Mixed-integer programming in motion planning

Daniel Ioan, Ionela Prodan, Sorin Olaru, Florin Stoican, Silviu-Iulian Niculescu

Version of Record: https://www.sciencedirect.com/science/article/pii/S1367578820300754
Manuscript_ec83aea6eb99db02acbc0f9eb5187bc6

Mixed-integer Programming in Motion Planning

Daniel Ioan[a,*], Ionela Prodan[b], Sorin Olaru[a], Florin Stoican[c], Silviu-Iulian Niculescu[a]

[a] *Laboratory of Signals and Systems, Univ. Paris-Sud - CentraleSupelec - CNRS, Université Paris Saclay, France*
[b] *Univ. Grenoble Alpes, Grenoble INP (Institute of Engineering Univ. Grenoble Alpes), LCIS, F-26000, Valence, France.*
[c] *Department of Automatic Control and Systems Engineering, UPB, Romania.*

Abstract

This paper presents a review of past and present results and approaches in the area of motion planning using MIP (Mixed-integer Programming). Although in the early 2000s MIP was still seen with reluctance as method for solving motion planning-related problems, nowadays, due to increases in computational power and theoretical advances, its extensive modeling capabilities and versatility are coming to the fore and enjoy increased application and appreciation. This class of control problems involves, essentially, either a selection from a limited number of alternatives or a constrained optimization problem over a non-convex domain. In both situations, MIP has proven to be an efficient modeling technique as it will be shown in the present review paper. Furthermore, an emphasis is laid on the existing alternatives for implementation and on various experimental validations documented in the literature.

Keywords: MIP (Mixed-integer Programming), Motion planning, MPC (Model Predictive Control), Path following, Trajectory tracking, Task assignment, Collision avoidance

1. Introduction

This section introduces some general real-world problems/situations actually reducing to optimization problems which contain both integer and real variables and can be assimilated to open-loop or closed-loop control problems. Also, for a more general perspective it presents the class of problems which are tackled through MIP formulations.

1.1. Motivation and background

During more than 60 years of existence, the field of integer programming was extensively studied in the mathematics community due to its promising modeling capability and flexibility. In recent years (mainly the last two decades), mostly owing to the growing computational capabilities, the integer programming was brought to the attention of the control and robotic communities. There exists a broad variety of decision making problems that can be dealt through a MIP framework/approach.

MIP (Mixed-integer Programming) is a mathematical optimization problem in which some or all the variables are integers.

As its name indicates, MIP (Mixed-integer Programming) represents a mathematical optimization problem in which the objective is a linear, quadratic function or sometimes a more general criterion to be minimized or maximized, the constraints are linear (or non-linear) equalities (or inequalities) and there exist some (non-empty) subsets of integer and real variables playing the role of arguments (Jünger et al., 2009; Williams, 2013).

MIP is used to model several design problems and decision processes.

In a larger perspective, MIP is used to model several design problems and decision processes. Consider a typical logistics problem: an airport, which serves on average 50 flights per hour. The airport has only four runways. The task assignment problem that appears is to assign flights to runways, such that the runways are efficiently and uniformly used, while respecting some regulations (e.g., time separation between two consecutive landings/takeoffs or a minimum distance between two runways for simultaneous takeoffs). Another classical situation is described by the well-known *traveling salesman problem* and its variations, where the salesman wants to visit a number of customers in a minimal time or to cover a minimal distance. This has applications in several domains (e.g, overhauling gas turbine engines or X-ray crystallography Matai et al. (2010)). The above problems can be solved either intuitively, based on experience or by a trial and error method, but for critical situations an accurate mathematical formulation is necessary in view of certification. There are of course many other use cases which may employ MIP. For example, the

*Corresponding author
Email addresses: daniel.ioan@l2s.centralesupelec.fr (Daniel Ioan), ionela.prodan@lcis.grenoble-inp.fr (Ionela Prodan), sorin.olaru@l2s.centralesupelec.fr (Sorin Olaru), florin.stoican@upb.ro (Florin Stoican), silviu.niculescu@l2s.centralesupelec.fr (Silviu-Iulian Niculescu)

optimal power flow in the energy transmission networks Bahiense et al. (2001) or the transportation problems in a cluttered environment. Consider a boat moving within a fjord region. In order to safely arrive to its destination, the boat should follow a given path and avoid collision with the fjords. Thus, the feasible region is non-convex and should be efficiently described.

In the following, a brief classification of the types of problems, which can be modeled through MIP, is provided. A first class of problems is designated by those that involve integer quantities (i.e. discrete/quantified inputs or outputs), e.g. the *knapsack problem* (Williams, 2013). For this type of problem, MIP does not seem the obvious, natural, first choice, but, usually, it represents a better solution than a classical approach (e.g., use of the classical linear programming and approximate the provided solution to the nearest integer value).

Another MIP-modelisable class of problems is the one involving logical conditions, extensively treated in: Bemporad & Morari (1999); Smith & Taskin (2008); Williams (2013). For example, in Bemporad & Morari (1999), using the notations from Williams (2013), boolean algebra tools are aggregated, which allows to transform logical conditions on continuous variables into *mixed-integer inequalities* (linear inequalities involving continuous and binary variables).

As well, MIP is a popular modeling tool for sequencing and/or allocation problems (also, named combinatorial problems) (Smith & Taskin, 2008), including here the typical *task assignment* problem and its variations (e.g travelling salesman problem Dantzig et al. (1954)). This class of problems can be easily extended to networks (and graph theory) problems: *resource allocation on a PERT (Project Evaluation and Review Techniques) network* (Williams, 2013).

Lastly, but most importantly for this paper's purpose, MIP turned out to be a captivating method to model non-linearity (Bemporad & Morari, 1999; Vielma, 2015) and/or non-convexity (Prodan et al., 2015; Richards & How, 2005). A plethora of control engineering problem are naturally and intrinsically characterized by non-linearity and/or non-convexity. For this reason and due to the increasing interest in optimization-based control (Mayne et al., 2000), MIP has became an essential technique, which allows to include logical decisions and non-convex constraints in the optimization problem. Therefore, MIP's presence in control can be perceived in: piecewise-affine system identification (Bemporad et al., 2001; Roll et al., 2004), assignment problems (Alighanbari et al., 2003), persisting exciting control (Marafioti et al., 2012), control of hybrid systems (Bemporad & Morari, 1999), fault detection (Stoican et al., 2012) or motion planning (Prodan et al., 2015; Richards & How, 2002). As the title suggests, in the present review, the main objective is to identify and summarize the state of the art of MIP-based motion planning. Hence, in what follows, we place less emphasis on the other control areas employing MIP, even if throughout the paper we occasionally refer the interested readers to the references covering the other MIP-based control topics.

1.2. Contributions

This paper offers a detailed literature review of breakthrough research results and open issues in the field of multi-agent motion planning in a mixed-integer framework. This work can be employed to the benefit of both control and optimization research communities allowing to swiftly identify previous, timely and relevant research topics in the field and, at the same time, decreasing the time for literature review, although we acknowledge that the present review effort is not exhaustive but merely covers the experience of the authors in the last 10 years in these topics.

To the best of the authors' knowledge, there are no other exhaustive surveys in this topic although valuable attempts with different objectives are to be found in the literature. For instance, the tutorial session Richards & How (2005) offers a brief overview on how MIP can be employed for (feedback) control. As well, the paper of Smith & Taskin (2008) represents a concise introduction to the MIP modeling, providing the basic concepts regarding MIP formulations (the principles and some handful recipes) and, at the same time, discussing the techniques widely-used in the resolution of MIP problems. Moreover, the work "50 Years of integer programming 1958-2008: From the early years to the state-of-the-art" (Jünger et al., 2009) presents a historical perspective of the field of integer programming and discusses the theoretical, algorithmic and computational aspects of MIP throughout more than five decades of existence. Additionally, but with significant influence, there exist surveys, as, e.g., Vielma (2015), reviewing the advanced MIP formulations techniques, aiming to provide the guidelines for obtaining stronger and/or smaller formulations for a certain class of decision making problems. These works either limit themselves looking to the programming side, either to decision making in general or to a narrow control design topic. We henceforth decide to focus our review on control problems based on MIP and in particular those emerging from the active research field of motion planning which has a plethora of applications in automotive, robotics or multi-agent systems to mention just a few.

1.3. Outline

The remainder of the paper is organized as follows. We begin in Section 2 with a brief delineation of the evolution of MIP mathematical descriptions, providing the necessary prerequisites used in these formulations. Section 3 presents and details the standard MIP-based problems in motion planning, introducing the generic control strategy employed in such problems . Next, Section 4 makes the transition to multi-agent systems and to the formation control problems involving MIP. Section 5 gives a concise overview of the control architectures exploited in MIP-based navigation problems. Further, we proceed in

Section 6 with a brief presentation of both software and hardware implementations of MIP solutions in the literature. Section 7 succinctly presents the motion planning alternatives to MIP. We close our review with the conclusions and with some challenges in MIP-based motion planning and some suggestions for future research.

Notation

Throughout this paper we use the following standard notations. The logical operators: $\wedge$ (AND), $\vee$ (OR) and $\neg$ (negation). The Minkowski sum of two sets: $A \oplus B = \{x : x = a + b,\ a \in A, b \in B\}$. For $x \in \mathbb{R}^d$ we denote $\|x\|_Q^2 = x^\top Q x$. Given a compact set $S \subset \mathbb{R}^n$, $\mathcal{C}_X(S)$ denotes the complement of S over X, while $cl(S)$ is the closure of the subset S. For any polyhedron $P \subset \mathbb{R}^d$, $\mathcal{V}(P)$ is the (finite) set of its vertices. Any polytope (i.e. a bounded polyhedron) has a dual representation in terms of intersection of half-spaces or convex hull of extreme points: $P = \{x : s_i^\top x \leq r_i, \forall i\} = \{x : x = \sum \alpha_j v_j, \sum \alpha_j = 1, \alpha_j \geq 0, \forall j\}$. For a discrete set $\mathcal{I}$, $\#\mathcal{I}$ represents its cardinality.

2. MIP formulations

There are various MIP formulations which go back to the early '80 (or even earlier), each emphasizing the modeling capabilities of MIP. All these formulations share a common characteristic: the encoding of discrete decisions using binary and/or integer variables. These decisions appear in different problems, each using a certain formulation. This section provides a brief description of the most used MIP techniques and, concurrently, introduces some basic theoretical notions and tools.

Although generalized disjunctive programming (GDP) is not explicitly used in motion planning, we succinctly present it for the sake of the generality of modeling disjunctions through MIP.

2.1. Generalized Disjunctive Programming

Generalized Disjunctive Programming (GDP) has appeared for the first time in Raman & Grossmann (1994) in the research effort to use both quantitative and qualitative information in order to optimally solve chemical engineering problems. To this end, the qualitative information is represented using disjunctions and logic propositions. Compared to MIP, the GDP approach has a relatively more compact formulation, because the logical conditions are not transformed using boolean algebra and inequalities but rather in their natural (logic) form. In other words, GDP represents a combination of algebraic and logical equation as shown by the typical GDP from (1):

$$\min_x \quad f(x) + \sum_k c_k \tag{1a}$$

$$\text{s.t.} \quad r(x) \leq 0,\ x \in \mathbb{R}^n, \quad c_k \in \mathbb{R}, \tag{1b}$$

$$\bigvee_{j \in J_k} \begin{bmatrix} Y_{jk} \\ g_{jk}(x) \leq 0 \\ c_k = \gamma_{jk} \end{bmatrix}, \quad k \in K, \tag{1c}$$

$$\Omega(Y) = \text{true}, \quad Y_{jk} \in \{\text{true}, \text{false}\}, \tag{1d}$$

where $r(x)$ is a generic constraint, which does not depend on logical decisions; c_k cost variables, γ_{jk} fixed charges and $k \in K$ is the number of disjunctions. The logical function $\Omega(Y)$ corresponds to logical decisions in terms of boolean algebra and it is expressed in conjuctive normal form (CNF- "product of sums"). The idea of GDP can be summarized as follows: If $Y_{jk} = true$, then the constraints $g_{jk}(x) \leq 0$ and $c_k = \gamma_{jk}$ are imposed. Otherwise, they are ignored.

It is worth mentioning that the disjunction $\bigvee_{j \in J_k}$ represents, in fact, an exclusive relationship, in a disjunction k only a boolean variable Y_{jk} can be *true*. Some formulations state this requirement explicitly (as a separate constraint), others include it implicitly in the boolean function $\Omega(Y)$.

The problem (1) can be written as MIP, employing the binary variables $y_{jk} \in \{0, 1\}$ instead of the boolean ones Y_{jk} and replacing the constraint (1c) with:

$$g_{jk}(x) \leq M_{jk}(1 - y_{jk}) \tag{2a}$$

$$\sum_{j \in J_k} y_{jk} = 1, \quad k \in K \tag{2b}$$

where M_{jk} are "big-M" parameters, sufficiently large constants.

