delta t_i \right], \quad (17)$$

$$B_i \in \{\text{True}, \text{False}\}, \quad \forall i \in I,$$

$$\begin{bmatrix} B_i \\ x_i \geq x_{\min}^F \\ x_i \leq x_{\max}^F \\ y_i \geq y_{\min}^F \\ y_i \leq y_{\max}^F \end{bmatrix} \vee \begin{bmatrix} \neg B_i \end{bmatrix}, \quad \forall i \in I. \quad (18)$$

We reformulate the disjunction in (18) using a big-M reformulation (Wolsey, 1998, p. 11; Vecchietti et al., 2003), leading to

$$-x_i \leq -x_{\min}^F + M_i^{1,1}(1 - b_i), \quad \forall i \in I, \quad (19)$$

$$x_i \leq x_{\max}^F + M_i^{1,2}(1 - b_i), \quad \forall i \in I, \quad (20)$$

$$-y_i \leq -y_{\min}^F + M_i^{1,3}(1 - b_i), \quad \forall i \in I, \quad (21)$$

$$y_i \leq y_{\max}^F + M_i^{1,4}(1 - b_i), \quad \forall i \in I, \quad (22)$$

where $M_i^{1,1} = \max\{-x_i + x_{\min}^F : x_{\min} \leq x_i \leq x_{\max}\}$, $M_i^{1,2} = \max\{x_i - x_{\max}^F : x_{\min} \leq x_i \leq x_{\max}\}$, $M_i^{1,3} = \max\{-y_i + y_{\min}^F : y_{\min} \leq y_i \leq y_{\max}\}$, $M_i^{1,4} = \max\{y_i - y_{\max}^F : y_{\min} \leq y_i \leq y_{\max}\}$, and b_i is a binary variable that is equal to 1 if the vehicle is in the target area at time t_i , and zero otherwise. The disjunction in (17) is reformulated using a convex-hull reformulation (Balas, 1985) (see Appendix A), leading to

$$t_f = \sum_{j \in J} [(j+1)\Delta t b_{j+1}], \quad (23)$$

$$\sum_{i \in I} b_i = 1. \quad (24)$$

The obstacles are represented as squares or rectangles through the coordinates of their lower left and upper right corner points: $(\underline{x}_o, \underline{y}_o)$ and $(\bar{x}_o, \bar{y}_o)$. The regions within the obstacles are considered as infeasible regions of the space. To enforce that the position of the vehicle is always in the feasible region, we define the following logical propositions over each obstacle and time grid point:

$$x_i \leq \underline{x}_o \vee \underline{x}_o \leq x_i \leq \bar{x}_o \vee x_i \geq \bar{x}_o, \quad \forall o \in O, \forall i \in I, \quad (25)$$

$$y_i \leq \underline{y}_o \vee \underline{y}_o \leq y_i \leq \bar{y}_o \vee y_i \geq \bar{y}_o, \quad \forall o \in O, \forall i \in I, \quad (26)$$

$$\underline{x}_o \leq x_i \leq \bar{x}_o \Rightarrow y_i \leq \underline{y}_o \text{ or } y_i \geq \bar{y}_o, \quad \forall o \in O, \forall i \in I, \quad (27)$$

$$\underline{y}_o \leq y_i \leq \bar{y}_o \Rightarrow x_i \leq \underline{x}_o \text{ or } x_i \geq \bar{x}_o, \quad \forall o \in O, \forall i \in I. \quad (28)$$

These logical propositions can be translated into the disjunctions

$$\left[\begin{array}{c} Z_{o,i}^1 \\ x_i \leq \underline{x}_o \end{array} \right] \vee \left[\begin{array}{c} Z_{o,i}^2 \\ \underline{x}_o \leq x_i \leq \bar{x}_o \end{array} \right] \vee \left[\begin{array}{c} Z_{o,i}^3 \\ x_i \geq \bar{x}_o \end{array} \right], \quad \forall o \in O, \forall i \in I, \quad (29)$$

$$\left[\begin{array}{c} Z_{o,i}^4 \\ y_i \leq \underline{y}_o \end{array} \right] \vee \left[\begin{array}{c} Z_{o,i}^5 \\ \underline{y}_o \leq y_i \leq \bar{y}_o \end{array} \right] \vee \left[\begin{array}{c} Z_{o,i}^6 \\ y_i \geq \bar{y}_o \end{array} \right], \quad \forall o \in O, \forall i \in I, \quad (30)$$

$$Z_{o,i}^1, Z_{o,i}^2, Z_{o,i}^3, Z_{o,i}^4, Z_{o,i}^5, Z_{o,i}^6 \in \{\text{True}, \text{False}\}, \quad \forall o \in O, \forall i \in I, \quad (31)$$

$$Z_{o,i}^2 \Rightarrow Z_{o,i}^4 \vee Z_{o,i}^6, \quad \forall o \in O, \forall i \in I, \quad (32)$$

$$Z_{o,i}^5 \Rightarrow Z_{o,i}^1 \vee Z_{o,i}^3, \quad \forall o \in O, \forall i \in I, \quad (33)$$

where $Z_{o,i}^1, \dots, Z_{o,i}^6$ are Boolean variables that establish whether a given term in a disjunction is True. The disjunctions in (29) and (30) and the logical propositions in (32) and (33) are reformulated as (see Appendix B):

$$x_i \leq \underline{x}_o + M_{o,i}^{2,1}(1 - z_{o,i}^1), \quad \forall o \in O, \forall i \in I, \quad (34)$$

$$-x_i \leq -\bar{x}_o + M_{o,i}^{2,2}(1 - z_{o,i}^3), \quad \forall o \in O, \forall i \in I, \quad (35)$$

$$y_i \leq \underline{y}_o + M_{o,i}^{2,3}(1 - z_{o,i}^4), \quad \forall o \in O, \forall i \in I, \quad (36)$$

$$-y_i \leq -\bar{y}_o + M_{o,i}^{2,4}(1 - z_{o,i}^6), \quad \forall o \in O, \forall i \in I, \quad (37)$$

$$z_{o,i}^1 + z_{o,i}^3 + z_{o,i}^4 + z_{o,i}^6 \geq 1, \quad \forall o \in O, \forall i \in I, \quad (38)$$

where $M_{o,i}^{2,1} = \max\{x_i - \underline{x}_o : x_{\min} \leq x_i \leq x_{\max}\}$, $M_{o,i}^{2,2} = \max\{-x_i + \bar{x}_o : x_{\min} \leq x_i \leq x_{\max}\}$, $M_{o,i}^{2,3} = \max\{y_i - \underline{y}_o : y_{\min} \leq y_i \leq y_{\max}\}$, $M_{o,i}^{2,4} = \max\{-y_i + \bar{y}_o : y_{\min} \leq y_i \leq y_{\max}\}$, and $z_{o,i}^1, z_{o,i}^3, z_{o,i}^4$, and $z_{o,i}^6$ are binary variables associated with the Boolean variables in (29) and (30). The constraints that force the path of the vehicle to be outside of the obstacles are only enforced at the time grid points defined, which may lead to solutions where the trajectory cuts the corner of an obstacle, or the trajectory goes over an obstacle. To avoid these situations, we add a safety margin to the obstacles defined as $S = \bar{v}\Delta t/2$ (Radmanesh and Kumar, 2016). Note that more elaborate approaches have been developed for inter-sample avoidance in Earl and D'Andrea (2005), Maia and Galvão (2009), Richards and Turnbull (2015), and Afonso et al. (2016).

The upper bounds on the velocity and acceleration are enforced by the following constraints:

$$\|\mathbf{v}_i\| \leq \bar{v}, \quad \forall i \in I, \quad (39)$$

$$\|\mathbf{a}_i\| \leq \bar{a}, \quad \forall i \in I. \quad (40)$$

Finally, combining the constraints described above, the trajectory optimization problem is recast as the following MINLP problem:

$$\begin{aligned} \min \quad & \text{time to arrive to the destination target (23)} \\ \text{s.t.} \quad & \text{arrival check (20) to (21), (24),} \\ & \text{discrete-time kinematics(12), (13),} \\ & \text{obstacle avoidance (34) to (38),} \\ & \text{initial conditions (4),} \\ & \text{upper bounds on velocity and acceleration (MINLP}_D\text{)} \\ & \text{magnitude (39) and (40),} \\ & \mathbf{x}_{\min} \leq \mathbf{x}_i \leq \mathbf{x}_{\max}, \\ & \mathbf{x}_i, \mathbf{v}_i, \mathbf{a}_i \in \mathbb{R}^2, \quad \forall i \in I, \\ & z_{o,i}^1, z_{o,i}^3, z_{o,i}^4, z_{o,i}^6 \in \{0, 1\}, \quad \forall o \in O, \forall i \in I, \\ & b_i \in \{0, 1\}, \quad \forall i \in I. \end{aligned}$$

In the model above, the nonlinearities are present in the constraints on the maximum velocity and acceleration of the vehicle, and in the discrete kinematic constraints involving the flow field. The constraints on the maximum velocity and acceleration of the vehicle are convex, but the discrete kinematic constraints are non-convex, resulting in a non-convex MINLP problem.

2.2. Simplified MINLP models

2.2.1. Without current acceleration

Based on the problem (MINLP_D), we propose a simplified model by neglecting the term of the current acceleration in (12). This simplification leads to:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + (\mathbf{f}(\mathbf{x}_i) + \mathbf{v}_i) \Delta t + \mathbf{a}_i \Delta t^2 / 2, \quad \forall i \in J, \quad (41)$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \mathbf{a}_i \Delta t, \quad \forall i \in J. \quad (42)$$

Accordingly, the MINLP model without considering the current acceleration is stated as

$$\begin{aligned}
 \min \quad & \text{time to arrive to the destination target (23)} \\
 \text{s.t.} \quad & \text{arrival check (20) to (21), (24),} \\
 & \text{discrete-time kinematics(41), (42),} \\
 & \text{obstacle avoidance (34) to (38),} \\
 & \text{initial conditions (4),} \\
 & \text{upper bounds on velocity and acceleration (MINLP}_C\text{)} \\
 & \text{magnitude (39) and (40),} \\
 & \mathbf{x}_{\min} \leq \mathbf{x}_i \leq \mathbf{x}_{\max}, \\
 & \mathbf{x}_i, \mathbf{v}_i, \mathbf{a}_i \in \mathbb{R}^2, \quad \forall i \in I, \\
 & z_{o,i}^1, z_{o,i}^3, z_{o,i}^4, z_{o,i}^6 \in \{0, 1\}, \quad \forall o \in O, \forall i \in I, \\
 & b_i \in \{0, 1\}, \quad \forall i \in I.
 \end{aligned}$$

2.2.2. Without current

By neglecting the current velocity in (12), the discrete-time kinematics simplify to

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{v}_i \Delta t + \mathbf{a}_i \Delta t^2 / 2, \quad \forall i \in J, \quad (43)$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \mathbf{a}_i \Delta t, \quad \forall i \in J. \quad (44)$$

In this MINLP model, we consider an additional feature that allows us to fix waypoints between the starting point and the destination. The waypoints are easy to model and they are based on the constraints of the arrival check with some minor adjustments, specifically

$$x_i \leq x_{\max}^w + M_i^{3,1}(1 - s_i), \quad \forall i \in I, \quad (45)$$

$$-x_i \leq -x_{\min}^w + M_i^{3,2}(1 - s_i), \quad \forall i \in I, \quad (46)$$

$$y_i \leq y_{\max}^w + M_i^{3,3}(1 - s_i), \quad \forall i \in I, \quad (47)$$

$$-y_i \leq -y_{\min}^w + M_i^{3,4}(1 - s_i), \quad \forall i \in I, \quad (48)$$

$$t_{wp} = \sum_{j \in J} [(j+1)\Delta t s_{j+1}], \quad (49)$$

$$\sum_{i \in I} s_i = 1, \quad (50)$$

$$t_{wp} \leq \sum_{j \in J} [(j+1)\Delta t b_{j+1}], \quad (51)$$

where $M_i^{3,1} = \max\{x_i - x_{\max}^w : x_{\min} \leq x_i \leq x_{\max}\}$, $M_i^{3,2} = \max\{-x_i + x_{\min}^w : x_{\min} \leq x_i \leq x_{\max}\}$, $M_i^{3,3} = \max\{y_i - y_{\max}^w : y_{\min} \leq y_i \leq y_{\max}\}$, $M_i^{3,4} = \max\{-y_i + y_{\min}^w : y_{\min} \leq y_i \leq y_{\max}\}$, s_i is a binary variable that is equal to 1 if the vehicle is in the area defined by the coordinates of the waypoint, $x_{\max}^w$, $x_{\min}^w$, $y_{\max}^w$, $y_{\min}^w$ at time t_i and zero otherwise, t_{wp} is the time from the starting

point to the waypoint that must be visited before the vehicle arrives to the destination. The present model is stated as

$$\begin{aligned}
 \min \quad & \text{time to arrive to the destination target (23)} \\
 s.t. \quad & \text{arrival check (20) to (21), (24),} \\
 & \text{discrete-time kinematics(43) to (41),} \\
 & \text{obstacle avoidance (34) to (38),} \\
 & \text{waypoint (45) to (51),} \\
 & \text{initial conditions (4),} \\
 & \text{upper bounds on velocity and acceleration} \\
 & \text{magnitude (39) and (40),} \\
 & \mathbf{x}_{\min} \leq \mathbf{x}_i \leq \mathbf{x}_{\max}, \\
 & \mathbf{x}_i, \mathbf{v}_i, \mathbf{a}_i \in \mathbb{R}^2, \quad \forall i \in I, \\
 & z_{o,i}^1, z_{o,i}^3, z_{o,i}^4, z_{o,i}^6 \in \{0, 1\}, \quad \forall o \in O, \forall i \in I, \\
 & b_i, s_i \in \{0, 1\}, \quad \forall i \in I.
 \end{aligned} \tag{MINLP_S}$$

2.3. Approximate MILP model

We consider a further simplified model that does not consider the current and approximates the nonlinear terms in (39) and (40) by linear expressions. Neglecting the current means that this model will provide solutions with paths free of obstacles, but the overall solution (not the path) are not feasible in the MINLP models with current. However, the solution of the MILP model can be useful for initialization purposes, taking advantage of the fast computational times of the MILP model by comparison to the MINLP models.

In the MILP model, the nonlinear term that evaluates the norm of the velocity vector is approximated by the projection of the vector of the velocity into a set of vectors with angle θ_k , as proposed in Richards and How (2002). Thus, we approximate $[(u_i)^2 + (w_i)^2]^{0.5} \approx \cos(\theta_k)u_i + \sin(\theta_k)w_i$, and we enforce that the projections for all θ_k are less or equal than the maximum velocity:

$$\cos(\theta_k)u_i + \sin(\theta_k)w_i \leq \bar{v}, \quad \forall k \in K, \forall i \in I. \tag{52}$$

The other nonlinear term in the MINLP models involves the constraint on the upper bound on the acceleration of the vehicle in (40). This bound on the norm of the acceleration is also approximated by considering the projection of the acceleration vector into a set of vectors with angle $\alpha_k, \forall k \in K$, as we presented for the velocity. Therefore, $[(\dot{u}_i)^2 + (\dot{w}_i)^2]^{0.5}$ is approximated by $\cos(\alpha_k)\dot{u}_i + \sin(\alpha_k)\dot{w}_i$, and the upper bound on the projection is enforced on a set of discretized α_i ,

$$\cos(\alpha_k)\dot{u}_i + \sin(\alpha_k)\dot{w}_i \leq \bar{a}, \quad \forall k \in K, \forall i \in I. \tag{53}$$

An advantage of these two approximations is that they only introduce linear inequalities in the MILP, without adding new binary variables.

With the option to consider waypoints, the MILP model is formulated as

$$\begin{aligned}
 \min \quad & \text{time to arrive to the destination target (23)} \\
 \text{s.t.} \quad & \text{arrival check (20) to (21), (24),} \\
 & \text{discrete-time kinematics(43) to (44),} \\
 & \text{obstacle avoidance(34) to (38),} \\
 & \text{waypoint (45) to (51),} \\
 & \text{initial conditions (4),} \\
 & \text{upper bounds on velocity and acceleration} \\
 & \text{magnitude (52) and (53),} \\
 & \mathbf{x}_{\min} \leq \mathbf{x}_i \leq \mathbf{x}_{\max}, \\
 & \mathbf{x}_i, \mathbf{v}_i, \mathbf{a}_i \in \mathbb{R}^2, \quad \forall i \in I, \\
 & z_{o,i}^1, z_{o,i}^3, z_{o,i}^4, z_{o,i}^6 \in \{0, 1\}, \quad \forall o \in O, \forall i \in I, \\
 & b_i, s_i \in \{0, 1\}, \quad \forall i \in I.
 \end{aligned} \tag{MILP}$$

3. Solution approach

MINLP models are a versatile approach to capture complex phenomena and decisions of real problems because they accommodate nonlinear functions and decisions involving binary variables. However, they require a special attention in the problem formulation, and they may pose some difficulties to MINLP solvers due to their intrinsic characteristics such as nonconvexities and combinatorial complexity. Therefore, there is a trade-off between the complexity and the details captured by MINLP models and the capability to solve them.

In general, MINLP solvers rely on the solution of two types of NLP problems: 1) a relaxed MINLP problem where the binary variables are treated as continuous variables between 0 and 1; and 2) an MINLP problem with the binary variables fixed. Therefore, the solution of MINLP problems relies on the solution of NLP problems, which are known to be sensitive to the initialization used. The initialization has an impact on finding an initial feasible solution, but also on the final solution returned by the solver. This dependence on the solution of NLP problems and initialization is present in solvers based on outer-approximation or branch & bound, such as DICOPT or SBB. Therefore, because the solution procedure of MINLP problems relies on the solution of NLP problems, initialization is paramount to obtain a solution of the MINLP problems.

The two solvers mentioned above can only guarantee the global optimum for convex MINLP problems. For nonconvex MINLP problems, the outer-approximations used in DICOPT may cut-off the global optimum; in SBB, failing to get a solution of the relaxed MINLP problem in a node may also prune the global optimum.

To overcome these issues, we propose a solution approach involving strategies for initialization and improvement of the solutions.

3.1. Initialization

In our solution approach, we implement three initialization strategies: 1) a sequential solution procedure of problems of increasing detail; 2) the generation of multiple starting solutions; and 3) an iterative procedure involving two NLP solvers to solve the initial relaxed MINLP problem.

The first strategy is based on the sequential solution of the problems presented in Section 2. This strategy starts with the solution of the MILP problem presented in 2.3, which provides a solution with a path that does not intersect any obstacle and minimizes the trajectory time in a short computational time. The solution sequence continues as illustrated in Figure 1. The main drawback of this solution sequence is that the two first problems do not consider the current. Therefore, the neighborhood region along the path resulting from the MINLP model without current may be an infeasible region or a region with low-quality solutions in the MINLP problem with current.

The second strategy aims at mitigation of the drawback above by forcing the MILP problem and MINLP problem without current to find alternative paths by enforcing the vehicle to pass by a waypoint. The independent solution of a set of sequences of MILP and MINLP without current, wherein each set a different waypoint is specified, provides a collection of different paths. These different paths are then used to initialize the models with current. Note that the MINLP problems with current do not consider the waypoints.

The third strategy addresses the initialization of the nonlinear solvers. Before calling DICOPT to solve one of the MINLP problems, we ensure that the relaxed MINLP provides a feasible solution by employing an iterative procedure involving the solvers IPOPT (Wächter and Biegler, 2006) and CONOPT (Drud, 1994), see Algorithm 1. This procedure is integrated into the proposed sequential solution and it is used before the solution of each MINLP problem.

Algorithm 1 The algorithm to solve RMINLP

```

1: STOP ← 0
2: while STOP < 0 do                                     ▷ Or until it reaches the maximum time
3:   {Solve RMINLP using IPOPT}
4:   if ModelStat ≤ 2 Or ModelStat = 7 then                 ▷ Solution achieved
5:     STOP ← 1
6:   else
7:     {Solve RMINLP using CONOPT}
8:     if ModelStat ≤ 2 Or ModelStat = 7 then                 ▷ Solution achieved
9:       STOP ← 1
10:  end if
11: end if
12: end while
13: return all of the variables
14: {ModelStat=1, optimal solution achieved; ModelStat=2, local optimal solution achieved;
    ModelStat=7, solver terminated early and model was feasible but not yet optimal}

```

3.2. Diversification

The strategies proposed in the previous section increase the robustness of our solution approach by helping to find feasible solutions. The benefits of using waypoints to search for alternative paths are twofold: 1) the set of solutions obtained with the waypoints help with the initialization of the relaxed MINLP problems, since one path obtained with a waypoint may not be a good starting point for the models with current, but the solution from another waypoint may be a successful starting point; and 2) the solution obtained with one waypoint may lead to a local solution, while another waypoint may help to find a better solution. Therefore, waypoints play also an important role in assessing the quality of the local solutions obtained.