Remark 1. *The "big-M" formulation*[1] *(Williams (2013)) consists in choosing a very large positive number M which plays the role of a relaxation constant (Richards & How (2005); Vielma & Nemhauser (2011)). In this formulation, the binary variables play the role of a "switch", activating/deactivating the corresponding constraints. This is a powerful technique able to encode logical conditions but one should proceed with caution in selection of the value of M. A too large value may hinder the resolution of the MIP. For a detailed analysis regarding the "big-M" technique, we refer the interested reader to, e.g. Hooker (2011); Vielma & Nemhauser (2011).* ♦

Remark 2. *It is worth discussing the value of the coefficient M. For illustration, we consider a common case of modeling the logical implication: "$x > 0 \longrightarrow \alpha = 1$" for a variable x which evolves implicitly within a bounded*

[1] It is also referred to as "big-D" or "big-R".

domain[2] *of variation* $\mathbb{X} = \{x \mid |x| \leq \bar{x}\}$. *The "big-M" formulation simply reverts the logical implication in terms of an inequality* $x - M\alpha \leq 0$. *Intuitively, as long as* M *is chosen to satisfy* $M \geq \bar{x}$, *the implication follows naturally. Nonetheless, by considering a too large value of the coefficient* M *we increase the size of the searching domain. Thus, the idea is to take* M *as small as possible (but still large enough to fulfill its relaxation role). In our simple example a proper choice is:* $M = \max_{x \in \mathbb{X}} x$, *i.e.,* $M = \bar{x}$. ♦

Moreover, the cost is reformulated by rewriting each cost variable c_k as the product $\gamma_{jk} y_{jk}$. As well, the condition (1d) is written in an algebraic form $Ay \leq a$.

The apparent advantage of the GDP over MIP is diminished because all the existing GDP-dedicated solvers are based on a MIP reformulation. It is enough in general to assume that a direct MIP modeling of the problem at hand could lead to a better and more compact model. In spite of eluding GDP in such a way, there is another possibility to benefit from GDP. The problem can be modeled as a GDP and the resulting MIP reformulation can be parsed using one of the several tools, as detailed in (Williams, 2013).

2.2. MIP - a geometric viewpoint

Complex control synthesis or design leads to a constrained optimization problems and whenever a non-convex feasible domain needs to be handled, the ability of MIP to mathematically formalize disjunction constraints can be exploited. An active research topic was the MIP efficient description of these non-convex regions. Preliminary results make use of hyperplane arrangements to characterize these regions. Hereinafter, we briefly recall these results which involve besides hyperplane arrangements some set-theoretic notions. It is beyond the scope of this review to provide an exhaustive presentation of these techniques and refer the interested readers to complementary materials referenced in this manuscript(e.g., Vielma (2015); Vielma & Nemhauser (2011) or Prodan et al. (2015)).

Most sets involved in MIP formulations are polyhedral ones due to their ability to provide a geometrical representation of the linear constraints. In what follows, we use the notion of polytope which is a bounded polyhedral set and has a dual representation in terms of intersection of half-spaces or convex hull of extreme points: $P = \{x : s_i^\top x \leq r_i, \forall i\} = \{x : x = \sum \alpha_i v_i, \sum \alpha_i = 1, \alpha_i \geq 0\}$.

Consider a finite collection of hyperplanes from $\mathbb{R}^d$

$$\mathcal{H}_i = \left\{x \in \mathbb{R}^d : \; h_i x = k_i\right\}, \; i \in \mathcal{I} \tag{3}$$

where $\mathcal{I} \triangleq \{1 \dots N\}$, and $(h_i, k_i) \in \mathbb{R}^{1\times d} \times \mathbb{R}$.

[2]Due to the physical limitations.

Each of these hyperplanes divides the space in two disjoint[3] regions:

$$\mathcal{R}_i^+ = \left\{x \in \mathbb{R}^d : \quad h_i x \leq \quad k_i\right\}, \tag{4a}$$

$$\mathcal{R}_i^- = \left\{x \in \mathbb{R}^d : \; -h_i x \leq -k_i\right\}. \tag{4b}$$

A polytope P is a bounded intersection of these half-spaces[4]:

$$P = \bigcap_{i \in \mathcal{I}} \mathcal{R}_i^-, \tag{5}$$

Its complement (up to its relative interior, see also footnote 1) $\mathcal{C}(P) \triangleq cl(\mathbb{R}^d \setminus P)$ over $\mathbb{R}^d$ represents the union of all regions covering the entire space except P. Recalling the fact that the union and intersection operators interchange under complement operation, we write:

$$\mathcal{C}(P) = \mathcal{C}\left(\bigcap_{i \in \mathcal{I}} \mathcal{R}_i^-\right) = \bigcup_{i \in \mathcal{I}} \mathcal{C}\left(\mathcal{R}_i^-\right) = \bigcup_{i \in \mathcal{I}} \mathcal{R}_i^+. \tag{6}$$

The region (6) is a union of convex sets[5] and for a tractable characterization we may employ mixed-integer techniques. Hence, the binary variables $(\alpha_1 \dots \alpha_N) \in \{0,1\}^N$ are introduced to obtain the representation of the polytopic set in the extended space of state and auxiliary binary variables:

$$h_i x \leq k_i + M\alpha_i, \; i \in \mathcal{I}, \tag{7a}$$

$$\sum_{i \in \mathcal{I}} \alpha_i \leq N - 1, \tag{7b}$$

where M is a big-M coefficient as in Remark 1.

The requirement that a point is outside of the polytopic set P is converted to the condition that the point has to be within the complement of at least one of the half-spaces defining the polytope.

Remark 3. *The conditions* (7a)–(7b) *describe the region* (6) *via a proper combination of binary variables. For instance, the region* $\mathcal{R}_i^+$ *is expressed by* (7a) *with the following binary variables:*

$$(\alpha_1 \dots \alpha_N) = (1 \dots 1 \underbrace{0}_{i} 1 \dots, 1). \tag{8}$$

When a binary variable takes the value "1" the associated inequality describes, for the limit case $(M \to \infty)$ *the entire domain* $\mathbb{R}^n$. *Hence, the condition* (7b) *is necessary in order to ensure that at least one of the binary variables is "0" and, thus, that at least an inequality remains active.* ♦

[3]The relative interiors of these regions are disjoint, but their closures have as a common boundary the hyperplane $\mathcal{H}_i$

[4]The sign "-" was chosen for the simplicity of notation, any other feasible combination of signs from (4a)-(4b) could be chosen in order to describe the polytope P.

[5]In general (6) is non-convex with notable exceptions: the complement of an empty set, the complement of the unconstrained space or the complement of a half-space (possibly in a non-minimal representation).

The above reasoning is not restricted to the complement of one convex polytopic region and can be generalized for the case in which the interdicted region is a union of polytopes as in (Stoican et al., 2013). Explicitly, the non-convex region is characterized by:

$$-h_{i_l}x \leq -k_{i_l} + M\alpha_{i_l},\ \forall i_l \in \mathcal{I}_l, \tag{9a}$$

$$\sum_{i_l \in \mathcal{I}_l} \alpha_{i_l} \leq \#\mathcal{I}_l - 1. \tag{9b}$$

with $\mathcal{I}$ in (3) redefined as a union of discrete intervals[6]: $\mathcal{I} = \bigcup_l \mathcal{I}_l$.

Remark 4. *As stated in Remark 1, choosing the value of the "big-M" constant may lead to increasing complexity in terms of the computational effort. It is straightforward that the constraint (9a) cannot describe the entire $\mathbb{R}^n$ for a $\alpha_i = 1$, but rather a domain including the region of interest, $\mathbb{X}$. Assuming, for instance, that the union of interdicted regions lies in a bounded cluttered environment $\mathbb{X}$, the value of M is obtained via the following LP:*

$$M = \max_i M_i, \quad M_i = \max_{x \in \mathbb{X}} (k_i - h_i x).$$

Moreover, we may consider a different value of M for each half-space, that is, M_i.

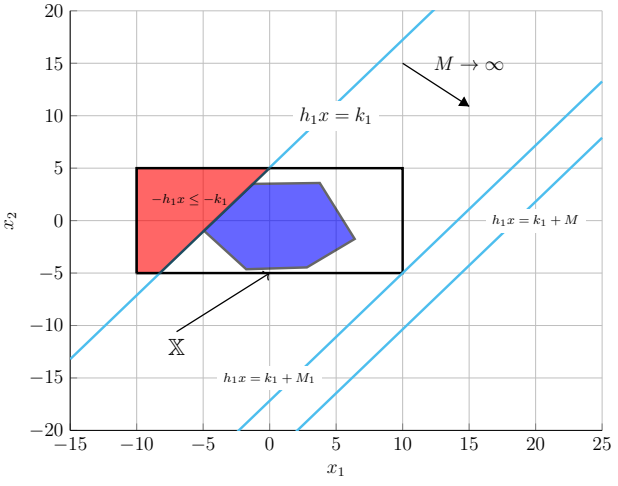

Figure 1: Illustration "big-M", M computed as in Remark 4. While the constraint is active (i.e., $\alpha_1 = 0$ in (9a)), the resulting inequality: $-h_ix \leq -k_i$ describes the red region. Once $\alpha_1 = 1$ (inactive/relaxed constraint), the selection of M gives the measure of relaxation. Thus, the idea is to relax sufficiently such that the remainder of the constraints is not affected.

The formalism can be further extended and the space is partitioned into cells using the hyperplane arrangement notion (Ziegler (2012)).

[6] Sets of indices.

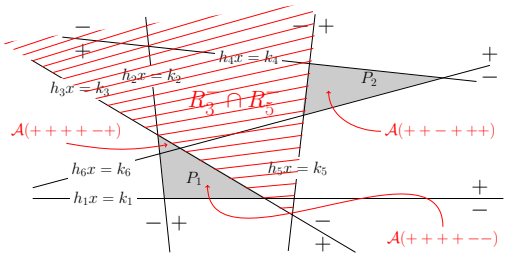

Figure 2: Polytopic obstacles and hyperplane arrangements.

Definition 1 (Hyperplane arrangements). *The collection $\mathbb{H}$ partitions the space into a union of disjoint cells $\mathcal{A}(\sigma)$, characterized by a sign tuple $\sigma \in \{-,+\}^N$:*

$$\mathcal{A}(\sigma) = \bigcap_{i \in \mathcal{I}} \mathcal{R}_i^{\sigma(i)}. \tag{10}$$

The hyperplane arrangement of cells covering the entire space is described by the collection of all feasible sign tuples:

$$\mathcal{A}(\mathbb{H}) = \bigcup_{l=1\ldots\gamma(N)} \mathcal{A}(\sigma_l), \tag{11}$$

where $\mathbb{H} = \{\mathcal{H}_i\}_{i\in\mathcal{I}}$, $\sigma_l \in \{-,+\}^N$ is the sign tuple resulting from a non-empty intersection of half-spaces and $\gamma(N)$ is the number of feasible cells. ♦

Consider a collection of obstacles:

$$\mathbb{P} = \bigcup_{j=1}^{N} P_j$$

Briefly, gathering the collection of associated support hyperplanes defined as (3) we reach the hyperplane arrangement (11). Labeling the feasible cells (10) into interdicted $\Sigma_{\mathbb{P}} = \{\sigma : A(\sigma) \cap \mathbb{P} \neq \emptyset\}$ or allowed $\Sigma_{X\setminus\mathbb{P}} = \{\sigma : A(\sigma) \cap \mathbb{P} = \emptyset\}$ one can obtain a mixed-integer characterization of the feasible domain:

$$h_ix \leq k_i + M(1-\alpha_i), \tag{12a}$$

$$-h_ix \leq -k_i + M\alpha_i, \tag{12b}$$

$$\sum_{\sigma_l(i)='+'} (1-\alpha_i) + \sum_{\sigma_l(i)='-'} \alpha_i > 0, \tag{12c}$$

$$\forall \sigma_l \in \Sigma_{\mathbb{P}} \tag{12d}$$

Even though the construction presented above is a generic one, the binary part of the representation is substantially large as we have a binary variable for each region (4b). There exists a variety of technical procedures like cell merging (Prodan et al., 2015) or logarithmic formulations (Vielma & Nemhauser, 2011) which may be employed to reduce the complexity of the formulation, a large number of obstacles and/or agents may still lead to an impractically large number of binary variables.