In addition, we propose an efficient strategy to improve the solution obtained with the MINLP model with current. Given that we have a solution from the MINLP model with current with $t_f = t_f^*$, we solve the same problem but we enforce the constraint

$$t_f \leq t_f^* - n\Delta t, \quad (54)$$

where n is a fixed positive integer number. This new constraint forces the solver to find a better solution with $t_f < t_f^*$. We implement this strategy by solving first the MINLP model with current and then by solving in parallel a set of the MINLP problems with current for different values of n . Note that the problems solved in parallel use the solution of the first problem as starting solution, which is infeasible only for the new constraint added. This strategy is oriented to find better solutions than the first solution obtained, which may compensate for the local convergence of DICOPT.

3.3. Parallel computing

Figure 1 presents a diagram of the solution approach described in the two previous sections. In this diagram, each column corresponds to the sequential solution of the MILP and MINLP problems for a given waypoint. Each sequential solution is independent and performed in parallel. Within each column, there is an additional parallelization of the solution of the MINLP problems including the constraint in (54). The best solution from the MINLP problems with current solved in parallel is used to initialize the detailed MINLP problem within the same solution sequence. The overall solution approach uses two levels of parallelization, the first represented by the main columns in the diagram, and a second level represented by the parallel blocks with the solution of the MINLP problem with current.

3.4. Specific probing techniques

The number of binary variables in the proposed models depends on the number of time steps postulated and obstacles considered. Given that the starting and ending points and obstacles are known, we can use specific probing techniques to fix some binary variables to 0 or 1, and therefore, reduce the initial size of the problem.

Regarding the binary variables b_i , which indicate if the vehicle reaches the destination in the time period i , some b_i can be fixed to zero by calculating a lower bound on the travel time

Figure 1: Proposed parallel solution approach. Each column represents a solution sequence of the problems MILP, MINLP_S, MINLP_C, and MINLP_D, which is solved independently in parallel. Within each column there is a set of MINLP_C problems that are also solved independently in parallel.

between the starting point and destination target. The lower bound is calculated considering that the path of the vehicle is a straight line between the starting point and destination using the maximum velocity of the vehicle and the current. Based on this lower bound, some binary variables b_i can be fixed to zero.

A similar rationale is applied to the binary variables $z_{o,i}^1$, $z_{o,i}^3$, $z_{o,i}^4$, and $z_{o,i}^6$, which are used to avoid obstacles. In this case, we calculate a lower bound on the travel time of a straight line between the starting point and each obstacle, and this bound is applied to fix specific binary variables.

To infer the value of the specific probing techniques described, we present in Table 1 a set of results from four MILP problems that are described and further analyzed in Section 4. Table 1 shows the optimal values of the linear relaxations of the MILP problems and linear relaxations at the beginning of the root node for two cases: 1) without using the specific probing techniques; and 2) using the specific probing techniques. The results in Table 1 show that the optimal value of the LP relaxation increases from 1.97 to 16.04 in Problem 1, due to the specific probing techniques. However, there is only a small positive increment in the optimal value of the LP relaxation at the root node when the specific probing techniques are used with the MILP solver. For example, from 16.91 to 17.04 for Problem 1. These results indicate that the specific probing techniques are efficient in tightening the linear relaxation, but also that the pre-processing techniques applied by the MILP solver before the root node step are as efficient as

Table 1: Optimal values of the relaxations. Comparison between using specific probing techniques and do not using.

Problem	LP relaxation		LP relaxation at the root node after pre-processing		Optimal value of the MILP
	Without SP	With SP	Without SP	With SP	
1	1.97	16.04	16.91	17.04	27.00
2	1.30	7.05	10.00	10.00	19.00
3	1.38	10.01	11.00	12.00	21.00
4	1.54	13.03	14.00	13.00	24.00

SP - Specific probing techniques. CPLEX 12.6.3 is used as the LP and MILP solvers.

the specific probing techniques tested. In the numerical results section, we analyze the impact of these specific probing techniques on the performance of the MILP solver.

4. Example Applications

In this section, the performance of the proposed solution approach is assessed for various idealized scenarios. First, we assess and discuss the computational performance of the MILP model using three case studies: Case 1, 2, and 3. We consider a fourth case to demonstrate the full capabilities of the MINLP model.

Case 1 involves a scenario without current and three obstacles, which is used to evaluate the effect of different time step sizes on the performance of the MILP model and trajectories. Case 2 considers a more complex setting without current comprising five obstacles and four scenarios. Each scenario includes four instances, where each instance is differentiated by the initial position or the target position. This case provides interesting insights regarding the computational performance of the formulations and the MILP solvers. Case 3 involves a setting with 7 obstacles and we consider one scenario with four instances. This case demonstrates that some MILP models have tight relaxations, and therefore, optimal solutions are obtained fast. Case 4 addresses a domain with three obstacles around which a realistic flow field is assumed. In this case, we consider two scenarios, labeled 1 and 2, in which the destination of the vehicle is respectively downstream and upstream of the initial position. In all cases, we set $\bar{v} = 1.5$ m/s and $\bar{a} = 1.5$ m/s². The maximum velocity corresponds to a medium-range or small-range AUV (Zeng et al., 2015). The coordinates of the obstacles for each case are presented in Table 2 and the coordinates of the initial and target positions are given in Table 3. Note that in Case 2, the initial positions of Scenario A are the targets in Scenario B, whereas the destination target in Scenario A is the starting position in Scenario B. A similar reasoning is applied to Scenarios C and D; see Table 3.

To generate a realistic velocity field in Case 4, we rely on a simulation of the transient incompressible Navier-Stokes (NS) equation

$$\frac{\partial \mathbf{f}}{\partial t} + (\mathbf{f} \cdot \nabla) \mathbf{f} - \nu \nabla^2 \mathbf{f} + \nabla p = 0,$$

Table 2: Coordinates of the obstacles.

	Obstacles	$\mathbf{x}_{\min}^o$	$\mathbf{x}_{\max}^o$
Case 1	1	(15, 24)	(25, 30)
	2	(40, 26)	(50, 35)
	3	(60, 10)	(85, 27)
Case 2	1	(-4, 3.5)	(-3, 6)
	2	(0, 1)	(1, 5)
	3	(2, 3)	(3, 9)
	4	(3, 2)	(10, 3)
	5	(3, 9)	(10, 10)
Case 3	1	(-1.75, -2.6)	(-0.4, -1.4)
	2	(0.4, -2.6)	(1.7, -1.4)
	3	(3.4, -3.5)	(4.6, -2.4)
	4	(2.5, -1.6)	(3.6, -0.5)
	5	(3.4, -0.4)	(4.6, 1.5)
	6	(-1.75, 1.3)	(-0.4, 2.7)
	7	(1.4, 1.3)	(2.8, 2.6)
Case 4	Same as Case 1		

Table 3: Coordinates of the initial and target positions for all cases and scenarios.

Cases/Scenarios	Initial position (x,y)	Target (x,y)
Case 1	(5, 25)	(90, 25)
Case 2		
Scenario A	1-(-20, 4)	(5, 5)
	2-(-7, 4)	(5, 5)
	3-(-10, 8)	(5, 5)
	4-(-15, 6)	(5, 5)
Scenario B	(5, 5)	1-(-20, 4)
	(5, 5)	2-(-7, 4)
	(5, 5)	3-(-10, 8)
	(5, 5)	4-(-15, 6)
Scenario C	1-(-50, 4)	(5, 5)
	2-(-37, 4)	(5, 5)
	3-(-40, 8)	(5, 5)
	4-(-45, 6)	(5, 5)
Scenario D	(5, 5)	1-(-50, 4)
	(5, 5)	2-(-37, 4)
	(5, 5)	3-(-40, 8)
	(5, 5)	4-(-45, 6)
Case 3	(-4, 0)	1-(6, 3)
	(-4, 0)	2-(6, 4)
	(-4, 0)	3-(6, 2)
	(-4, 0)	4-(6, -5)
Case 4		
Scenario A	(0, 25)	(90, 25)
Scenario B	(90, 25)	(45, 15)

$$\nabla \cdot \mathbf{f} = 0,$$

where $\mathbf{f}$ is the current velocity vector, p the pressure, and $\nu = 0.06$ the kinematic viscosity. The computational domain is illustrated in Fig. 2. No-slip boundary conditions are imposed on all

Figure 2: Schematic of the computational domain. The length and height are $L = 150$ m and $H = 50$ m, respectively. For the trajectory-planning problem, we only consider the 100-m long subdomain to the right of the dashed line.

solid surfaces. At the left boundary of the domain, the inlet profile,

$$\mathbf{f} = \begin{pmatrix} \frac{4y(y-H)}{H^2} \\ 0 \end{pmatrix}$$

is imposed, whereas outflow conditions are imposed at the right boundary. The NS is integrated starting from a state of rest using a mixed P2/P1 finite element method in space. An implicit-explicit scheme is used for this purpose, where implicit treatment of the diffusive term and explicit treatment of advection. We make an instantaneous snapshot of the flow available to the nonlinear solver within the MINLP solver by first projecting $\mathbf{f}$, $\nabla \mathbf{f}$ and $\nabla^2 \mathbf{f}$ on the functional space of P1 elements. Library functions are then used to return the values required by the nonlinear solver.

Note that the peak velocity associated with the inlet profile is greater than the vehicle's maximum speed. The mission of trajectory planning is much more delicate in such strong-current scenarios, because of the risk of the vehicle becoming trapped, or overwhelmed by the current.

The planning models are implemented in GAMS (Bruce A. McCarl et. al., 2013) and solved on a computer with a 2.8-GHz, Intel Xeon (R) CPU E5-2680 processor with 12 cores and 128GB RAM. The solvers used are CPLEX 12.6.3, GAMS/DICOPT 24.7.3, GAMS/CONOPT 3.17A and GAMS/IPOPT 3.12. The stopping criteria for each models are set to a maximum wall clock time of 36000 s, and a 0.0% gap of CPLEX.

4.1. Case 1

Time-optimal trajectories with different Δt are depicted in Fig. 3; also depicted are the velocity and acceleration profiles along the trajectory. With a large time step, $\Delta t = 3$ s, the

vehicle takes a longer path and requires 63 s to reach its destination. With a small time step, $\Delta t = 0.5$ s, the vehicle needs 59 s and follows a shorter path to reach its destination. Note that the vehicle maintains its maximum speed for most of the trajectory.

The constraints for obstacle avoidance are applied at the time grid points defined by the discretization; consequently, the collision-free motion between these grid points is not guaranteed, and this limitation is inherent in the discrete time approach. For a practical perspective, this issue is mitigated with the obstacles enlarged with a margin defined as $S = \bar{v}\Delta t/2$. As shown in Fig. 3, the difference in size of the time steps results in the difference in optimal paths. As Δt increases, the safety margins become larger, and the vehicle is forced to take a longer path than with a small time step. With $\Delta t = 0.5$ s, the optimal path is in close agreement with the one produced by the Dijkstra's algorithm (Dijkstra, 1959). With $\Delta t = 3$ s, the path cuts the corner of the enlarged obstacle but does not overlap with the obstacles. Note also that with $\Delta t = 3$ s, the path does not seem the shortest one. For example, between the time 22 s and 50 s, the path does not follow a straight line. This is explained by the discretization of the time. In this case with $\Delta t = 3$ s, the optimal t_f is 63 s, which means that any path that satisfies $t_f = 63$ s is an optimal path. Therefore, small adjustments on the path presented in Fig. 3 that do not lead to a better final time are redundant. Note that a better solution than $t_f = 63$ s is not feasible, for example to obtain $t_f = 60$ s, the path needs to be similar to the path obtained with $\Delta t = 1$ s, which is considerably different from the path obtained with $\Delta t = 3$ s and would violate the safety margins imposed by $\Delta t = 3$ s.

Table 4 summarizes the computational statistics of the MILP model with different time steps. It is clear that as the time step is reduced, the size of the model increases significantly. Specifically, with $\Delta t = 0.5$ s the model is nearly six times greater than with $\Delta t = 3$ s. Even though the size of the model is large, the MILP model can be solved very quickly, namely in less than 2 s.

Table 4: Case 1. Size of the MILP model and results for different time step sizes.

Δt	N_t	E#	V#	BV#	t_f (s)	T (s)
0.5	140	8862	3666	1833	59	2
1	70	4452	1846	923	60	2
3	23	1491	624	312	63	0.1

E-Equation, V- total variables, BV-Binary variables, T-Wall clock time.

Figure 3: Case 1 optimal paths and profiles of velocities and accelerations of the vehicle for $\Delta t = 0.5$ s, 1 s, and 3 s. The obstacles are represented by $\blacksquare$, the initial position and destination by $\square$, and the noncontinuous lines are the safety margins.

4.2. Case 2

In this case, we present and discuss the results for the four scenarios presented in Table 3; each scenario has four instances. Figures 4 and 5 show the optimal trajectories and travel times for the Scenarios A and B, respectively. The results show that the travel times increase as the distance between the initial position and target increases, and also, the travel times for the same pair of initial position and target are the same. For example, the travel time between the initial position 1 and the target in Scenario A is the same as the travel time from the initial position to the target 1 in Scenario B.

The optimal trajectories obtained for Scenarios C and D are presented in Figures 6 and 7, respectively. In Scenarios C and D, the positions 1, 2, 3, and 4 are moved further to the left of the obstacles, and therefore, the travel times increase. The travel times for the same pair of the initial starting point and target are also the same for Scenarios C and D. However, in these scenarios, there are two pairs of symmetric solutions with the same travel time. For example, the starting/target positions with the label 2 have the same travel time but lead to different trajectories in Scenarios C and D.

The size of the models and computational results for Scenarios A and B are presented in

Table 5. For each scenario, we present the computational time for the MILP model using the specific probing techniques described in Section 3.4 and without using these techniques. In the

Figure 4: Optimal paths (top) and velocity and acceleration profiles of a single vehicle for Case 2, Scenario A with 5 obstacles. The obstacles are represented by $\blacksquare$, and the destination by $\square$. The noncontinuous lines are the safety margins. Different paths correspond to different initial starting points, all with the same destination.

Figure 5: Optimal paths (top) and velocity and acceleration profiles of a single vehicle for Case 2, Scenario B with 5 obstacles. The obstacles are represented by $\blacksquare$, and the destination by $\square$. The noncontinuous lines are the safety margins. Different paths correspond to different destination targets, all with the same initial starting position.

two scenarios and the four instances, the number of equations, number of total variables, and number of binary variables is the same. The instances where the specific probing techniques are applied, the number of binary variables is reduced as a function of the distance between the initial position and the obstacles and the destination target. For Scenario A, the results show that using the specific probing techniques in one instance there is a degradation of the computational time, from 10 s to 12 s, whereas in the other three instances there is a reduction: from 23 s to 21 s, 24 s to 16 s, and 16 s to 10 s. For Scenario B, there are no significant

Figure 6: Optimal paths (top) and velocity and acceleration profiles of a single vehicle for Case 2, Scenario C with 5 obstacles. The obstacles are represented by ■, and the destination by □. The noncontinuous lines are the safety margins. Different paths correspond to different initial starting points, all with the same destination.

Table 5: Case 2, Scenarios A and B. Size and results of the MILP without using and with the specific probing techniques. $\Delta t = 0.2$ s and $N_t = 50$.

Initial				BV#		t_f (s)		T (s)	
position	Target	E#	V#	No SP	With SP	No SP/With SP	No SP	With SP	
1-(-20, 4)	(5, 5)	3702	2397	1071	923	27	10	12	
2-(-7, 4)	(5, 5)	3702	2397	1071	1016	19	23	21	
3-(-10, 8)	(5, 5)	3702	2397	1071	991	21	24	16	
4-(-15, 6)	(5, 5)	3702	2397	1071	960	24	16	10	
(5, 5)	1-(-20, 4)	3702	2397	1071	1039	27	2	2	
(5, 5)	2-(-7, 4)	3702	2397	1071	1048	19	2	2	
(5, 5)	3-(-10, 8)	3702	2397	1071	1045	21	2	1	
(5, 5)	4-(-15, 6)	3702	2397	1071	1042	24	1	1	

E - Equation, V - Total variables, BV - Binary variables, No SP - no specific probing techniques, With SP - with specific probing techniques, T-Wall clock time.

improvements by using the specific probing techniques. Overall, the specific probing techniques have a positive impact on the instances that require larger computational times. Comparing Scenarios A and B, the computational time required by Scenario B is one order of magnitude faster than in Scenario A.

Table 6 presents the size of the models and computational results for Scenarios C and D. Scenarios C and D require less computational time than Scenario A, and Scenario D is faster

Figure 7: Optimal paths (top) and velocity and acceleration profiles of a single vehicle for Case 2, Scenario D with 5 obstacles. The obstacles are represented by ■, and the destination by □. The noncontinuous lines are the safety margins. Different paths correspond to different destination targets, all with the same initial starting position.

Table 6: Case 2, Scenarios C and D. Size and results of the MILP without and with the specific probing techniques. $\Delta t = 0.2$ s and $N_t = 50$.

Initial				BV#		t_f (s)	T (s)	
position	Target	E#	V#	No SP	With SP	No SP/With SP	No SP	With SP
1-(-50, 4)	(5, 5)	3702	2397	1071	703	48	2	1
2-(-37, 4)	(5, 5)	3702	2397	1071	796	39	7	6
3-(-40, 8)	(5, 5)	3702	2397	1071	772	41	3	3
4-(-45, 6)	(5, 5)	3702	2397	1071	740	44	3	3
(5, 5)	1-(-50, 4)	3702	2397	1071	1019	48	1	1
(5, 5)	2-(-37, 4)	3702	2397	1071	1028	39	2	1
(5, 5)	3-(-40, 8)	3702	2397	1071	1026	41	1	1
(5, 5)	4-(-45, 6)	3702	2397	1071	1022	44	1	1

E - Equation, V - Total variables, BV - Binary variables, No SP - no specific probing techniques, With SP - with specific probing techniques, T-Wall clock time.

than Scenario C.

The four Scenarios A, B, C, and D have the same original size because the size only depends on the number of grid points postulated and the number of obstacles. Therefore, moving the initial positions or targets further away from the obstacles does not increase the size of the models. However, it is clear that Scenario A requires more computational time than Scenarios B, C, and D. In Table 7, we compare the performance of the MILP solver for Scenarios A and B. Both scenarios lead to similar optimal values of the LP relaxations at the beginning of the

Table 7: Case 2, Scenarios A and B. Performance of the MILP solver, without the specific probing techniques. $\Delta t = 0.2$ s and $N_t = 50$.

Initial position	Target	Gap (%)			T (s)	
		t_f^R	end of root node	t_f (s)	1 st 0-1 solution	Final
1-(-20, 4)	(5, 5)	16.91	NA	27	2	10
2-(-7, 4)	(5, 5)	10.00	NA	19	10	23
3-(-10, 8)	(5, 5)	11.00	NA	21	4	24
4-(-15, 6)	(5, 5)	15.00	70.00	24	0.2	16
(5, 5)	1-(-20, 4)	16.47	25.00	27	1	2
(5, 5)	2-(-7, 4)	8.83	38.10	19	1	2
(5, 5)	3-(-10, 8)	10.54	28.57	21	1	2
(5, 5)	4-(-15, 6)	13.51	27.83	24	1	1

t_f^R - optimal value of the LP relaxation at the root node; NA - Not available, which means no integer solution found at the root node; T-Wall clock time.

root node, but for Scenario B, the MILP solver can find integer solutions at the root node in the four instances and in three instances it can find integer solutions faster than in Scenario A.

Overall, these results show that the size of the models is the same and that the maximum difference between the optimal values of the LP relaxations, t_f^R , of equivalent instances of Scenarios A and B is 1.49 over the optimal value of $t_f = 24$ s for Instance 3, see Table 7. However, the performance of the MILP solver with Scenario A is one order of magnitude slower than with the other three scenarios. These results can be explained by the effect of the trajectory of the AUV after the destination target.

The formulations described in Section 2 do not assume that the vehicle stops at the destination target, they just minimize the time to arrive at the target. Therefore, when the number of time grid points is greater than the travel time, after the AUV arrives at the destination, the models still account for a feasible trajectory, a tail, until the last time grid point postulated. The results show that when the tail includes several time grid points near some obstacles, the MILP solver requires additional computational time, as in Scenario A. This is explained by the fact that this tail considers binary variables, which relate the trajectory with the obstacles, that do not influence the objective function and lead to symmetric solutions. For example, in Scenario A, the trajectory with the initial position 2 reaches the destination in 19 s. Therefore, it has 31 additional time grid points that define a feasible trajectory avoiding the obstacles after the destination target. Moreover, also in Scenario A, the initial positions that are closer to the obstacles require more computational time than the initial positions further left from the

obstacles. For example, the initial position 2 (closer to the obstacles) requires a computational time of 21 s (using the specific probing techniques), whereas, the initial positions 1 and 4 lead to a computational time of 12 s and 10 s, respectively; see Table 5. In Scenario B, the trajectory tail is on the further left of the obstacles and by construction the binary variables of the tail do not lead to symmetric solutions.