The representation of the interdicted domains is not limited to the polyhedral sets. For instance in Earl &

D'Andrea (2005a), circular obstacles are considered as the one depicted in Fig. 3, each being determined by its radius R_o and by the coordinates of its center (x_o, y_o):

$$\mathcal{O} = \{(x, y) : (x - x_o)^2 + (y - y_o)^2 \leq R_o^2\} \quad (13)$$

However, to incorporate the corresponding avoidance constraints, the obstacles as in (13) are approximated with polygons (polyhedra in $\mathbb{R}^2$, having p vertices). Hence, each interdicted zone is described by a set of K_p inequalities:

$$\tilde{\mathcal{O}} = \{(x, y) : (x - x_o) \sin \frac{2\pi m}{K_p} + (y - y_o) \cos \frac{2\pi m}{K_p} \leq R_o, \ \forall m = 1, \ldots, K_p\} \quad (14)$$

Using the same reasoning as in (7) and the big-M technique, the avoidance constraints are formulated as:

$$(x - x_o) \sin \frac{2\pi m}{K_p} + (y - y_o) \cos \frac{2\pi m}{K_p} \geq R_o - M\beta_m, \forall m \quad (15a)$$

$$\sum_{m=1}^{K_p} \beta_m \leq K_p - 1. \quad (15b)$$

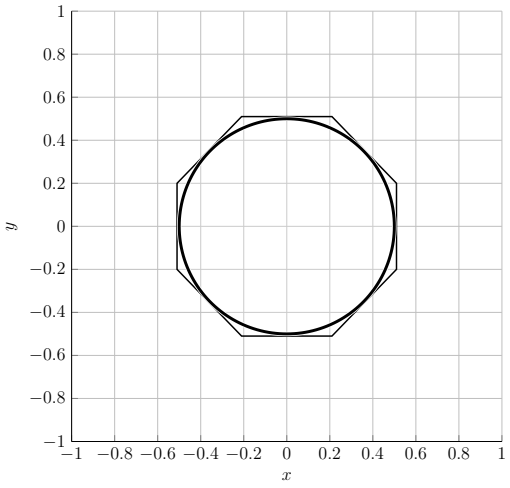

Figure 3: A circular obstacle and its polytopic approximation.

Remark 5. *Besides the polytopic representation of the obstacles, some others representations are considered in the literature. For instance, Ioan et al. (2019) considers additional structural constraints by the use of zonotopic sets (a particular class of polytopes with increased symmetry). The use of a such representation leads to a trade off between the complexity and the fidelity of the representation.* ♦

It is worth to mention that restrictions within the state space can be directly treated in the dynamical system modeling. In particular is the case of the hybrid systems where switching (a disjunctive type of selection in between modes of functioning) include binary variables or discrete alternatives. The Piece-Wise Affine (PWA) models represent for example an integrated approach, where non-convex domains and local dynamics find a unitary representation. This unitary modeling needs to be differentiated from the classical motion planning framework where the dynamics are defined globally and the control decisions need to satisfy the non-convex feasible domain or disjunctive constraints. Because our review is focused rather on classical motion planning techniques, we mention some relevant papers, e.g. Bemporad & Morari (1999); Vielma & Nemhauser (2011), which provide an extensive presentation of the PWA modeling tools.

2.3. GDP formulation for region description

In order to highlight the relationship between GDP and MIP, consider the following illustrative example. We consider four supporting hyperplanes describing the interdicted region and use a GDP formulation.

Hence, corresponding to each hyperplane $i \in \{1, 2, 3, 4\}$ we have the next disjunction:

$$\begin{bmatrix} Y_{i1} \\ g_i(x) \leq 0 \end{bmatrix} \vee \begin{bmatrix} Y_{i2} \\ g_i(x) \geq 0 \end{bmatrix}$$

$$Y_{i1} \vee Y_{i2} \leftrightarrow Y_{i2} = \neg Y_{i1}$$

where $g_i(x) = x_1 \pm x_2 \pm 6$ and we associate: $Y_{i1} = true$ with $\mathcal{R}_i^-$ (the below half-space) and $Y_{i2} = true$ with $\mathcal{R}_i^+$ (the above half-space). The interdicted and the allowed region (both illustrated in Fig. 4) can be described using the following boolean function:

- convex region: $\Omega(Y) = \neg Y_{11} \wedge Y_{21} \wedge \neg Y_{31} \wedge Y_{41}$
- non-convex region: $\Omega(Y) = Y_{11} \vee \neg Y_{21} \vee Y_{31} \vee \neg Y_{41}$

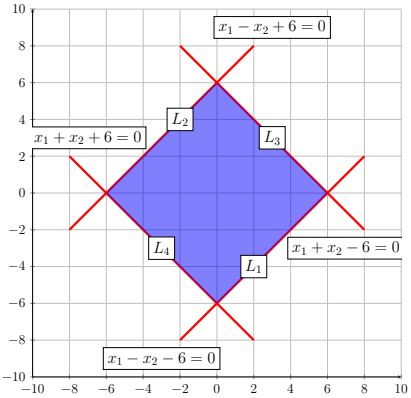

Figure 4: The illustrative example for GDP formulation. $\{L_i\}_{i=1:4}$ hyperplanes

The set-theoretic tools involved in MIP formulations were briefly sketched. For a more extensive presentation

the interested reader is referred to the well-known works about polyhedral and hyperplane arrangement notions Kuhn (1998); Ziegler (2012) or to the more recent monographies, as Prodan et al. (2015), which offer a more detailed mathematical descriptions of the concepts presented here.

3. MIP in motion planning

The presence of MIP in motion planning can be originated by algebraic or geometrical approaches. The former relates to situations and circumstances, in which logical decisions are involved, e.g. the selection from a priori known set of possible alternatives. The later usually refers to the ability of MIP to describe non-convex constraints.

Prerequisites

Ahead of proceeding with typical MIP implementations in motion planning, some clarifications are in order. Generally, for motion planning there exists plenty of applications that will be treated in detail in Section 6. An important aspect is that all those applications have a dynamical behavior of a point of mass in 2D or 3D spaces governed by dynamical mathematical models (either linear or nonlinear).

Up front, let us consider a generic[7] model of the controlled dynamical system designated by an ordinary differential equation (ODE):

$$\dot{x}(t) = f\left(x(t), u(t), w(t)\right) \tag{16}$$

where $x(t) \in \mathbb{R}^{n_x}$ denotes the state vector, $u(t) \in \mathbb{R}^{n_u}$ the control input vector, and $w(t) \in \mathbb{R}^{n_w}$ the disturbance[8]. The mapping $f(\cdot,\cdot,\cdot) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$ is a continuous function admitting an equilibrium point (i.e., $f(\bar{x}, \bar{u}, 0) = 0$; without any loss of generality we may assume $f(0, 0, 0) = 0$).

Remark 6. *The model (16) describes a dynamic in continuous time. As will be detailed below, a considerable part of the literature is based on discrete time dynamics. Hence, we need to consider, concurrently, keeping similar notations, the discrete time counterpart:*

$$x(k+1) = f\left(x(k), u(k), w(k)\right) \tag{17}$$

The correlation between (16) and (17) is done using one of the various discretization techniques (Sontag, 2013). ♦

In what follows, a relevant feature of the previously-mentioned models is the way of selecting the input variables and consequently the decision domain. In the literature we can distinguish these choices with respect to the linearity/ non-linearity of the models. Thus, for linear models the references concentrate towards the models in which the state is composed by the position and the speed of the agent and the input is represented by the acceleration of the agent, see, for example: Bellingham et al. (2002a); Liu et al. (2017); Papen et al. (2017); Richards & How (2002); Yu et al. (2014). This type of model implies actuation on each direction and is usually based on the kinematics equations describing the behavior of the controlled systems. Indeed, there are several variations of this model derived from more complex modelisation techniques that do not represents the topic of this paper. Regarding the nonlinear models, the favored variant consists of models, in which the components of the state are the position and heading angle, and the input is represented by speed and steering angle,e.g., (Rey & Hijazi, 2017).

A usual approach is to do trajectory tracking which hides the nonlinearities induced by obstacle avoidance but moves the difficulty in the trajectory design step. This can be mitigated through the use of flatness (quite popular approach in this setting) have it. Still, computing offline a trajectory (or a path) assumes a known environment which may not be the case. Either linear or nonlinear, the models involved in motion planning are characterized by flatness. Generally speaking, a system is flat if there exists a set of variables (so-called, flat outputs) such that we can determine all states and inputs based on these variables without integration Murray et al. (2009). In Martin et al. (2002), the interested reader can find a detailed presentation of this type of system and how their properties are used in trajectory tracking or motion planning.

From the control perspective, the earliest methodologies have been based either on optimal control or on nonlinear programming, but the controlled systems have operated in obstacle-free environment. Recently, this type of environment is no longer an appropriate assumption from practical reasons (especially, in a multi-agent context). Hence, the control community has had to propose some novel techniques or to adapt the old ones. First adjustments consisted in restricting the controlled system to track predetermined trajectories around the obstacles, but the the resulting control strategies were working only for particular systems and environments (LaValle, 2006).

MIP framework is *frequently* encountered within optimization-based control

While the focus of this manuscript is MIP and its various implementations, ultimately, the obtained formulation comes as the result of optimization-based control strategy. Thus, in order to provide a general overview of the MIP-based motion planning problems we briefly introduce some notions from optimal control theory.

Essentially, the optimal control problem (Diehl, 2014; Kirk, 2012) consists in finding a control input $u^\star \in \mathbb{R}^{n_u}$ such that the system (16) follows a trajectory $x^\star(t) \in \mathbb{R}^{n_x}$,

[7] There exists in the literature several variation of models, but most of them can be written in the form designated by (16).

[8] Usually, the disturbance is bounded $w(t) \in \mathcal{W}$, and $\mathcal{W} \subset \mathbb{R}^{n_w}$ is convex and compact set.

Table 1
Classification of the research in motion planning using MIP

Applications	References (alphabetical order)
Task assignment	Alighanbari et al. (2003); Bethke et al. (2008); Chen et al. (2016a); Chen & Wang (2005); Fayazi et al. (2017); Haghighi et al. (2016); Liu et al. (2017); Richards & How (2005); Wang et al. (2015b); Zhang et al. (2017a,b)
Path planning	Janeček et al. (2017a); Ragi & Mittelmann (2017); Richards & How (2005, 2002); Schouwenaars (2006); Wang et al. (2015b)
Trajectory planning	Ballesteros-Tolosana et al. (2017); Cetin et al. (2007); Deits & Tedrake (2015b); Earl & D'Andrea (2002, 2005a,b); Fayazi et al. (2017); Huang & Peng (2017); Zhang et al. (2017b)
Obstacles and collision avoidance	Bali & Richards (2018); Bellingham et al. (2002a); Haghighi et al. (2016); Janeček et al. (2017a); Mukai et al. (2017); Rey & Hijazi (2017); Richards & How (2002); Richards & Turnbull (2013); Wang et al. (2015b)

minimizing a specific criterion:

$$\min_{x(t),u(t)} J = \min_{x(t),u(t)} \quad H\left(x(t_f)\right) + \int_{t_0}^{t_f} L\left(x(t),u(t)\right) dt \tag{18a}$$

$$s.t. \quad \dot{x}(t) - f\left(x(t),u(t),w(t)\right) = 0 \tag{18b}$$

$$x(t_0) - x_0 = 0 \tag{18c}$$

$$g\left(x(t),u(t)\right) \leq 0 \tag{18d}$$

where $t \in [t_0, t_f]$, $(t_f - t_0)$ denotes the time horizon length, x_0 the initial state, and $g(\cdot,\cdot)$ is a function incorporating the physical constraints (e.g., terminal or stage constraints of the controlled system). The OCP (optimal control problem) (18) can take various forms according to i) the choice of functions $H(\cdot)$ and $L(\cdot,\cdot)$, e.g. minimum time problem or terminal control problem Kirk (2012); ii) the selection of the horizon ($[t_0, t_f]$), e.g., Garg et al. (2011).

Standard MIP-based problems in motion planning

The motion planning problems (LaValle, 2006) are well established and have been studied in time optimal control, nonlinear control, stabilization, reachability and other related topics. To review all these topics is beyond the scope of this manuscript, but we retain those of them in which the presence of alternatives or discrete decisions making is explicitly considered. The emphasis is placed on the specific applications where non-convex feasible domains appear and are subsequently encoded by mixed-integer techniques.

In this section are presented the motion planning subareas, which can be efficiently formulated through a MIP approach:

i) **Task assignment** (TA) is the strategic decision of allocating one objective to a particular subsystem (*who-goes-where?*), these objectives can be interchanged and their suitability is measured by a specific criterion. TA is a discrete decision-making process where the number of alternatives within the pairs (resource objective) association is countable.

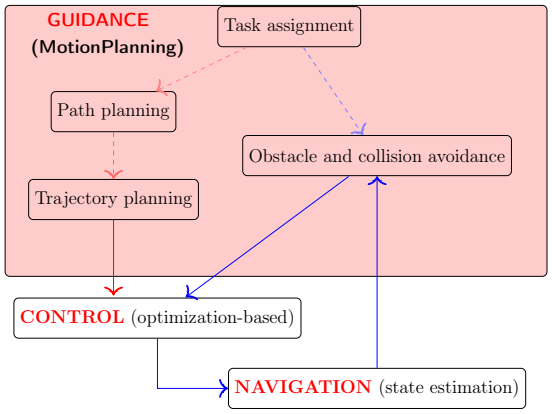

Figure 5: Motion planning sub-areas and the link among them

ii) **(Collision-free) Path planning** is the construction of a route in the position space without an explicit parametrization in time and without explicitly considering the dynamics.

iii) **(Collision-free) Trajectory planning** represents the problem of constructing a function which associates a time interval to a path. It takes into account the particularities of the agent's dynamics and the resulting function is generated in an open-loop manner.

iv) **(Online) Obstacle and collision avoidance** is the problem of finding input control signals which

minimize a performance criterion as in (18a) while simultaneously avoiding collisions and using a closed-loop strategy.

We have graphically illustrated Fig. 5 the functional relationship among the items described above. A noteworthy aspect is that for the task assignment and path planning sub-areas the particularities of the dynamic have, generally, almost no influence in problem statement, in contrast with the other sub-areas. However, there are situations, in which the dynamic impacts on the definition of static costs. Besides, TA and path planning are not tackled in most of the motion planning literature as stand-alone topics, but together with or as part of the others sub areas. As depicted in Fig. 5 (the dotted lines), path planning is usually implicitly included in trajectory planning and TA is either a part of obstacle avoidance or included in path planning.

As well, collision-free trajectory planning and collision avoidance represent dual approaches, the difference between them being the manner of interaction with navigation and control. While the obstacle avoidance intervenes into the navigation-control loop, the trajectory planning only generates the reference (to be tracked) for this loop.

As a side remark, a considerable part of the control community includes also the formation control in motion planning sub-areas, although it represents, in fact, a control level which gathers all the above sub-areas of motion planning, and, as TA, it makes sense only in a multi-agent context. Therefore, we separately treat this topic in Section 4, where connectivity maintenance problems in multi-agent systems are presented.

Remark 7. *Before detailing the motion planning problems, it is necessary to highlight the differences between the path and trajectory planning. There exists a wealth of manuscripts which treat this topic in detail (see, e.g., Beard & McLain (2012) or Yang et al. (2014)). The choice between these two methodologies depends on particularities of the controlled systems, e.g. for a fixed wing UAV, a trajectory planning methodology can have some undesired consequences (Beard & McLain, 2012). These particularities are broadly presented and discussed in Section 6. However, there does not exist in the literature a generally accepted differentiation between these two concepts. Hence, for the ease of presentation we opted to consider the definitions as in ii), and iii) respectively.* ♦

In Table 1 we depicted the classification of the references with respect to the motion planning problem predominantly treated therein.

Remark 8. *In general, a pre-condition of the MIP approaches is the existence of an awareness map which entails the global knowledge about the environment. Thus, representation of the environment has a significant impact on the performances of the motion planning strategies (except the task assignment problem where the impact is not crucial).* ♦

3.1. Task assignment

In what follows, we present a standard MIP formulation for the task assignment problem, emphasizing its importance in the context of motion planning for a multi-agent system.

Task assignment refers to a strategic decision of allocating one objective to a particular subsystem (*it responds to the question "who-goes-where?"*).

At large, a task assignment algorithm must provide an assignment which optimizes a specific criterion. Consider, for instance, the undesired occurrence of an earthquake, which affects 10 buildings. An emergency committee must assign for each building a rescue team, in order to minimize the intervention time, taking into account the number of rescue teams and their necessary time to arrive at respective buildings. This example can be easily reformulated for the case of target tracking in a multi-agent system.

More generally, let us consider N agents and M targets. In order to minimize the amount of time to reach all the targets, we should optimally assign agents to targets. To this end, we define a binary variable x_{ij}, which equals one if the agent i visits the target j and zero otherwise. Also, we consider a cost associated to each combination agent-target: c_{ij}. We take into account that each agent must attain at least one target, and, at the same time, each target must be reached by an agent (constraints (19b) and (19c), respectively). Thus, we obtain a standard MIP formulation of task assignment problem, described in (19).

$$\min_{x_{ij}} \sum_{i=1}^{N}\sum_{j=1}^{M} c_{ij}x_{ij} \tag{19a}$$

$$s.t. \quad \sum_{i=1}^{N} x_{ij} \geq 1,\ \forall j \tag{19b}$$

$$\sum_{j=1}^{M} x_{ij} \geq 1,\ \forall i \tag{19c}$$

$$x_{ij} \in \{0,1\}, \quad \forall i,j \tag{19d}$$

Since we have an integer requirement on the variables x_{ij}, the problem (19) belongs to the class of ILP (integer *linear* programming) (a subclass of MIP, where all the variables are integer.) A noteworthy aspect is given by a special characteristic of (19): solving its LP relaxation[9] allows obtaining the solution of ILP original problem. Nonetheless, if the task assignment problem is part of an enhanced control design, this property will be lost and the problem should be solved in the generic class of (M)ILP. For instance, let us define a variable t_{ij} which represents the necessary time for agent i to reach the target

[9] We replace (19d) with the *relaxation* $x_{ij} \in [0,1]$

j. We impose that the total time to not exceed a certain value (20).

$$\sum_{i=1}^{N}\sum_{j=1}^{M} t_{ij}x_{ij} \leq T \tag{20}$$

In specific situations, LP relaxations may lead to an integer solution.

Furthermore there are many others variations of the task assignment problem (19). For example, in Smith & Taskin (2008) the authors considered a similar problem but they introduce in the cost a non-linear term corresponding to the case when an agent has to attain more than one target. Using some refinements (detailed therein) they posed the problem as a MILP, even if the formulation is somewhat more complicated compared with (19).

In Alighanbari et al. (2003) and in Bethke et al. (2008) the task assignment problem for a fleet of UAV is extensively treated. The initial requirements are more complicated than the formulation in (19) in the sense that the targets are transformed in waypoints and each agent must visit a sequence of these. Moreover, some restrictions as the existence of coupled tasks or of tight timing constraints increase significantly the complexity of the formulation.

A further example is found in Chen et al. (2016a) where a variation of the task assignment problem is used to provide higher level control logic to synthesize a cooperative safety controller. Actually, the objective is to group the agents in pairs such that the corresponding avoidance maneuvers may not lead to a dangerous configuration for the other agents. Thus, the theoretical guarantees given by the lower level controller are extended for a greater number of agents.

Similar approaches are used for optimal scheduling at intersections where a prioritization among agents is necessary. The scheduling problem is a variation of the task assignment problem, while the tasks are assimilated to time slots or time instants. This variation is one of the few resource allocation problems in which the dynamic behaviour is not completely ignored. For example, in Fayazi et al. (2017) the agent's dynamic impacts on the objective function by introducing a non-linearity in the cost. This non-linearity is eliminated using a MIP formulation as in Smith & Taskin (2008).

MILP formulations are NP-hard (Non-deterministic Polynomial-time hard) (Van Leeuwen & Leeuwen, 1990).

Given the fact that MILP formulations are NP-hard, the control community has a preference for solving this type of problems (task assignment or resource allocation) through heuristic methods, e.g., in Alighanbari et al. (2003) a comparison between the Tabu search and the MIP formulation is done, and, for higher dimensions the Tabu search exhibits better performance. However, the tangible theoretical advantage of the MIP formulation (19) is that it guarantees the global optimum. This has a substantial importance in some critical situations.

Remark 9. *In a similar manner as the task assignment, the notorious travelling-salesman problem (TSP) appears first time in Dantzig et al. (1954) posed as MILP. For the classical TSP the formulation (19) is used by converting the constraints (19b) and (19c) into equalities and imposing that $N = 1$ and $M \geq 2$.* ♦

Besides task assignment, a plethora of resource allocation problems can be readily formulated as MIP. Many of these problems are used in several applications within related motion planning domains such as aircraft maintenance (or repair) Bajestani & Beck (2013), crew scheduling(and flight retiming) Mercier & Soumis (2007) etc.

There exist approaches building on the idea of state space (workspace) organization in terms of available resources. More precisely, these approaches partition the environment in equally-shaped cells and the cells are viewed as a shared resource. In this manner, the collision and obstacle avoidance problems are transformed in resource allocation with mutual exclusion.

The collision avoidance problems can be viewed as resource allocation with mutual exclusion.

For instance, in Wang et al. (2015b) the collision avoidance constraints are given by the fact that each cell cannot be visited by two distinct agents at the same time, i.e. the agents do not use the same resource simultaneously. The formulation is an extension of the task assignment problem (19), the novelty comes from the evaluation and resolution of the MIP in an online manner.

Another reference using this "resource allocation" approach is Haghighi et al. (2016) where a robotic swarm has the objective of respecting specific and complex patterns in a bidimensional workspace. These patterns are described using the spatio-temporal logic (SpaTeL). SpaTeL consists in propositions indicating the number of agents which can access a cell at a certain time. These SpaTeL formulas are converted in mixed-integer linear constraints, resulting a formulation comparable with the task assignment (19) but more complex. A similar approach is employed in Liu et al. (2017), but the difference is given by the inclusion of the communication constraints in the motion planning problem under the form of signal temporal logic (STL).

3.2. Path planning

As stated earlier in Section 3, a path planning strategy involves the generation of a route without an explicit parametrization[10]. Further, this route is called the planned path.

Path planning refers to the construction of a route in the position space without an explicit parametrization in time and explicitly considering the system dynamics.

[10] in the sense that the time is not explicitly associated

Definition 2. *A path between two points x_0, x_f from the navigation space is given by a map $\gamma : [0,1] \to \mathcal{X}$ with*

$$\gamma(0) = x_0,\ \gamma(1) = x_f \tag{21}$$

♦

There are in the literature two major classes of methods to describe the function γ. The first one, the explicit methodology, consists in providing the value of γ for the entire interval $[0,1]$ Janeček et al. (2017a). The second class of methods is the most used alternative, the function has to take their values from the given set of, so-called, waypoints Afonso et al. (2013).

Remark 10. *The waypoints represent either the nodes of graph in a dynamic-free setting, or the boundary conditions for intervals of time.* ♦

For further use, we define the set of all waypoints describing the planned path: $\mathcal{W} = \{x_{w_0} \dots x_{w_{N_w}}\}$ and consider an additional condition to Def. 2: $\forall x_{w_k}, \exists! \theta_k \in [0,1]$ such that $\gamma(\theta_k) = x_{w_k}$.

The methodologies involved in the selection and determination of the waypoints are outside the scope of this paper, although they are succinctly presented in Section 7. In the path planning problem, MIP is not employed in the generation of the path, but rather to address the problem of temporal distribution of the waypoints (i.e., their ordering along the path), satisfying some criteria. In other words, MIP plays an instrumental role for optimizing a path obtained with one of the well-established techniques from the robotic field (e.g., sampled based methods - see Section 7). The criterion of this optimization procedure has to include/cover costs given by the physical and "economical[11]" limitation.

One of the most used approaches treating this type of problem is based on a particular case of the task assignment (19), the travelling-salesman problem (TSP). Under this form we can identify the MIP usage in path planning and a classical example in this sense is to be found in Richards & How (2002); Schouwenaars (2006). Therein, they deal with a robot which should visit N_w waypoints while minimizing the cost of the maneuver. Hence, the MI constraints which enforce the visiting of the considered waypoints are the following:

$$\forall i \in \{1, \dots, T\},\ \forall k \in \{1, \dots, N_w\}, \tag{22a}$$

$$\|x_i - x_{w_k}\| \leq M(1 - b_{ik}), \tag{22b}$$

$$\sum_{i=1}^{T} b_{ik} = 1,\ \forall k \text{ and } b_{ik} \in \{0,1\}, \tag{22c}$$

where T is the number of time steps, and b_{ik} is a binary variable that indicates whether or not the waypoint k was visited at the time step i. The constraint (22c) ensures that each of the waypoints is visited once by the agent. The ordering of the waypoints along the interval depends on the selected criterion (e.g., the minimum time to visit all the waypoints). In practice, the attainability of the waypoints is affected by various factor and, thus, in most of the situations, reaching a vicinity of the waypoint is a reasonable objective. In such cases, (22) needs to be adapted,e.g., the constraint (22b) is replaced by $H_k(x_i - x_{w_k}) \leq h_k + M(1 - b_{ik})$, where H_k and h_k are given by the supporting hyperplanes describing the considered vicinity region.