In Scenario C, the initial positions and destination target are at the same relative position to the obstacles by comparison with Scenario A, but the number of time grid points after the destination is smaller than in Scenario A.

To further support the claim regarding the influence of the trajectory tail, we built an alternative model where the AUV stops at the destination target, but it still carries the constraints and continuous and binary variables for the remaining time grid points after the arrival time. The immobilization at the destination target is enforced by the new constraints proposed in Appendix C. In this new model, the binary variables associated with the tail only have one optimal value. Table 8 compares the MILP performance of the MILP solver applied to the Scenario A, where the vehicle does not stop at the target and applied to the Scenario A where the vehicle stops at the target. The results show that the LP relaxations at the beginning of the root node are similar, but the formulation where the vehicle stops is more efficient than the original. In the original, the minimum and maximum computational times are 10 s and 24 s, whereas in the new formulation are 4 s and 7 s. These results support the conclusion that in Scenario A the tail has an impact on the performance of the MILP solver.

Table 8: Case 2. Comparison between the Scenario A and Scenario A with a new model where the vehicle stops at the target. Performance of the MILP solver, without the specific probing techniques. $\Delta t = 0.2$ s and $N_t = 50$.

Scenario	Initial position	Target	t_f^R	t_f (s)	T (s)
A	1-(-20, 4)	(5, 5)	16.91	27	10
	2-(-7, 4)	(5, 5)	10.00	19	23
	3-(-10, 8)	(5, 5)	11.00	21	24
	4-(-15, 6)	(5, 5)	15.00	24	16
A - with new formulation where the vehicle stops at the target See Appendix C	1-(-20, 4)	(5, 5)	15.00	28	5
	2-(-7, 4)	(5, 5)	9.00	19	5
	3-(-10, 8)	(5, 5)	11.00	21	7
	4-(-15, 6)	(5, 5)	14.00	25	4

t_f^R - optimal value of the LP relaxation at the root node; T-Wall clock time.

4.3. Case 3

In this case, we consider one scenario with four instances: each instance has a different target, but all instances have the same initial position. The optimal trajectories for each instance are presented in Figure 8, where we can observe that the model can lead the AUV to navigate around the seven obstacles to arrive to the target.

Table 9 presents the size of the models and computational results for the four instances. These results show that the MILP solver requires less than 2 seconds to solve each of the four instances. In Table 10, we show that the optimal value of the LP relaxations at the beginning

Figure 8: Optimal paths (top) and velocity and acceleration profiles of a single vehicle for Case 3 with 7 obstacles. The obstacles are represented by $\blacksquare$, and the starting point by $\circ$. The noncontinuous lines are the safety margins. Different paths correspond to the same starting point, all with different destinations.

Table 9: Case 3. Size and results of the MILP. $\Delta t = 0.2$ s and $N_t = 50$.

Initial position	Destinations	E#	V#	BV#	t_f (s)	T (s)
(-4, 0)	1-(6, 3)	4212	1326	1479	7.4	1
(-4, 0)	2-(6, 4)	4212	1326	1479	7.6	1
(-4, 0)	3-(6, -2)	4212	1326	1479	7.2	1
(-4, 0)	4-(6, -4.5)	4212	1326	1479	8	2

E - Equation, V - Total variables, BV - Binary variables, T-Wall clock time.

Table 10: Case 3. Performance of the MILP solver. $\Delta t = 0.2$ s and $N_t = 50$.

Initial position	Target	Gap (%)			T (s)	
		t_f^R	end of root node	t_f (s)	1 st 0-1 solution	Final
(-4, 0)	1-(6, 3)	6.98	5.26	7.4	0.5	1
(-4, 0)	2-(6, 4)	7.15	2.56	7.6	1	1
(-4, 0)	3-(6, -2)	6.84	0.00	7.2	1	1
(-4, 0)	4-(6, -4.5)	7.26	22.38	8	1	2

t_f^R - optimal value of the LP relaxation at the root node; T-Wall clock time.

of the root node have a maximum relative gap of 9.25%. In the third instance, the MILP solver finds the optimal solution at the root node, given the tightness of the LP relaxation; see Table 10.

The MILP solver requires less time to solve these instances than the instances from Case 2, Scenario A, although the model from Case 3 is larger than the model from Case 2; see Tables 5 and 9. This difference in the performance of the solver is normal, given that the relative performance of a given MILP solver does not depend exclusively on the size of the problems,

but rather on the combination of several factors: a) overall size of the problem, which has an impact on the time spent on each node of the branch & cut algorithms to solve the LP problem; b) tightness of the LP relaxation; c) number of binary variables; d) existence of symmetric solutions; and e) localization of the integer solutions in the branch & cut tree. Note that the performance of the MILP solver for a given problem depends on the features of the solver, namely the pre-processing strategies, the sophistication of the simplex algorithm, the cuts implemented, the heuristics used, the branching options, and the parallelization strategies implemented (Wolsey, 1998; Lima, 2010; Lima and Grossmann, 2011).

4.4. Case 4

Scenario 1. The optimization runs performed to demonstrate the proposed solution approach are summarized in Table 11. Without current, the time-optimal paths generated by the MILP model with and without waypoints are plotted in Fig. 9. These paths are used to initialize the MINLP models. The optimal trajectories corresponding to the best solution in Table 11 are presented in Fig. 10 and 11, along with the vehicle's velocity and acceleration profiles.

Table 11: Case 4, Scenario 1. Size of the models and results of the proposed methodology. $\Delta t = 1$ s and $N_t = 70$. **Bold** - the best solution.

Model	E#	V#	BV#	Optimization runs in parallel									
				No WP		WP1 (60, 32.75)		WP2 (60, 38.5)		WP3 (60, 44.25)		WP4 (60, 5)	
				t_f	T	t_f	T	t_f	T	t_f	T	t_f	T
MILP	4739	1847	994	60	2	61	1	62	1	66	1	70	1
MINLP _S	2487	926	994	60	14	62	31	63	18	66	20	70	22
MINLP _C	1916	666	898	60	31	62	63	63	58	65	37	64	79
MINLP _{C-1}	1916	666	898	55	127	50	100	48	102	46	42	51	84
MINLP _{C-2}	1916	666	898	43	154	46	113	37	66	43	85	48	187
MINLP _D	2484	1234	898	**	**	46	3652	37	3351	**	**	48	3562

WP (x, y) - waypoint with coordinates (x, y) , E - Equation, V - Total Variables, BV - Binary Variables, T - Wall clock time (s). The unit for t_f is seconds. MINLP_{C-1} and MINLP_{C-2} are the MINLP_C model run in parallel. ** - solver aborted without returning a solution.

As shown in Table 11, not all of the columns can return a solution of the detailed MINLP model with current. Specifically, in some cases (initializations) the solver terminates without returning a solution, which highlights the importance of the diversification. The MILP and MINLP_S model have more binary variables than the MINLP_C and MINLP_D model. The MILP model is always faster than the MINLP model. For example, with WP1 the MILP model takes 1 second while the MINLP_S model takes 31 seconds to return a solution. Other MINLP models with more details require more computational time to complete the refinement step. Starting from the solution of the MINLP_S model, for the present case the MINLP_C model always returns a solution, and the constraint in (54) forces the solver to seek improved solutions in parallel. As the problems are solved in parallel, the strategy is efficient; it is also effective in improving the solution, particularly with WP2 where the travel time is reduced

from $t_f = 63$ s to $t_f = 37$ s. The best solution from MINLP models with current is used to initialize the detailed models, MINLP_C and MINLP_D. The travel times given by the MINLP_C and MINLP_D model are the same: 46 s with WP1, 37 s with WP2, and 48 s with WP4. The corresponding paths in currents are also very close, see Fig. 10, but the path obtained with MINLP_D is smoother than that obtained with MINLP_C. Thus, the MINLP_D model does not yield a significantly “better” solution than the MINLP_C model but requires a computational time that is approximately 51 times larger.

It is also interesting to note that the optimal travel times obtained with the MINLP models with current are smaller than those obtained with the MILP without current. This result indicates that the time optimal trajectories are determined in such a way that the vehicle effectively takes advantage of favorable conditions. Also note that for the optimal trajectory, the vehicle’s acceleration profile has the smallest energy, see Fig. 11.

Figure 9: Case 4, Scenario 1. Optimal paths with and without waypoints obtained with the MILP model. The waypoints are represented by $\diamond$, the destination by $\square$, and the noncontinuous lines are the safety margins. $\Delta t = 1$ s and $N_t = 70$.

Scenario 2. The optimization runs performed to demonstrate the proposed solution approach are summarized in Table 11. Figure 12 illustrates time-optimal trajectories generated by the MILP model with and without waypoints. The optimal trajectories corresponding to the best solutions in Table 12 are plotted in Fig. 13.

In Table 12, the solver returns solutions only when initialized without waypoints and with WP2. The algorithm terminates without returning a solution with WP1, WP3, WP4, and WP5. Thus, even starting from a good initialization, namely the solution of the MINLP model

Figure 10: Case 4, Scenario 1. Optimal paths obtained with each model of the proposed solution approach with the waypoint WP2. The correspondence between the lines and models is the following + - - MILP, o - - MINLP_S, yellow, purple, and green lines- MINLP_C, • - - MINLP_D.

Figure 11: Case 4, Scenario 1. Profiles of the velocities and accelerations of the vehicle obtained with each model of the proposed solution approach with the waypoint WP2.

without current, the MINLP model may fail to find a solution when the current is accounted for. This outcome is in contrast with the previous case (Table 11), where the MINLP with current always returned a solution. This difference in the performance can be explained by noting that in adverse conditions, the initial conditions determined assuming no current may become infeasible when the current is accounted for.

Table 12: Case 4, Scenario 2. Size of the models and results of the proposed methodology. $\Delta t = 1$ and $N_t = 100$. **Bold** - the best solution.

Model	E#	V#	BV#	Optimization runs in parallel									
				No WP		WP1 (60, 32.75)		WP2 (60, 38.5)		WP3 (60, 44.25)		WP4 (60, 5)	
				t_f	T	t_f	T	t_f	T	t_f	T	t_f	T
MILP	6749	2627	1414	35	2	38	5	42	6	48	6	42	7
MINLP _S	3537	1316	1414	35	18	38	23	42	24	48	60	42	32
MINLP _C	2726	925	1299	51	218	**	**	42	1911	**	**	**	**
MINLP _C	2726	925	1299	**	**	**	**	39	1279	**	**	**	**
MINLP _D	3534	1733	1299	51	4434	**	**	39	5997	**	**	**	**

WP (x, y) - waypoint with coordinates (x, y) , E - Equation, V - Total variables, BV - Binary variables, T - Wall clock time (s). The unit for t_f is seconds. MINLP_{C-1} and MINLP_{C-2} are the MINLP_C model run in parallel. ** - solver aborted without returning a solution.