Remark 11. *In the majority of sampled-based methods LaValle (2006), the problem of finding the shortest path is tackled using one of the various algorithms (Dijkstra, $A^\star$ are common choices). However, in Taccari (2016) several MIP formulations for the elementary shortest path problem are introduced as extensions of TSP.* ♦

Besides the approach relying on (22), there exist a few others works, which treat MIP-based path planning in a different manner. For instance, in Vitus et al. (2008), a Tunnel-MILP approach is employed. Basically, this algorithm divides the global motion planning problem in three main tasks. The first task consists in finding a path as in Def. 2, while ignoring the vehicle's dynamics constraints. Next, the path is used along with a convex decomposition of the space to generate a sequence of N_R convex polytopes from the start to the goal: $F_i = \{x \in \mathbb{R}^2 : A_i x \leq b_i\},\ \forall i \in \{1, \dots, N_R\}$. In the third task, this sequence (a tunnel of polytopes) is used to constrain the position of the agent $p(t)$ to an optimal and dynamically feasible route from the initial point to the goal inside the tunnel. More precisely, a MILP problem is employed, forcing the agent to remain in one of the regions of the tunnel at all times. This leads to some OR-constraints[12], similar with the GDP formulation (1), which can be readily formulated using MI techniques:

$$\bigvee_{i \in \{1,\dots,N_R\}} A_i p(t) \leq b_i,\ \forall t \tag{23}$$

where $p(t)$ is the position of the considered agent.

The same idea is used in Afonso et al. (2013), where the procedure to find the path relies on generalized Voronoi diagrams. Here, a number of points is considered: on the facets of the obstacles (modeled as convex bounded sets) and on the boundaries of the navigation environment in order to obtain a Voronoi graph. Subsequently, the nodes of the graph that are inside the prohibited region are removed along with the edges that connected these nodes to the other ones in the graph. What then remains is a graph whose edges form paths that present the nice property of being equally between the obstacles which are the closest

[11] In the sense of the economical MPC.

[12] Interpreted as XOR.

to an edge and the edges themselves. This property comes from the nature of the Voronoi graph construction, which maximizes the minimum distance to all points. The graph is augmented with the initial and final position, preserving its connectivity. After that, using a Delaunay triangulation the space is divided into triangles whose intersection is empty, except for their sides. This triangulation relies on the vertices of the obstacles and of the known region. The triangles that are crossed by the planned path are computed and merged to form convex polytopes, attempting to have the minimal number of convex regions along the tunnel. These polytopes are used further to impose the obstacle avoidance constraints as in the tunnel-MILP formulation (23).

In a similar approach, Deits & Tedrake (2015b) treats the problem of path planning in a multi-obstacle environment for a rotary-wing UAV. The underlying idea is based on a mixed-integer optimization which assigns polynomial trajectories to convex regions (known to be obstacle free). The paths are defined as piece-wise polynomial functions in time with vector-valued coefficients. Having an a priori (offline) chosen degree of polynomials and number of pieces, the optimization problem returns the coefficients of each polynomial which ensure a collision-free trajectory. This polynomial parametrization is possible because the model of the considered system[13] is characterized by differential flatness. Regarding the obstacle-free convex regions, they are obtained offline using IRIS (Iterative Regional Inflation by Semidefinite programming Deits & Tedrake (2015a)), a technique for greedy convex segmentation of the free space. The proposed approach is tested for various simulations scenarios with satisfactory results, although sometimes the convex segmentation does not cover the entire obstacle-free space. Moreover, a comparison with the classical[14] MIP method is done, the second method may involve a more complex integer program. An interesting aspect of this work is that the generated paths are dependent on the considered convex safe region segmenting the space.

Besides the above presented works, there exists others which treat, but only marginally, the problem of path planning. For example, in Janeček et al. (2017a) or in Janeček et al. (2017b) a toolbox is introduced and it includes various helpers to generate a reference path. Thus, a circular trajectory of a known radius or a polygonic reference passing through the given set of waypoints are provided, but the use of MIP is kept to a minimum.

3.3. Trajectory planning

The trajectory planning pertains to the problem of determining both the path and how to move along it. Thus, a trajectory planning strategy returns a path which is explicitly parametrized in time. In this section we consider first the works dealing with general problems and after that we focus on classical applications in the field of autonomous vehicles (i.e., autonomous overtaking, merging junctions or lane changing).

Trajectory planning refers to the construction of a function generated in open-loop, which associates a time to each point on the path and takes into account the particularities of the system dynamics.

Definition 3. *A trajectory between two points x_0, x_f from the navigation space is given by the continuous function $\gamma : [t_0, t_f] \to \mathcal{X}$ with*

$$\gamma(t_0) = x_0,\ \gamma(t_f) = x_f. \tag{24}$$

♦

As the path planning task, the specification of the continuous function describing the trajectory can be done in two different ways. For the second one, the set of all waypoints describing the trajectory is modified as follows: $\tilde{\mathcal{W}} = \{(x_{w_0}, t_{w_0}) \dots (x_{w_{N_w}}, t_{w_{N_w}})\}$ and an additional condition to the Def. 3 is considered: $\gamma(t_{w_k}) = x_{w_k},\ \forall x_{w_k}$. Nevertheless, the waypoint method is rarely used in MIP-based trajectory planning (and in the related literature). Hence, whenever it is not specified otherwise, the trajectory is given explicitly as the continuous function $\gamma(t)$.

Trajectory planning in various tasks

One of the classical works treating the problem of trajectory planning is Earl & D'Andrea (2005a) which introduces an iterative MILP algorithm. More precisely, considering circular obstacles (see Section 2.2, eq. (13)) and a traditional nonholonomic (car-like) vehicle, the authors propose an algorithm which guarantees obstacle avoidance over the entire trajectory and distributes avoidance times efficiently, resulting in smaller MILP formulations.

The proposed algorithm needs the discretization in time of the continuous nonholonomic model. An interesting aspect is that the non-uniform discretizations are exploited in order to bypass an extensive computational effort corresponding to the resolution of large MILPs. Support for non-uniform discretizations in time allows the use of intelligent time step selection algorithms for the generation of more efficient MILP formulations. Thus, the idea behind of the iterative MILP algorithm is to find a convenient distribution of the avoidance times, avoiding in this manner the growing of the specific MIP problem complexity (induced by the enumeration of the constraints at each sampling time). Therein, two different problems are considered: trajectory-generation with obstacle-avoidance requirements and minimum-time trajectory-generation problems.

Iterative MILP algorithms address the issues of MILP coping with large-scale models.

[13] The nano-quadcopter Crazyflie 2.0

[14] From the authors point of view: use the obstacle faces ("hyperplanes") directly in the generation of safe region

The work of Earl & D'Andrea (2002) studies the cooperative control of multi-vehicle systems. Building on the requirements of a robotic competition (but in a simplified version), the authors model these simplified competition rules using a hybrid system. Further, they put the control problem within an optimization framework, using MILP. The considered simplified competition involves two teams of robots, the attackers and the defenders, on a playing field with a region at its center called the defense zone. The attackers are drones directed toward the defense zone. The objective for the defenders is to block the attackers from entering the defense zone by intercepting each attacker before it enters the zone. Once an attacker enters the defense zone or is intercepted by a defender it remains stationary for the rest of the competition. While pursuing its objective defenders must avoid collisions with other defenders and obstacles as well as avoid entering the defense zone which is off limits to defending robots. The control strategy is implemented with a centralized controller with perfect knowledge of the system, perfect access to all states, and with the ability to transmit control signals to the defenders instantaneously. The controller needs to figure out the inputs to provide each defending robot so that the objective is achieved. Using MPC, they obtained a set of control inputs that minimize the number of attackers that enter the defense zone over the duration of the drill and, in addition, is consistent with the system dynamics (robot dynamics) and the constraints (no collisions, etc.). The obtained model is a system composed of continuous and discrete states (a hybrid system) with linear dynamics subject to inequality constraints and logical rules. In other words they obtained a MLD (mixed-logical dynamical) system (Bemporad & Morari, 1999). The resulting optimization problem is a MILP which can be readily solved using one of the state-of-the-art solvers (e.g., ILOG CPLEX).

Multi-robot systems can be modeled as hybrid (MLD) systems using MIP techniques.

Also, in Earl & D'Andrea (2005a) iterative MILP algorithms are presented, algorithms which address the issue of the MILP coping with large-scale models. The considered problem is trajectory-generation with obstacle-avoidance requirements and minimum-time trajectory-generation problems. These problems involve vehicles that are described by mixed logical dynamical Bemporad & Morari (1999) equations, a form of hybrid system. The algorithms use fewer binary variables than standard MILP methods, and require less computational effort. The iterative methods presented in that paper apply to MLD systems, which are governed by a mixture of logical rules (or state machines) and linear dynamical equations and are validated using a vehicle in an obstacle field, a particular MLD system.

In Earl & D'Andrea (2005b) the results in Earl & D'Andrea (2002), and Earl & D'Andrea (2005a) are gathered and extended. Hence, the multi-vehicle control problems is tackled using MILP and MLD systems Bemporad & Morari (1999). As in Earl & D'Andrea (2002), the methods are motivated on problems derived from an adversarial game between two teams of robots. The strategy for one team is fixed and modeled by state machines, while for the other the behavior is controlled using the iterative MILP method. On a side note, the approach therein was developed independently from a similar one introduced in Richards & How (2002). One of the interesting aspects regarding this paper is that the considered environment involves an adversarial component whose intelligent behavior is modeled using finite state machines and, implicitly, MLD systems.

As well, the trajectory planning topic is treated in Cetin et al. (2007). Therein, the addressed problem is the generation of collision-free trajectories for the reconfiguration of spacecraft formations, aiming also for an optimal fuel consumption. In order to model the spacecraft representation and their corresponding safety regions unrotated cubes (a particular polytopic representation) are used. Moreover, the trajectories to be followed are discretized in time using cubic splines and, thus, the generic problem is translated to an optimization problem. The resolution of this problem led to a trajectory parameterized by the spacecraft positions and velocities at a set of waypoints. To this end, the "big-M" technique is used to write the parametrized optimization as a MILP whose solution can be obtained either using standard MILP solvers (see Section 6) or using the concept of a sequential linear program. These two alternatives are compared therein, being applied on two standard validation tests whose aim is to swap the position within a spacecraft with minimum fuel consumption. The comparison leads to an ample discussion on the feasibility of MILP and on the methods necessary to shorten the resolution time.

Also, in Richards & How (2003) MILP has been used for open-loop vehicle trajectory design, enabling the inclusion of non-convex constraints such as plume impingement avoidance.

Trajectory planning in autonomous vehicles tasks

Classical applications in trajectory planning involve in most of the cases either merging junctions problem (e.g., Bali & Richards (2018); Huang & Peng (2017)) or autonomous overtaking (e.g., Ballesteros-Tolosana et al. (2017); Molinari et al. (2017)).

Recent advances in the field of autonomous vehicles bring to light the issues like vehicle lane change and overtaking on highways in the assisted driving framework. In order to perform such maneuvers, it is fundamental to compute suitable and comfortable trajectories that take into account the vehicle limitations as well as safety restrictions.

Safety restriction and vehicle limitations require efficient MIP formulations.

For instance, in Molinari et al. (2017) an efficient MIP formulation for the autonomous overtaking problem is in-

troduced. The considered vehicles have a simple dynamic model, the non-convex feasible region being represented using hyperplanes arrangement similar with Prodan et al. (2015)(as in Section 2). A complete formulation for trajectory generation with collision avoidance guarantees is presented for the case of an agent surrounded by a number of agents. The MPC criterion contains a desired reference state, which ensures that the overtaking takes place. They also considered two methods for reducing the numbers of binary variables: logarithmic formulation and cell merging (Stoican et al., 2013). The illustrative example therein and a comparative analysis between the two reduction methods from complexity point of view, are validated in IPG CarMaker (an accurate vehicle simulator).

In Ballesteros-Tolosana et al. (2017), the lane change and overtaking maneuvers aim to generate trajectories able to ensure the comfort of passengers. Mathematically, the problem is formulated as an Optimal Control Problem (OCP) due to its handling kinematic and collision avoidance constraints (the last ones in terms of hyperplane arrangements as in Section 2). In order to mitigate the drawback of the substantial computational effort associated with that kind of representation, a pre-analysis step is presented therein, a step consisting in the enumeration of the all possible overtaking configurations and of their resulting compact MIP formulation. The so-obtained nonlinear constrained optimal control problem is solved using a multiple-shooting approach which leads to improvements in the computational burden when compared to the pre-analysis step.

The presence of MIP in merging junctions problem can be found in a plethora of works. For instance, Fayazi et al. (2017) deal with the optimal scheduling of autonomous vehicle arrivals at intersection, eliminating the need of traffic signals. The idea is to design an intersection controller able to coordinate the flow of vehicles through intersection, scheduling the intersection access in safe conditions and receiving information from all subscribing vehicles. Moreover, each vehicle informs the controller about its movement and desired schedule. Therefore, the optimization-based controller should find the optimal sequence of vehicles crossing the intersection and their corresponding time of intersection access, minimizing the difference between the current time and the expected arrival time of the last vehicle passing the intersection and, also, the gap between assigned and desired access time of each vehicle. The resulting optimization problem has to take into account some physical constraints (speed limit and maximum acceleration) and a safety "window" between two consecutive vehicles accessing the intersection. These constraints lead to disjunctions and, respectively if-then statements, which are modeled through MIP (big-M formulation). In addition to this, the total optimization cost is composed of a min-max objective and a non-linear cost (an absolute value sign).

MIP may be employed to eliminate the non-linearity (Smith & Taskin, 2008).

Likewise, Huang & Peng (2017) tackled the problem of speed trajectory planning at signalized intersection. The idea is to optimize the vehicle speed trajectory over multiple intersections in order to minimize the fuel consumption and the travel time. In order to obtain a mathematical formulation of the problem , the authors assume that the signal traffic state is known and the influence of a lead vehicle is ignored. The main advantage of this formulation (with respect to, e.g., Fayazi et al. (2017)) is that they have considered the turnings at intersections (including a turning speed constraint). After a comprehensive description of the fuel consumption model, effect of turning and acceleration model, a MIP formulation is stated. The presence of integer variables models the crossing of the intersection without violating red light, indicating the active green phase window.

Also, Zhang et al. (2017b) presents a control strategy for a heterogeneous traffic network. Heterogeneity comes from the presence of both signalized and non-signalized[15] intersections. After the validation of the proposed model of the heterogeneous traffic network, a control strategy is stated using a MIP formulation. Finally, some comparisons between homogeneous and heterogeneous systems are provided in order to leave the door open for developing a systematic planning approach on deciding what traffic junctions require signal control to ensure a good traffic control performance. Moreover, Zhang et al. (2017a) proposes a traffic signal scheduling strategy which takes into account both vehicles and pedestrians presence. After developing a mathematical model for the pedestrians flow and for the vehicle traffic *network* (more than one intersection/junction), the problem of scheduling is stated as a MIQP (mixed-integer quadratic program).

3.4. Obstacle and collision avoidance

Obstacle and collision avoidance is the problem of finding input control signals which minimize a performance criterion as in (18a) while simultaneously avoiding collisions through a closed-loop strategy.

An adequate collision and obstacle avoidance strategy has became critical in order to ensure the safety and the integrality of both the system and the environment. In the last years, the control community has put the effort towards the obtaining of collision-free control strategies, which permit real-time operations. It is generally accepted that the collision avoidance problem is a challenging task due to the presence of the non-convex feasible domain. Describing this non-convex domain has computational and structural implications, generally leading to a trade-off between computational efficiency and control performance.

[15] A non-signalized intersection follows the first-come-first-serve (FCFC) principle.

Since MIP is able to explicitly model the non-convexity as in Section 2, it has become over time a proper approach for expressing collision avoidance problem.

The MIP approaches in the field of obstacle and collision avoidance problems can be distinguished in two main categories. The first class of MIP approaches is more often used, being a wide-spread approach in non-MIP (robot) motion planning literature. It consists in choosing a partitioning method which does not depend on the particularities of the environment (e.g the obstacles' form). A standard partitioning is the one formed by equally-sized square cells (Haghighi et al., 2016; Wang et al., 2015b). Using this approach, the collision-avoidance problem becomes, in most of the cases, a resource allocation problem, as presented in the Section 3.1. The second MIP approach or the geometric approach relies on set theory and on the capability of mixed-integer techniques to efficiently encode the description of non-convex sets (see Section 2). Even though different, both approaches share a common characteristic: the workspace is decomposed in a certain kind of cells and in some manner the MIP is employed in modeling disjunctive constraints.

Remark 12. *In the literature there are works which make a distinction between obstacle and collision avoidance. More precisely, the collision avoidance refers only to the mutual collision avoidance within multi-agent systems, and obstacle avoidance may refer to the collision avoidance either with stationary obstacles or with moving obstacles which have an a priori known trajectory, in most of the cases a periodic one. As well, some others references consider that obstacle avoidance refers to the static environment, while collision avoidance covers any kind of moving obstacles.* ♦

Static obstacles

The work of Culligan (2006) presents a path planner using MILP to solve a receding horizon optimization problem for unmanned aerial vehicles (UAVs). The MILP formulation has two important components: the hard constraints for obstacle and multi-vehicle avoidance and an approximation of vehicle dynamics. The 3D case is, as well, considered and comprehensively presented. Moreover, some enhancements of the MILP framework are discussed in order to provide decreased resolution time and also increase the capability of the path planner. These improvements consist of various techniques like a variable time step size, linear interpolation points, and horizon minimization. A noteworthy aspect, the concept of variable time steps is extended to the receding horizon, non-iterative MILP formulation. Variable time step sizing allows the simulation horizon time to be lengthened without increasing resolution time dramatically. Horizon minimization decreases resolution time by removing unnecessary obstacle constraints from the the problem (similar with Janeček et al. (2017a)).

In Schouwenaars (2006) the author presents a framework for safe online trajectory planning of unmanned vehicles through *partially unknown* environments. A MPC framework is employed, MILP playing an instrumental role in order to incorporate the collision avoidance constraints (similar with the techniques in Section 2). An interesting aspect is that the agents can be controlled either through a standard (velocity) control system, or by using a maneuver scheduler that allows the implementation of a maneuver from a discrete set of possible ones. This hybrid control architecture is applied and enhance for a particular type of dynamics, corresponding to a small-scale helicopter. As well, the problem of the feasibility of the proposed control strategy has been extensively treated. Therefore, a beneficial concept was considered: terminal feasible invariant set, a set in which an agent can remain for an indefinite period of time with anti-collision guarantees. Effectively, these sets are computed online, being represented as affine constraints on the last state of the horizon. Via these sets, it is provided an a priori known backup plan that is dynamically feasible and obstacle-free and, thus, the feasibility and safety can be guaranteed. The proposed strategy was tested on an unmanned Boeing aircraft using scalable loiter circles as feasible invariant sets. From the multi-agent perspective, the control strategy is a distributed one, each agent only computing its own trajectory while accounting for the latest planned behaviour of the agents from its proximity. The potential conflicts are solved in real-time such as to preserve the feasibility guarantees. In order to illustrate the benefits of the considered strategy, the algorithm was run over a scenario involving a fleet of small-scale helicopters which aimed at maintaining wireless connectivity in a cluttered environment.

Obstacle and collision avoidance constraints are usually imposed at the sampling time without regards to the intra-sample behavior of the agent. Hence, it is possible for an agent to "cut the corner" of an obstacle while apparently respecting the constraints. The idea employed in the literature is to consider additional constraints which ensure that the segment between two consecutive positions does not cut the obstacle Maia & Galvao (2009); Richards & Turnbull (2013). Stoican et al. (2018) provides a treatment of the multi-obstacle case within a hyperplane arrangement setting with exact and over-approximated representations.

Mobile obstacles

The mobile obstacles in the MIP framework are not so common as the static environments due to the associated computational burden. In most of the references treating the time-varying environments, the interdicted mobile regions are modeled as rectangular/cubical exclusion regions Richards & How (2005, 2002). In fact, this representation is a particular case of the polytopic representation (see Section 2), but with a limited number of corresponding mixed-integer linear constraints. Thus, the exclusion region constraints in a MIP formulation for two moving obstacles within a 2D environment (x-y coordinates) are

given as:

$$x_1 - x_2 \geq d - M\alpha_{d1} \tag{25a}$$
$$x_2 - x_1 \geq d - M\alpha_{d2} \tag{25b}$$
$$y_1 - y_2 \geq d - M\alpha_{d3} \tag{25c}$$
$$y_2 - y_1 \geq d - M\alpha_{d4} \tag{25d}$$
$$\sum_{i=1}^{4} \alpha_{di} \leq 3 \tag{25e}$$

where d is the safety distance (length of the edge of square exclusion region), α_{di} are binary variables, while (25e) ensures that at least one of the above constraints is active. It is worth to mention that the rectangular exclusion zone can be used in order to model other non-convex constraints, e.g. Culligan (2006); Richards & How (2005)

Like trajectory planning, the collision avoidance problem can be found/applied in tasks regarding the autonomous vehicles. For instance, in Mukai et al. (2017) the problem of merging for vehicles on a motor way is tackled using MIP along with a receding horizon strategy (MPC). Thus, interdicted regions are described using logical statements (AND/OR), which are further modeled as mixed -integer constraints, using the "big-M" method.

Furthermore, Molinari et al. (2017) treat the autonomous overtaking problem using MPC in an efficient MIP formulation. The considered vehicles have a simple dynamic model, the non-convex feasible region being represented as in Section 2. The MPC criterion contains a desired reference state, which ensures that the overtaking take place.

As well, Bali & Richards (2018) propose a method for vehicle merging scenarios in junctions with relative cost prioritization. The method is based on MPC, employing MIQP optimization. The scheme provides optimal control properties while maintaining safety and recursive feasibility. The latter properties are ensured through positive control invariance of simple time headway constraints. For examples with two vehicles, tunable prioritization and gap acceptance are verified and presented on a decision graph. Priorities are then demonstrated to be respected in an example with four vehicles.

Equivalent MLD formulation for obstacle avoidance

The mixed logical dynamical (MLD) systems were introduced for the first time in Bemporad & Morari (1999) and represent a relevant framework for modeling and controlling systems which incorporates linear dynamic equations, logic rules, and operating constraints. These are described by linear dynamic equations subject to linear inequalities involving real and integer variables as in (26).

$$x_{k+1} = Ax_k + Bu_k + B_2\delta_k + B_3 z_k \tag{26a}$$
$$y_k = Cx_k + Du_k + D_2\delta_k + D_3 z_k \tag{26b}$$
$$E_2 z_k + E_3\delta_k \leq E_1 u_k + E_4 x_k + E_5 \tag{26c}$$

where x_k, y_k, u_k denote state,output and, respectively, input vectors of the system. The $\delta_k \in \{0,1\}^{n_\delta}$ and $z_k \in \mathbb{R}^{n_z}$ are auxiliary logic and continuous variables, respectively.

Let us consider a system described by a single agent which has to move in a non-convex domain. The agent is described by the dynamics[16]:

$$x_{k+1} = Ax_k + Bu_k \tag{27a}$$
$$y_k = Cx_k + Du_k \tag{27b}$$

We examine a basic example, the agent has to stay outside a polytope $P = \{x : Sx < R\}$:

$$x_{k+1} \notin P.$$

We aim to describe the considered system via a MLD formalism, more precisely, to rewrite the above condition in the form of (26c).

Firstly, it is straightforward that (27a) and (26a) are equivalent if we consider the matrices B_2 and B_3 null. Next, writing the imposed condition ($x_{k+1} \notin P$) in a similar manner as in (7a) and replacing x_{k+1} according to the dynamic, we compactly[17] write:

$$\begin{bmatrix} -MI_N \\ \mathbf{1}^\top \end{bmatrix} \alpha_k \leq \begin{bmatrix} SA \\ \mathbf{0} \end{bmatrix} x_k + \begin{bmatrix} SB \\ \mathbf{0} \end{bmatrix} u_k + \begin{bmatrix} -R \\ N-1 \end{bmatrix}, \tag{28}$$

where M is a sufficiently large constant (according to the "big-M" formulation), N is the number of linear constraints describing the polytope $P = \{x \in \mathbb{R}^d \mid Sx \leq R, S \in \mathbb{R}^{N\times d}, R \in \mathbb{R}^N\}$, and $I_N \in \mathbb{R}^{N\times N}$ is the identity matrix. By adding constraints on input ($S_u u_k \leq R_u$), the MLD formulation of the considered system is:

$$x_{k+1} = Ax_k + Bu_k \tag{29a}$$
$$\begin{bmatrix} -MI_N \\ \mathbf{1}^\top \\ \mathbf{0} \end{bmatrix} \alpha_k \leq \begin{bmatrix} SA \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} x_k + \begin{bmatrix} SB \\ \mathbf{0} \\ -S_u \end{bmatrix} u_k + \begin{bmatrix} -R \\ N-1 \\ R_u \end{bmatrix} \tag{29b}$$

Remark 13. *The MLD system* (29) *is* well posed *according to Definition 1 from Bemporad & Morari (1999).* ♦

In Ritter et al. (2014) the equivalent MLD formulation is further extended. A multi-vehicle cooperating system is modelled and controlled such that the system reaches some target locations in a reasonable time avoiding any possible collision among the agents. Thus, MLD-system represents a valid compact formulation for modeling multi-vehicle system, because all existing constraints (dynamics, *collision avoidance*, measurement etc.) are embedded in a complete configuration, which allows to use existing methods to solve the control problem.

[16] For the clarity of the presentation, in what follows we will neglect the equations (26b) and (27b) but the reasoning can be readily adapted.