Figure 13 shows that the trajectory leading to the smallest travel time, $t_f = 39$ s, comes close to the surface of the obstacle, namely to minimize the impact of the adverse current. The paths generated by the MINLP_C and MINLP_D models are again very close to each other, but the computational time required by MINLP_D is approximately 4.7 times larger than that of MINLP_D, see Table 12.

Figure 12: Case 4, Scenario 2. Optimal paths with and without waypoints obtained with the MILP model. The waypoints are represented by $\diamond$, the destination by $\square$, and the noncontinuous lines are the safety margins. $\Delta t = 1$ s and $N_t = 70$.

Figure 13: Case 4, Scenario 2. Optimal paths obtained with each model of the proposed solution approach with the waypoint WP2. The correspondence between the lines and models is the following + - MILP, o - MINLP_S, yellow and green lines- MINLP_C, ●- MINLP_D.

5. Conclusions

In this paper, we have explored the application of MINLP models for AUV trajectory planning in the presence of currents and obstacles. Due to the nature of the currents studied, the resulting MINLP models are non-convex, which poses difficulties in solver initialization and convergence to local solutions. To tackle these hurdles, we developed an approach involving a sequential solution of problems of increasing complexity, based on diversification strategy for initialization and progressive refinement of individual solutions. In particular, the diversification strategy relies on the utilization of waypoints to explore different regions of space.

The performance of the solution approach is optimized by performing several tasks in parallel, namely the parallel solution of multiple independent MILP and MINLP problems. The computational experiments show that: 1) in general, the MILP solver solves to optimality the MILP problems in less than 10 s, several cases are solved in approximately 1 s; 2) for the cases where the number of grid points is significantly larger than the minimum travel time and the target is surrounded by obstacles, the performance of the MILP solver decreases by comparison with the performance for a similar model and same obstacles but with the target apart from the obstacles; 3) the MILP models are much faster than the MINLP models, suggesting the utilization of MILP models for initializing MINLP models; 4) the solution approach can initialize the detailed MINLP model, and the waypoint approach contributes to finding improved

solutions; 4) the detailed MINLP model considering the current acceleration requires longer computational times comparing with the MINLP model without current acceleration; 5) the trajectory obtained with the detailed MINLP model is smoother than the one obtained with the MINLP model without current; 6) the optimal trajectories obtained effectively take advantage of favorable currents and mitigate the impact of adverse currents.

Future work will involve the extension of the proposed models to address transient flows, uncertainty in the flow fields, and long-term missions. Considering uncertainty in the flows, the models will be extended to two-stage stochastic programming MINLP models with recourse actions to account for the adaptability of the vehicle to uncertain conditions. We will also investigate the characterization of the uncertain flows through ensembles obtained from ocean general circulation models.

Appendix A. Derivation of the objective function

In this Appendix, we show the derivation of the objective function in (23) for a uniform time discretization, and for a special case where the time steps are considered variables. First, using a uniform time discretization with equal known time steps Δt_j , $\forall j \in T$, the disjunction in (17) is transformed into the following equations using a convex-hull reformulation (Balas, 1985):

$$t_f = \sum_{j \in J} t_f^j, \quad (\text{A.1})$$

$$t_f^j \geq \left(\sum_{i=0}^j \Delta t_i \right) b_{j+1}, \quad \forall j \in J, \quad (\text{A.2})$$

$$t_f^j \leq \left(\sum_{i=0}^j \Delta t_i \right) b_{j+1}, \quad \forall j \in J, \quad (\text{A.3})$$

$$\sum_{j \in J} b_j = 1. \quad (\text{A.4})$$

Assuming a uniform time grid, we have

$$t_f = \sum_{j \in J} t_f^j = \sum_{j \in J} \left(\sum_{i=0}^j \Delta t_i \right) b_{j+1} = \sum_{j \in J} [(j+1)\Delta t b_{j+1}]. \quad (\text{A.5})$$

If the time steps are variable in a non-uniform time discretization, then the disjunction (17) can be reformulated using a Big-M reformulation, according to

$$t_f \geq \sum_{i=0}^j \Delta t_i - M(1 - b_{j+1}), \quad \forall j \in J, \quad (\text{A.6})$$

$$t_f \leq \sum_{i=0}^j \Delta t_i + M(1 - b_{j+1}), \quad \forall j \in J, \quad (\text{A.7})$$

$$\sum_{i \in I} b_i = 1. \quad (\text{A.8})$$

Appendix B. Derivation of the constraints for obstacle avoidance

This appendix provides details about the derivation of the constraints for obstacle avoidance, and then offers an alternative formulation.

The disjunctions in (29) and (30) are reformulated using the Big-M reformulation, leading to

$$x_i \leq \underline{x}_o + M_{2,1}(1 - z_{o,i}^1), \quad \forall o \in O, \forall i \in I, \quad (\text{B.1})$$

$$-x_i \leq -\bar{x}_o + M_{2,2}(1 - z_{o,i}^3), \quad \forall o \in O, \forall i \in I, \quad (\text{B.2})$$

$$y_i \leq \underline{y}_o + M_{2,3}(1 - z_{o,i}^4), \quad \forall o \in O, \forall i \in I, \quad (\text{B.3})$$

$$-y_i \leq -\bar{y}_o + M_{2,4}(1 - z_{o,i}^6), \quad \forall o \in O, \forall i \in I, \quad (\text{B.4})$$

$$z_{o,i}^1 + z_{o,i}^2 + z_{o,i}^3 = 1, \quad \forall o \in O, \forall i \in I, \quad (\text{B.5})$$

$$z_{o,i}^4 + z_{o,i}^5 + z_{o,i}^6 = 1, \quad \forall o \in O, \forall i \in I. \quad (\text{B.6})$$

The logical propositions in (32) and (33) are transformed to:

$$\neg Z_{o,i}^2 \vee Z_{o,i}^4 \vee Z_{o,i}^6, \quad \forall o \in O, \forall i \in I, \quad (\text{B.7})$$

$$\neg Z_{o,i}^5 \vee Z_{o,i}^1 \vee Z_{o,i}^3, \quad \forall o \in O, \forall i \in I, \quad (\text{B.8})$$

and are converted to

$$z_{o,i}^4 + z_{o,i}^6 \geq z_{o,i}^2, \quad \forall o \in O, \forall i \in I, \quad (\text{B.9})$$

$$z_{o,i}^1 + z_{o,i}^3 \geq z_{o,i}^5, \quad \forall o \in O, \forall i \in I, \quad (\text{B.10})$$

which, by using the equations (B.5) and (B.6), are converted into a single inequality (38). In fact, this derivation shows that an equivalent disjunctive formulation is defined as

$$\left[\begin{array}{c} Z_{o,i}^1 \\ x_i \leq \underline{x}_o \end{array} \right] \vee \left[\begin{array}{c} Z_{o,i}^3 \\ x_i \geq \bar{x}_o \end{array} \right] \vee \left[\begin{array}{c} Z_{o,i}^4 \\ y_i \leq \underline{y}_o \end{array} \right] \vee \left[\begin{array}{c} Z_{o,i}^6 \\ y_i \geq \bar{y}_o \end{array} \right], \quad \forall o \in O, \forall i \in I. \quad (\text{B.11})$$

Alternatively, the logic propositions (25) to (28) can be represented with the following form of linear disjunctions

$$\left[\begin{array}{c} \neg Z_{o,i}^1 \\ x_i \leq \underline{x}_o \end{array} \right] \vee \left[\begin{array}{c} Z_{o,i}^1 \wedge Z_{o,i}^2 \\ \underline{x}_o \leq x_i \leq \bar{x}_o \end{array} \right] \vee \left[\begin{array}{c} \neg Z_{o,i}^2 \\ x_i \geq \bar{y}_o \end{array} \right], \quad \forall o \in O, \forall i \in I, \quad (\text{B.12})$$

$$\left[\begin{array}{c} \neg Z_{o,i}^3 \\ y_i \leq \underline{y}_o \end{array} \right] \vee \left[\begin{array}{c} Z_{o,i}^3 \wedge Z_{o,i}^4 \\ \underline{y}_o \leq y_i \leq \bar{y}_o \end{array} \right] \vee \left[\begin{array}{c} \neg Z_{o,i}^4 \\ y_i \geq \bar{y}_o \end{array} \right], \quad \forall o \in O, \forall i \in I, \quad (\text{B.13})$$

$$Z_{o,i}^1, Z_{o,i}^2, Z_{o,i}^3, Z_{o,i}^4 \in \{\text{True}, \text{False}\}, \quad \forall o \in O, \forall i \in I,$$

$$Z_{o,i}^1 \wedge Z_{o,i}^2 \Rightarrow \neg Z_{o,i}^3 \vee \neg Z_{o,i}^4, \quad \forall o \in O, \forall i \in I, \quad (\text{B.14})$$

$$Z_{o,i}^3 \wedge Z_{o,i}^4 \Rightarrow \neg Z_{o,i}^1 \vee \neg Z_{o,i}^2, \quad \forall o \in O, \forall i \in I, \quad (\text{B.15})$$

where $Z_{o,i}^1, \dots, Z_{o,i}^4$ are Boolean variables. Using a Big-M reformulation, the disjunctions in (B.12) and (B.13) can be converted to

$$\begin{aligned} x_i &\leq \bar{x}_o + M_{o,i}^{2,1} (1 - (1 - z_{o,i}^1)), \quad \forall o \in O, \forall i \in I, \\ -x_i &\leq -\bar{x}_o + M_{o,i}^{2,2} (1 - (1 - z_{o,i}^2)), \quad \forall o \in O, \forall i \in I, \\ y_i &\leq \underline{y}_o + M_{o,i}^{2,3} (1 - (1 - z_{o,i}^3)), \quad \forall o \in O, \forall i \in I, \\ -y_i &\leq -\underline{y}_o + M_{o,i}^{2,4} (1 - (1 - z_{o,i}^4)), \quad \forall o \in O, \forall i \in I, \end{aligned} \quad (\text{B.16})$$

where $M_{o,i}^{2,1} = \max\{x_i - \bar{x}_o : x_{\min} \leq x_i \leq x_{\max}\}$, $M_{o,i}^{2,2} = \max\{-x_i + \bar{x}_o : x_{\min} \leq x_i \leq x_{\max}\}$, $M_{o,i}^{2,3} = \max\{y_i - \underline{y}_o : x_{\min} \leq x_i \leq x_{\max}\}$, and $M_{o,i}^{2,4} = \max\{-y_i + \underline{y}_o : x_{\min} \leq x_i \leq x_{\max}\}$.