[17] "$\mathbf{0}$" stands for the null matrix of dimensions appropriate to the place into which it is inserted

MIP reformulation for drift counteraction optimal control

Another form in which MIP is employed in motion planning is within DCOC (drift counteraction optimal control) or optimal exit-time control Zidek et al. (2017a). The main objective in such problem is to satisfy prescribed constraints for as long as possible Zidek et al. (2017b). In other words, DCOC is a particular optimal control problem, aiming to determine a sequence of control inputs that maximizes the first exit-time from a given set. There exist various applications in motion planning where DCOC represents an useful tool, handling well the systems with finite resources (fuel or energy). In Zidek et al. (2017b) the MILP formulation of DCOC with application in spacecraft attitude control is fully addressed and for its resolution a LP-based iterative procedure is introduced.

As well, in a similar approach, Maia & Galvao (2009) present an implementation of the shifting prediction horizon for an MPC controller, tackling the obstacle avoidance problem, using binary variables and implicitly MIP. Moreover, Richards & How (2003) MILP optimization is used to effect a variable horizon length, leading to guaranteed finite-time completion.

4. MIP in multi-agent systems. Connectivity maintenance and formation control

In today's complex and various environments, the vast majority of the activities is too difficult and time-consuming to be handled by only one agent/robot/entity. Thus, in order to perform these activities with increased accuracy, redundancy and in a reduced time, cooperative teams of robots/agents may be employed. In this manner, the key elements of risk for the safety and integrity of systems are mitigated at the expense of an extensive augmentation of the systems to be supervised and controlled. Factors like the large scale, geographical distribution, high failure rates and heterogeneity of network systems are becoming decisive in consideration of the suitable control architecture.

An important issue in the control of multi-agent systems is the coordination of a cooperative team of agents (robots) in order to accomplish a given "mission" in an efficient manner. In several cases, these teams of agents are required to converge (and maintain) a specific spatial configuration. Therefore, formation control is often a prerequisite for other applications in motion planning.

Connectivity maintenance refers to providing constraints which guarantee a connected communication graph among the agents at all times.

More precisely, connectivity maintenance for a formation of agents refers, generally, to the ensemble of feedback laws, path/trajectory planning, collision avoidance and task assignment which guarantee that agents can communicate between themselves (to ensure data flows for, e.g., control decisions, data gathering, distributed estimation, etc.).

Remark 14. *Even though connectivity maintenance and formation control are not different labels for the same phenomenon, the underlying idea is similar, the difference being the global objective. A key difference between the two notions is the relevance of relative distance between agents: for connectivity maintenance the unobstructed communication is the key aspect, while the inter-distance or relative velocity are relevant only in their effect on the "line of sight" between agents (e.g., distance less than a communication range threshold).* ♦

Formation control aims to provide constraints guaranteeing desired relative distances and velocities among a group of agents.

There are two main approaches in the literature which tackle the formation control problems Qu (2009): *the leader-follower design* and *leaderless approach*. The former consists in designating one of the agents as a leader moving in a certain way with the remaining agents tracking the leader in order to maintain formation around it. The later involves the coordination of the agents through a global consensus for attaining the global objective. Choosing between these two approaches depends on the particularities of the team/group of agents, i.e., their capability of communication and sensing. Moreover, these aspects affect further the control architecture, as will be detailed in Section 5.

Corner cutting avoidance conditions

The corner cutting avoidance problem is an important but often overlooked part in motion planning. Obstacle and collision avoidance constraints are usually imposed at the sampling time without regards to the intra-sample behavior of the agent(s). Hence, it is possible for an agent to "cut the corner" of an obstacle while apparently respecting the constraints.

Corner cutting aims to provide constraints guaranteeing intra-sample collision avoidance.

The idea is conceptually simple: an agent avoids corner cutting iff its next position lies in its visibility region (the union of all rays spanned from the current position not intersecting any of the obstacles). Note that the dual notion (the under-shadow region – the region hidden from the viewpoint of an agent) may be used to enforce full coverage of the feasible space, e.g., the gallery problem.

Since describing exactly such regions is impractical, mixed-integer formulations (in either exact or over-approximated form) are used. Within the hyperplane arrangement framework presented earlier, these constraints employ (either implicitly or explicitly) three sign tuples: the current and future positions of the agent, $\sigma, \sigma^+ \in \Sigma_{X\setminus\mathbb{P}}$, and the coordinates of the obstacle(s), $\sigma^\bullet \in \Sigma_{\mathbb{P}}$.

We are aware of results from Afonso et al. (2016); Maia & Galvao (2009); Richards & Turnbull (2015) which discuss over-approximated corner cutting constraints and

Table 2
MIP for multi-agent systems.

Connectivity maintenance	Afonso et al. (2020); Earl & D'Andrea (2005b); Richards & Turnbull (2013); Schouwenaars et al. (2001a); Sun & Cassandras (2015)
Formation control	Bellingham et al. (2002b); Papen et al. (2017); Prodan et al. (2012); Ritter et al. (2014)
Corner cutting avoidance	Afonso et al. (2016); Maia & Galvao (2009); Richards & Turnbull (2015); Stoican et al. (2018)

provide constructive details: Maia & Galvao (2009) provides the initial construction; Richards & Turnbull (2015) and Afonso et al. (2016) improve it by reducing the number of necessary constraints and by reducing the number of necessary binary variables, respectively. Stoican et al. (2018) goes further by providing exact and over-approximated descriptions of the under-shadow (and of its complement, the visible) region generated by multiple agents within a multi-obstacle environment.

Specifically, in Maia & Galvao (2009), to avoid cutting a single obstacle, there has to exist at least one half-space containing the obstacle which does not contain the current and successor positions of the agent:

$$\exists i \text{ s.t. } \sigma(i) = \sigma^+(i) = 0. \tag{30}$$

Since such conditions often appear in MPC problems where both current and successor sign tuples are decision variables, (30) is rewritten such as to avoid nonlinearities:

$$\sum_i [1 - \sigma^\circ(i)]\, \sigma^+(i) < N + \sum_i [1 - 2\sigma^\circ(i)]\, \sigma(i), \tag{31}$$

for all possible values $\sigma^\circ \in \{0,1\}^N$. The idea is that from all constraints (31) at least one reduces to[18] (30), i.e., the one for which $\sigma^\circ = \sigma^+$.

Richards & Turnbull (2015) improves on Maia & Galvao (2009) by reducing the number of constraints (31) from 2^N to a more manageable N. This is done by forcing two consecutive positions x, x^+ to respect the same constraint, with our notation: $-h_i^\top x \leq -k_i + M\sigma^+(i)$ and $-h_i^\top x^+ \leq -k_i + M\sigma^+(i)$ for all $i = 1 \dots N$. This implies that there exists at least one index i s.t. both x and x^+ lie on the same side of the hyperplane (and thus on the opposite side from the obstacle), similar with (30). Afonso et al. (2016) proposes a logarithmic scheme to reduce the number of binary variables involved in the selection of the active hyperplanes.

[18] Both (30) and (31) assume a single obstacle ($\Sigma_{\mathbb{P}} = \{\sigma^{\bullet,1}\}$) where, by convention, $\sigma^{\bullet,1}(i) = 1, \forall i$. This implies that $\sum_i \sigma(i) \leq N-1$, $\sum_i \sigma^+(i) \leq N-1$.

The common shortcoming of these approaches is that they do not easily handle the multi-agent multi-obstacle case. Stoican et al. (2018) generalizes (30) with:

$$\sum_i |\sigma^{\bullet,j}(i) - \sigma(i)| \cdot |\sigma^{\bullet,j}(i) - \sigma^+(i)| > 0,\ \forall \sigma^{\bullet,j} \in \Sigma_{\mathbb{P}}. \tag{32}$$

As stated earlier, when both σ, σ^+ are variables, (32) becomes nonlinear due the bi-linear term appearing in it. Hence, a relaxation such as in (31) may be used:

$$\sum_i |\sigma^{\bullet,j}(i) - \sigma^\circ(i)| \cdot |\sigma^{\bullet,j}(i) - \sigma^+(i)| > -|\sigma^\circ - \sigma|, \forall \sigma^{\bullet,j} \in \Sigma_{\mathbb{P}}, \tag{33}$$

for all $\sigma^\circ \in \Sigma_{X \setminus \mathbb{P}}$, i.e., (33) reduces to (32) when $\sigma^\circ = \sigma^+$.

Connectivity maintenance conditions

The optimal dynamic formation control is a popular topic in the multi-agent systems field, and MIP-based methods provide a suitable tool for the modeling of the corresponding constraints. In what follows, we briefly present the topics where the connectivity is induced by the internal factors (as, e.g., communication).

The mixed-integer constraints enforce that each agent is within the communication range of at least one other agent. Moreover, pairs isolation need to be avoided, i.e., if an agent is enforced to stay in a communication range of another, the latter cannot be constrained to stay in the range of the former, but in the range of a third one. In order to formulate the corresponding mixed-integer constraints we proceed as in Afonso et al. (2020). Hence, we consider a binary variable $\xi_{i,j,k} \in \{0,1\}$ which is 1, if the i-th agent is inside of the communication range of j-th agent at the timestep k, and 0 otherwise. Next, we can write the following constraint, ensuring the communication connectivity among the N agents:

$$\sum_{i=1}^{N-1} \sum_{j=1}^{N} \xi_{i,j,k} = N-1, \quad 0 \leq k \leq T, \tag{34a}$$

$$\sum_{v_i, v_j} \xi_{i,j,k} \leq \#S - 1, \ \ 0 \leq k \leq T, \ \ \forall S \subset V, \tag{34b}$$

$$\xi_{i,j,k} = 0, \ \ 1 \leq i \leq N, \ \ 1 \leq j \leq i, \ \ 0 \leq k \leq T. \tag{34c}$$

The constraint (34b) prevents the possible pairs isolation, imposing for each sub-graph S of the original communication graph V that at least one node has to be able to communicate with a node outside the graph. Moreover, alongside with (34a) it guarantees that only connected communication graph can be obtained (for a detailed discussion, see Theorem 2 from Afonso et al. (2020)), i.e. the connectivity is maintained.

Furthermore, Sun & Cassandras (2015) is focusing on the *coverage control problem*. The agent team works in a 2D mission space, maximizing coverage in this space. The formation control is approached from a leader-follower setting perspective, the leader moves on a given trajectory while the remaining agents must maintain the formation. Thus, the formation becomes dynamic as soon as the leader starts moving along the given trajectory, and it must adapt to any environmental change (a new mission, a new composition of the team or some obstacles detected). Firstly, it is presented a brief formulation for a general optimal formation problem as an optimization problem, whose objective function depends on the agent spatial positions. The constraints of the optimization problem are given by a feasible space for positions (included in the 2D mission space) and by the condition that an undirected graph must be connected. This graph is modeling the desirable links among agents. The feasible space can be either convex (without obstacles) or non-convex (with obstacles). This aspect complicates the way in which the graph constraint should be treated. Hence, these two cases must be tackled separately. For a convex feasible space, the optimal dynamic formation problem is rewritten as a MINLP, by introducing a set of flow variables over the undirected graph (associating to every link between two agents an integer flow amount). The problematic constraint is transformed into a set of mixed-integer inequalities. Also, the computationally demanding re-solving of the MILP for all time instants is avoided, providing a sufficient condition for maintaining optimal formation in a certain time interval. For a non-convex feasible space, the problem is solved in a different approach. At each moment, when the connectivity is lost, a new undirected connected graph is constructed such that the effort to maintain the initial formation is minimal. An algorithm is developed to construct the new graph, and this is used as an input for CPA (Connectivity Preservation Algorithm). An illustrative example of this second situation is considered therein, where the initial formation is computed using the MILP from the convex feasible space case, obstacles being considered afterwards.

Connectivity induced by external constraints

This subsection introduces the types of objectives/ goals encountered in the formation control, goals enforced by the external[19] factors.

One of the classical examples where connectivity maintenance is externally induced is the problem of efficient coverage of a specific area by mobile sensors. For instance, Ritter et al. (2014) tackled this problem for a mobile sensors platform composed of a number of sensor-equipped autonomous vehicles. An adaptive observation strategy for on-line estimation of the state of a dispersion process is proposed, based on the model of the process and the multi-agent collaborative systems. The multi-vehicle cooperating system is modelled using MLD systems (Bemporad & Morari, 1999); and controlled such that the target location is reached in a reasonable time while simultaneously avoiding any possible collision among the agents. The target positions of the agents are received from the PDE[20]-based state estimation, more precisely the optimal measurement locations. Thus, the MLD-systems represent a good alternative for modeling multi-vehicle systems, because all existing constraints (dynamics, collision avoidance, measurement etc.) are embedded in a complete configuration, which allows to use standard methods to solve the control problem.