The propositions in (B.14) (B.15) can be transformed to:

$$\begin{aligned} \neg (Z_{o,i}^1 \wedge Z_{o,i}^2) \vee \neg Z_{o,i}^3 \vee \neg Z_{o,i}^4, \quad \forall o \in O, \forall i \in I, \\ \neg (Z_{o,i}^3 \wedge Z_{o,i}^4) \vee \neg Z_{o,i}^1 \vee \neg Z_{o,i}^2, \quad \forall o \in O, \forall i \in I. \end{aligned} \quad (\text{B.17})$$

Applying De Morgan's laws to (B.17), the two propositions become

$$\neg Z_{o,i}^1 \vee \neg Z_{o,i}^2 \vee \neg Z_{o,i}^3 \vee \neg Z_{o,i}^4, \quad \forall o \in O, \forall i \in I, \quad (\text{B.18})$$

which can be represented as

$$z_{o,i}^1 + z_{o,i}^2 + z_{o,i}^3 + z_{o,i}^4 \leq 3, \quad \forall o \in O, \forall i \in I. \quad (\text{B.19})$$

Appendix C. Derivation of the constraints to stop the vehicle at the target

The constraints that model the arrival of the vehicle to the target, defined in Section 2.1, do not enforce the vehicle to stop at the destination target. Therefore, the formulations derived focus on the minimization of the time to arrive to the target, but the trajectory after the target is reached does not influence the objective function.

In this section, we propose a new formulation for models without current that forces the immobilization of the vehicle at the target. We define a new Boolean variable, C_i , which is True for all time grid points after the vehicle has arrived at the target:

$$B_i \Rightarrow C_{ii}, \quad \forall ii \geq i \in I, \quad (\text{C.1})$$

$$C_i \in \{\text{True}, \text{False}\}, \quad \forall i \in I, \quad (\text{C.2})$$

whereas, C_i is False for all the time grid points before the vehicle has arrived at the target:

$$B_i \Rightarrow \neg C_{ii}, \quad \forall ii < i \in I. \quad (\text{C.3})$$

Using the definition of the Boolean variable C_i , we define the disjunction

$$\left[\begin{array}{c} C_i \\ \cos(\theta_k)u_i + \sin(\theta_k)w_i \leq 0, \forall k \in K \\ \cos(\alpha_k)\dot{u}_i + \sin(\alpha_k)\dot{w}_i \leq 0, \forall k \in K \end{array} \right] \vee \left[\begin{array}{c} \neg C_i \\ \cos(\theta_k)u_i + \sin(\theta_k)w_i \leq \bar{v}, \forall k \in K \\ \cos(\alpha_k)\dot{u}_i + \sin(\alpha_k)\dot{w}_i \leq \bar{a}, \forall k \in K \end{array} \right], \forall i \in I, \quad (C.4)$$

which enforces $\|\mathbf{v}_i\| \leq 0, \forall i \in I$ and $\|\mathbf{a}_i\| \leq 0, \forall i \in I$ after the vehicle has arrived at the target.

The expressions in (C.1), (C.3), and (C.4) can be reformulated as

$$c_{ii} \geq b_i, \quad \forall i \geq i \in I, \quad (C.5)$$

$$b_i + c_{ii} \leq 1, \quad \forall i < i \in I, \quad (C.6)$$

$$\cos(\theta_k)u_i + \sin(\theta_k)w_i \leq \bar{v}(1 - c_i), \quad \forall k \in K, \forall i \in I, \quad (C.7)$$

$$\cos(\alpha_k)\dot{u}_i + \sin(\alpha_k)\dot{w}_i \leq \bar{a}(1 - c_i), \quad \forall k \in K, \forall i \in I, \quad (C.8)$$

where $c_i, c_{ii} \in \{0, 1\}, \forall i \in I$. The variable c_i is equal to 1 for all time grid points after the vehicle has arrived at the target and 0 otherwise.

Note that the constraints derived in this section do not force the immobilization of the vehicle in the presence of current.

Acknowledgment

Research reported in this publication was supported by research funding from King Abdullah University of Science and Technology (KAUST).

Nomenclature

Indices, sets

i, I indices for the time grid points and time steps

j, J indices for the time steps

k, K indices for the discretization of the angles of the velocity and acceleration projected vectors

o, O obstacles

Parameters

$\bar{a}$ upper bound on the acceleration of the vehicle relative to the current

Δt constant time step, $\Delta t_0 = \dots = \Delta t_{N_t-1} = \Delta t$

Δt_i time step

M_i big-M constant with values defined as indicated in the text

$M_{o,i}^{l,i}$ big-M constant with values defined as indicated in the text

n integer number

N_t number of time grid points in the time discretization

p pressure parameter used in the simulation to generate the velocity field

S safety margin added to the coordinates of the obstacles

t_0	initial time time grid point
t_i	time grid point
$\bar{v}$	upper bound on the velocity of the vehicle relative to the current
$\mathbf{x}_0$	initial position, (x, y)
$\mathbf{x}_f$	final position, (x, y)
$x_{\max}$	maximum x coordinate
$x_{\min}$	minimum x coordinate
$x_{\max}^F$	maximum x coordinate of the destination target
$x_{\min}^F$	minimum x coordinate of the destination target
$\bar{x}_o$	maximum x coordinate of the obstacle o
$\underline{x}_o$	minimum x coordinate of the obstacle o
$x_{\max}^w$	maximum x coordinate of the waypoint target
$x_{\min}^w$	minimum x coordinate of the waypoint target
$y_{\max}$	maximum y coordinate
$y_{\min}$	minimum y coordinate
$y_{\max}^F$	maximum y coordinate of the destination target
$y_{\min}^F$	minimum y coordinate of the destination target
$\bar{y}_o$	maximum y coordinate of the obstacle o
$\underline{y}_o$	minimum y coordinate of the obstacle o
$y_{\max}^w$	maximum y coordinate of the waypoint target
$y_{\min}^w$	minimum y coordinate of the waypoint target
α_k	discrete angles of the acceleration projected vectors
ν	kinematic viscosity parameter used in the simulation to generate the velocity field
θ_k	discrete angles of the velocity projected vectors
Continuous variables	
$\mathbf{a}(t)$	acceleration of the vehicle relative to the current, $(\dot{u}, \dot{w})$
$\mathbf{a}_i$	acceleration of the vehicle relative to the current, $(\dot{u}_i, \dot{w}_i)$
t_f	time to travel from the initial position $\mathbf{x}_0$ to the destination $\mathbf{x}_f$
t_{wp}	time to travel from the initial position to a waypoint
$\mathbf{v}(t)$	velocity of the vehicle relative to the current, (u, w)
$\mathbf{v}_i$	velocity of the vehicle relative to the current in period i , (u_i, w_i)
$\mathbf{x}(t)$	position of the vehicle in the two-dimension Euclidean space,
$\mathbf{f}(\mathbf{x}(t))$	velocity of the current at $\mathbf{x}(t)$
$\mathbf{x}_i$	position of the vehicle, (x_i, y_i)
x_i	x coordinate of the vehicle
y_i	y coordinate of the vehicle
Binary variables	
b_i	equal to 1 if the vehicle position is within the destination target area, and 0 otherwise
c_i	equal to 1 for all time grid points after the vehicle has arrived to the destination target, and 0 otherwise
s_i	equal to 1 if the vehicle position is within the waypoint target area, and 0 otherwise

$z_{o,i}^m$	equal to 1 if the vehicle position respects the constraints of the m term of the disjunction relative to obstacles avoidance
Boolean variables	
B_i	True if the vehicle position is within the destination target defined by $x_{\min}^F, x_{\max}^F, y_{\min}^F, y_{\max}^F$
C_i	True for all time grid points after the vehicle has arrived to the destination target
$Z_{o,i}^m$	True if the vehicle position respects the m term of the disjunction relative to obstacles avoidance.

References

References

- Abichandani, P., Ford, G., Benson, H. Y., Kam, M., May 2012. Mathematical programming for multi-vehicle motion planning problems. In: 2012 IEEE International Conference on Robotics and Automation. pp. 3315–3322.
- Abichandani, P., Torabi, S., Basu, S., Benson, H., 2015. Mixed Integer Nonlinear Programming Framework for Fixed Path Coordination of Multiple Underwater Vehicles Under Acoustic Communication Constraints. *IEEE Journal of Oceanic Engineering* 40 (4), 864–873.
- Afonso, R. J., Galvão, R. K., Kienitz, K. H., 2016. Reduction in the number of binary variables for inter-sample avoidance in trajectory optimizers using mixed-integer linear programming. *International Journal of Robust and Nonlinear Control* 26, 36623669.
- Alvarez, A., Caiti, A., Onken, R., 2004. Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *IEEE Journal of Oceanic Engineering* 29 (2), 418–429.
- Androulakis, I., Maranas, C., Floudas, C., 1995. α BB: A global optimization method for general constrained nonconvex problems. *Journal of Global Optimization* 7 (4), 337–363.
- Balas, E., 1985. Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *SIAM Journal on Algebraic and Discrete Methods* 6 (3), 466–486.
- Baumgartner, K. A. C., Ferrari, S., Rao, A. V., 2009. Optimal control of an underwater sensor network for cooperative target tracking. *IEEE Journal of Oceanic Engineering* 34 (4), 678–697.
- Belotti, P., Lee, J., Liberti, L., Margot, F., Waechter, A., 2009. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods & Software* 24 (4-5), 597–634.
- Bonami, P., Biegler, L. T., Conna, A. R., Cornuejols, G., Grossmann, I. E., Laird, C. D., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wachter, A., 2008. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization* 5 (2), 186–204.
- Borrelli, F., Subramanian, D., Raghunathan, A. U., Biegler, L. T., 2006. MILP and NLP techniques for centralized trajectory planning of multiple unmanned air vehicles. In: American Control Conference, 2006. IEEE, pp. 5763–5768.
- Bruce A. McCarl et. al., 2013. McCarl Expanded GAMS User Guide, GAMS Release 24.2.1. GAMS Development Corporation, Washington, DC, USA.
- Carbone, G., Gomez-Bravo, F., 2015. Motion and operation planning of robotic systems. Springer.
- Cardoso, M. F., Salcedo, R. L., Feyo de Azevedo, S. a., Barbosa, D., 1997. A simulated annealing approach to the solution of MINLP problems. *Computers & Chemical Engineering* 21 (12), 1349–1364.
- Dijkstra, E. W., 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* 1 (1), 269–271.
- Drud, A. S., 1994. CONOPT - a large scale GRG code. *ORSA Journal on Computing* 6, 207–216.
- Duran, M., Grossmann, I. E., 1986. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming* 36 (3), 307–339.

- Earl, M. G., D'Andrea, R., 2005. Iterative MILP methods for vehicle-control problems. *IEEE Transactions on Robotics* 21 (6), 1158–1167.
- Edison, E., Shima, T., 2011. Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms. *Computers & Operations Research* 38 (1, SI), 340–356.
- Eichhorn, M., 2015. Optimal routing strategies for autonomous underwater vehicles in time-varying environment. *Robotics and Autonomous Systems* 67 (SI), 33–43.
- Fukushima, H., Kon, K., Matsuno, F., 2013. Model Predictive Formation Control Using Branch-and-Bound Compatible With Collision Avoidance Problems. *IEEE Transactions on Robotics* 29 (5), 1308–1317.
- GAMS Development Corporation, 2017. SBB. Available at <https://www.gams.com/latest/docs/solvers/sbb/index.html>.
- Goerzen, C., Kong, Z., Mettler, B., 2010. A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance. *Journal of Intelligent & Robotic Systems* 57 (1-4, SI), 65–100.
- Grøtli, E. I., Johansen, T. A., 2016. Motion-and communication-planning of unmanned aerial vehicles in delay tolerant network using mixed-integer linear programming. *Modeling, Identification and Control* 37 (2), 77–97.
- Hart, P. E., Nilsson, N. J., Raphael, B., 1968. A formal basis for heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* SSC4 (2), 100–107.
- Kuwata, Y., How, J. P., 2011. Cooperative Distributed Robust Trajectory Optimization Using Receding Horizon MILP. *IEEE Transactions On Control Systems Technology* 19 (2), 423–431.
- Latombe, J. C., 1991. Robot motion planning. Springer Science, New York.
- LaValle, S. M., 1998. Rapidly-exploring Random Trees: A new tool for path planning. Tech. rep., Iowa State University.
- LaValle, S. M., 2006. Planning Algorithms. Cambridge University Press, Cambridge, U.K., also available at <http://planning.cs.uiuc.edu/>.
- Leyffer, S., 2001. Integrating SQP and branch-and-bound for mixed integer nonlinear programming. *Computational Optimization and Applications* 18, 295–309.
- Lima, R., Salcedo, R., Barbosa, D., 2006. SIMOP efficient reactive distillation optimization using stochastic optimizers. *Chemical Engineering Science* 61 (5), 1718–1739.
- Lima, R. M., 2010. IBM ILOG CPLEX What is inside of the box? EWO Seminar Slides. URL http://egon.cheme.cmu.edu/ewo/docs/rlima_cplex_ewo_dec2010.pdf
- Lima, R. M., Grossmann, I. E., 2011. Chemical Engineering Greetings to prof. Sauro Pierucci. AIDIC, Ch. Computational advances in solving mixed integer linear programming problems, pp. 151–160.
- Lindo Systems, I., 2017. LINDO and LINDOGlobal. Lindo Systems, Inc., Available at <https://www.gams.com/latest/docs/solvers/lindo/index.html>.
- Lolla, T., 2012. Path planning in time dependent flows using level set methods. Master's thesis, Massachusetts Institute of Technology. Dept. of Mechanical Engineering.
- Lolla, T., Haley, Jr., P. J., Lermusiaux, P. F. J., 2014a. Time-optimal path planning in dynamic flows using level set equations: realistic applications. *Ocean Dynamics* 64 (10), 1399–1417.
- Lolla, T., Lermusiaux, P. F. J., Ueckermann, M. P., Haley, Jr., P. J., 2014b. Time-optimal path planning in dynamic flows using level set equations: theory and schemes. *Ocean Dynamics* 64 (10), 1373–1397.
- Luo, J., Hauser, K., 2014. An Empirical Study of Optimal Motion Planning. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and systems (IROS 2014). pp. 1761–1768.
- Maia, M. H., Galvão, R. K. H., 2009. On the use of mixed-integer linear programming for predictive control with avoidance constraints. *International Journal of Robust and Nonlinear Control* 19, 822–828.

- Misener, R., Floudas, C. A., 2014. ANTIGONE: Algorithms for coNTinuous/Integer Global Optimization of Nonlinear Equations. *Journal of Global Optimization* 59 (2-3), 503–526.
- Mitchell, M., 1996. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA.
- Morgan, D., Chung, S.-J., Hadaegh, F. Y., 2014. Model Predictive Control of Swarms of Spacecraft Using Sequential Convex Programming. *Journal of Guidance, Control, and Dynamics* 37 (6), 1725–1740, AIAA/AAS Astrodynamics Specialist Conference, Minneapolis, MN, AUG 13-16, 2012.
- Mufalli, F., Batta, R., Nagi, R., 2012. Simultaneous sensor selection and routing of unmanned aerial vehicles for complex mission plans. *Computers & Operations Research* 39 (11), 2787–2799.
- Osher, S., Sethian, J. A., 1988. Fronts Propagating with Curvature-Dependent Speed - Algorithms based on Hamilton-Jacobi Formulations. *Journal of Computational Physics* 79 (1), 12–49.
- Park, J., Karumanchi, S., Iagnemma, K., 2015. Homotopy-based divide-and-conquer strategy for optimal trajectory planning via mixed-integer programming. *IEEE Transactions on Robotics* 31 (5), 1101–1115.
- Pereira, A. A., Binney, J., Hollinger, G. A., Sukhatme, G. S., 2013. Risk-aware path planning for autonomous underwater vehicles using predictive ocean models. *Journal of Field Robotics* 30 (5), 741–762.
- Pêtrès, C., Pailhas, Y., Patrón, P., Petillot, Y., Evans, J., Lane, D., 2007. Path planning for autonomous underwater vehicles. *IEEE Transactions on Robotics* 23 (2), 331–341.
- Radmanesh, M., Kumar, M., 2016. Flight formation of uavs in presence of moving obstacles using fast-dynamic mixed integer linear programming. *Aerospace Science and Technology* 50, 149–160.
- Rao, D., Williams, S. B., 2009. Large-scale path planning for Underwater Gliders in ocean currents. In: *Australasian Conference on Robotics and Automation (ACRA)*.
- Richards, A., How, J. P., 2002. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In: *Proceedings of the 2002 American Control Conference*. Vol. 3. IEEE, pp. 1936–1941.
- Richards, A., How, J. P., 2006. Robust variable horizon model predictive control for vehicle maneuvering. *International Journal of Robust and Nonlinear Control* 16 (7), 333–351.
- Richards, A., Schouwenaars, T., How, J. P., Feron, E., 2002. Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming. *Journal of Guidance, Control, and Dynamics* 25 (4), 755–764.
- Richards, A., Turnbull, O., 2015. Inter-sample avoidance in trajectory optimizers using mixed-integer linear programming. *International Journal of Robust and Nonlinear Control* 25, 521–526.
- Sahinidis, N., 2015. BARON. Available at <https://www.gams.com/latest/docs/solvers/baron/index.html>.
- Schouwenaars, T., De Moor, B., Feron, E., How, J., 2001. Mixed integer programming for multi-vehicle path planning. In: *Control Conference (ECC), 2001 European*. IEEE, pp. 2603–2608.
- Schouwenaars, T., Feron, E., How, J., Roche, E., Valenti, M., 2006. Linear programming and language processing for human-unmanned aerial-vehicle team missions. *Journal of Guidance, Control, and Dynamics* 29 (2), 303–313.
- Schouwenaars, T., How, J., Feron, E., 2004a. Receding horizon path planning with implicit safety guarantees. In: *American Control Conference, 2004. Proceedings of the 2004*. Vol. 6. IEEE, pp. 5576–5581.
- Schouwenaars, T., Mettler, B., Feron, E., How, J., 2004b. Hybrid model for trajectory planning of agile autonomous vehicles. *Journal of Aerospace Computing, Information, and Communication* 1 (12), 629–651.
- Schweiger, C. A., Floudas, C. A., 1998. Process synthesis, design, and control: A mixed-integer optimal control framework. In: *Proceedings of DYCOPS-5*. pp. 191–196.

- Sethian, J. A., 1999. Level set methods and fast marching methods. Cambridge University Press, Cambridge, Massachusetts.
- Shetty, V. K., Sudit, M., Nagi, R., 2008. Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles. *Computers & Operations Research* 35 (6), 1813–1828.
- Tawarmalani, M., Sahinidis, N., 2005. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming* 103, 225–249.
- Turnbull, O., Lawry, J., Lowenberg, M., Richards, A., 2016. A cloned linguistic decision tree controller for real-time path planning in hostile environments. *Fuzzy Sets and Systems* 293, 1–29.
- Vecchietti, A., Lee, S., Grossmann, I. E., 2003. Modeling of discrete/continuous optimization problems: characterization and formulation of disjunctions and their relaxations. *Computers & Chemical Engineering* 27 (3), 433 – 448.
- Viswanathan, J., Grossmann, I. E., 1990. A combined penalty-function and outer-approximation method for MINLP optimization. *Computers & Chemical Engineering* 14 (7), 769 – 782.
- Vitus, M., Pradeep, V., Hoffmann, G., Waslander, S., Tomlin, C., August 2008. Tunnel-MILP: Path planning with sequential convex polytopes. In: *Proceedings of the AIAA Guidance, Navigation and Control Conference*. Honolulu, Hawaii, pp. 1–13.
- Wächter, A., Biegler, L. T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming* 106 (1), 25–57.
- Wang, T., Le Maître, O. P., Hoteit, I., Knio, O. M., 2016. Path planning in uncertain flow fields using ensemble method. *Ocean Dynamics* 66 (10), 1231–1251.
- Westerlund, T., Lastusilta, T., 2017. AlphaECP. Available at <https://www.gams.com/latest/docs/solvers/alphaecp/index.html>.
- Westerlund, T., Porn, R., 2002. Solving pseudo-convex mixed integer optimization problems by cutting plane techniques. *Optimization and Engineering* 3 (3), 253–280.
- Wolsey, L., 1998. *Integer Programming*. Wiley.
- Yilmaz, N. K., Evangelinos, C., Lermusiaux, P. F. J., Patrikalakis, N. A., 2008. Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming. *IEEE Journal of Oceanic Engineering* 33 (4), 522–537.
- Zeng, Z., Lammas, A., Sammut, K., He, F., Tang, Y., 2014. Shell space decomposition based path planning for AUVs operating in a variable environment. *Ocean Engineering* 91, 181–195.
- Zeng, Z., Lian, L., Sammut, K., He, F., Tang, Y., Lammas, A., 2015. A survey on path planning for persistent autonomy of autonomous underwater vehicles. *Ocean Engineering* 110 (A), 303–313.