As well, a standard problem is given by the rendezvous maneuver, a group of agents (usually, a fleet of spacecraft) has to achieve a specific formation in a minimum time. Papen et al. (2017) treat this problem for a fixed-wing-UAV fleet in an environment containing both static and dynamic obstacles. The control strategy is implemented through a distributed architecture (specifically, MPC applied for each agent), an agent knowing only its own trajectory (defined by its current position and velocity). By introducing binary variables to model the obstacles and to eliminate some nonlinear constraints (like velocity and acceleration constraints), the MPC optimization problem is transformed into a MILP. Noteworthy, the wake vortex of turbulent air formed behind every UAV is modeled as a dynamic obstacle because within these regions it is arduous to maintain control over the UAVs. To solve the resulting MILP is necessary a considerable time, but with some relaxation of the constraints and fine-tuning the complexity is limited.

5. Control architectures

There are three well-established classes of control architectures, and they have been extensively studied in various application domains: centralized, distributed and decentralized. The last two methods require the local controllers to optimize over only their local inputs having similar computational burden. The difference between these two is given by the impact of communication, decentralized control requires no communication among the agents.

To identify/discuss the control architectures weaknesses and strengths with respect to the motion planning field is not the scope of this paper, thus, we only focus on how

[19] W.r.t the multi-agent system.

[20] Partial Differential Equation

Table 3
Classification of MIP approaches in motion planning

Agent(s)	Centralized	Decentralized	Distributed	References(e.g.)
$N = 3$ / $N \geq 3$,partially	✓	×	×	Chen et al. (2016a)
$1 \dots N$	✓	×	×	Earl & D'Andrea (2005b); Haghighi et al. (2016); Janeček et al. (2017a), Ritter et al. (2014); Sun & Cassandras (2015)
$1 \dots N$	✓	✓	×	Wang et al. (2015b)
$1 \dots N$	×	×	✓	Papen et al. (2017)
$N = 1$	✓	×	×	Molinari et al. (2017); Ragi & Mittelmann (2017)

Note: (✓) treated, (×) not treated, $(1 \dots N)$ any number of agents

they are employed along with MIP. In Table 3, we delineate a classification of MIP-based motion planning references w.r.t. the control architectures and number of agents involved in the formulations.

It is worth mentioning that the control strategies employed in other architectures than the centralized way are optimization-based, with predilection MPC. Thus, in this section an emphasis is put on the specific MPC implementations. Nevertheless, we have not neglected the references where MIP is used specifically in distributed/decentralized non-MPC strategies due to its capability to formulate task allocation problems. These are extensively treated in Section 3.1.

The control architectures involving MIP naturally evolve from centralized to decentralized and distributed strategies.

5.1. Centralized

Due to its theoretical simplicity, the centralized approach is the most used way of controlling a multi-agent system. In this architecture, the multi-agent system is treated as a whole, equating an extended single-agent system. The physical restrictions (e.g. communication limitations) are completely ignored, each agent having complete knowledge on the behavior/actions of the others, all information being available in the single global controller Rawlings & Mayne (2009). However, this methodology is limited, not only because of the undeniable physical constraints, but also due to numerical difficulties, induced by the substantial complexity of the extended system.

As stated in Section 3, in the literature, MIP may appear at different levels of control. For instance, in Chen et al. (2016a) the problem of collision avoidance is treated for a multi-agent system (minimum $N = 3$ agents). They consider N agents, each having a similar dynamic and $N-1$, so-called, danger zones: $\mathcal{Z}_{ij}$. Every agent makes use of two controllers: a "liveness controller" helps to complete the agent own objective (e.g. reach a target) and a "safety controller" has to keep away the agent from the *danger zones* of the other ones. The architecture of the "safety controller" is centralized and it guarantees, for $N = 3$, that every agent avoids entering into any other danger zones. Therein, MIP is used to provide the higher level control logic to synthesize the cooperative "safety controller". The objective is to group the agents in a pair such that the corresponding avoidance maneuvers may not lead to a dangerous configuration for the other agents.

For the generic obstacle and collision avoidance problem (in MI-MPC framework Section 3.4), in the corresponding OCP for the global system the dynamic behavior of the individual agent is coupled by means of the cost function and the constraints. Also, perfect knowledge of each agent dynamics (described by equations) is available to all the other agents. Consequently, the global model will be used in a predictive control context which permits the use of non-convex constraints for collision avoidance behavior.

A non-MPC example, Wang et al. (2015b) the problem studied in this paper is the collision avoidance in a multi-robot system. The approach is quasi-similar with one of previously-mentioned papers (Haghighi et al., 2016). The workspace is decomposed in equally-shaped cells and each cell cannot be visited by two robots at the same time. Each robot has to complete its own task by choosing from a set of possible trajectories. These trajectories are described by a sequence of adjacent cells and by the crossing time (in which an agent passes through respective cell). In this formulation the collision-avoidance problem becomes a resource allocation problem (the cells can be considered a shared resource). Assuming that the control of the agents is independent (i.e. they cannot pause their movement in order to give or to receive priority), the idea presented in this paper is to compute an initial delay time for each agent such that no collision to appear. In the centralized control strategy the trajectory is chosen by a central unit. Hence, the control algorithm has to return, besides the initial time delay, an "optimal" trajectory, the objective remaining unchanged. The associated optimization problem is MIP, binary variables modeling the disjunction, resulted from the corresponding resource allocation problem. Noteworthy, the objective is a min max one because it is

assumed that the robots work in parallel and the shortest (min) time for finishing the movement is the time of the slowest (max) robot. The objective can be reformulated as a standard minimization through the tools from, e.g., Smith & Taskin (2008).

While the complexity of MIP formulations increase in worst case situations exponentially w.r.t the number of binary variables, the reliability of MIP for real-time implementation decreases in the case of higher dimensions of the global systems. The same drawback is valid for the increase of the number of agents regardless of the formulation's efficiency.

5.2. Distributed

The underlying idea of distributed control approaches Maestre et al. (2014) is to divide the global control problem into a certain number of sub-problems, each involving a specific collection of local controllers or agents. Hence, each agent does not have access at the global information, but can be partially informed on the behavior of the other components of the local subsystem.

In large-scale multi-agent systems, where the agents are dispersed within a given workspace, it is more convenient to handle a set of smaller and/or simpler problems than treating the complex global system. The overall control strategy is given by the behavior of the local controllers, which may have a cooperative interaction. There are several advantages compared to centralized architecture: e.g., decrease of complexity and scalability. However, a loss of performance and of global stability may become apparent, being hard to be ensured compared to the centralized approach. Decoupling into independent nodes, design of robust control strategies, consensus seeking, all of these try to solve the problem, but still, with limited success Cao et al. (2012).

Within the MIP framework, most of the distributed control approaches using MPC strategy have been proposed. The features of MPC allow to handle explicitly the interactions between the different subsystems/agents. For example, in Schouwenaars (2006), a distributed MPC strategy is employed for navigation of a fleet of vehicles through a partially unknown cluttered environment.

Due to the inherent problem of complexity, and implicitly the lack of scalability, MIP was not a popular method for distributed architectures. However, the control community has given a particular attention to the distributed MIP resolution algorithms. For instance, Testa et al. (2017) propose an algorithm for resolution of a MILP where the constraints are *distributed* among the agents. Likewise, Vujanic et al. (2016) provides a decomposition method particularly useful for large scale MILPs, based on Lagrangian duality. Moreover, there are some works, which formulate the problem using MIP techniques, and for its resolution an heuristic method is employed, e.g., Van Parys & Pipeleers (2016) use Alternating Direction Method of Multipliers (ADMM).

5.3. Decentralized

During the last decades, decentralized control of multi-agent systems has gained a significant amount of attention due to the great variety of its applications, including multi-robot systems, transportation, multi-point surveillance and biological systems. An important topic of research is multi-agent navigation in both the robotics and the control communities, due to the need for autonomous control of multiple robotic agents in the same workspace. Important applications of multi-agent navigation arise also in the fields of air-traffic management and in autonomous driving by guaranteeing collision avoidance with other vehicles and obstacles.

As seen in the previous subsection, a main motivation for the modification of the control architecture is the computational burden of the centralized problem. A first alteration was to decompose ("distribute") the problem and solve the resulting sub-problems in expectation of achieving a consensus. Proceeding further and eliminating the aim of reaching consensus leads to a decentralized strategy ("everybody for itself"). Under this strategy, each agent owns a controller, which acts without taking into account any information about the behaviour of the other agents. Moreover, the exchange of information is limited (in most of the cases, reduced at a minimum).

A basic illustration for the decentralized approach is the merging junctions[21] problem Bali & Richards (2018). In a centralized (or in a well constructed distributed) approach, whenever two (or more) agents arrive at an intersection, there is a prioritization w.r.t a well defined criterion. For the decentralized approach, with limited communication and with no guarantee of optimality, in the most probable scenario the system reaches a deadlock. In other words, the decentralized approach cannot provide, usually, the theoretical guarantees characterizing the centralized and distributed approaches. However, in practice, the decentralized methods may lead to efficient strategies, avoiding the computational demand of the others methods; but an understanding of the sensitive aspects is absolutely necessary.

For instance, recalling the previously-mentioned (Subsection 5.1) example from Wang et al. (2015b) – the problem of collision avoidance in a multi-robot system. Using a decentralized control strategy, they assume that each agent chooses its trajectory independently (from an a priori known set), without informing the rest of the systems. In this case the collision avoidance strategy should consider all possible trajectories combinations and has to return the initial time delays such that the movement to be finished in the shortest time without collisions. The resulting optimization problem is MIP, as in the centralized case.

[21] Unsignalized intersections.

6. Control applications using MIP

While previous sections deal with modeling and control issues, this section covers some critical details for both the computer simulations and the hardware implementation of the solutions and MIP approaches presented throughout this manuscript.

It is worth mentioning that there does not exist some clear and uncontested guidelines capable of generating the most efficient MIP formulation for a given problem. The performance[22] of a formulation is customarily strongly dependent on the specific software tools or hardware platform. In the sequel, these aspects are briefly documented, the emphasis being placed on differences between the existing alternative implementations and their influence over the practical performance in motion planning.

6.1. Software for MIP

As was stated above, MIP is a powerful tool for planning and control problems due to its modeling capability and, additionally, the availability of specialized solvers. In the last decade, a consistent effort was put on developing MIP-specialized solvers in order to mitigate the numerical issues generated by the presence of integer/binary variables.

The optimization modeling languages are toolboxes which convert the mathematical formulation in a solvable form for the solvers.

Prior to proceeding further, we should emphasize the difference between optimization modeling languages and solvers. The former designates a toolbox/package/library which interacts with the latter: the mathematical model of the constrained optimization problem is put into an internal form which is then solved, and whose subsequent result is retrieved and displayed. Note that most modern tools do pre-processing steps (which may reduce the number of binary variables) or may even, automatically, put the problem in a MI form (e.g., in YALMIP when specifying a complementarity condition or a bilevel program).

Diverse programming languages and online resources facilitate the specification and resolution of MIP problems. In addition to classical Matlab, recently the attention of the optimization-based control community is moving towards other advanced programming languages as Python or Julia, which have become more accessible to the broader scientific and engineering community. In Table 4 we summarize these programming languages, the modeling languages and the solvers which may be joined in order to solve MIP.

Without being exhaustive, there are some popular optimization modeling tools: YALMIP (Lofberg, 2004), MPT (Herceg et al., 2013), AMPL (Fourer et al., 1987), CVXPY (Diamond & Boyd, 2016), PYOMO (Hart et al., 2017) or JuMP (Dunning et al., 2017). All[23] of them are open-source modeling languages which allows users to express a wide range of optimization problems (not exclusively, MIP) in a high-level (almost algebraic or pseudo-code) syntax. As depicted in Table 4, each modeling language is developed taking into account the specific features of a programming language. Note that an hierarchy among these modeling tools is strongly dependent on the experience of the user/researcher, the preference for one of them has a negligible impact regarding the resolution performances (e.g., computational burden). These performances are influenced rather by the choice of the solver w.r.t. the formulation.

Regarding the solvers, there exists a broad variety of options, we mention here that the most used in the field of MIP-based motion planning are CPLEX (2009), Inc. (2014) or Mosek (2015). For instance, Huang & Peng (2017); Zhang et al. (2017b) or Mukai et al. (2017), use GUROBI, while Richards & How (2002); Schouwenaars et al. (2001b) or Earl & D'Andrea (2005a) utilize CPLEX.

Remark 15. *The majority of the solvers (mentioned in Table 4) is able to deal with quadratic objectives and/or constraints, elements which are influential in many control strategies and/or applications, see, e.g.,* (18). ♦

While a review of how the resolution techniques employed by the solvers is well beyond the scope of this paper, we mention only the core techniques: branch-and-bound and cutting-plane algorithms. There exists a multitude of variations, each with its strengths and weaknesses. For instance, branch-and-cut method (Stubbs & Mehrotra, 1999) gathers the advantages of branch-and-bound and cutting-plane algorithms, decreasing the number of nodes to be explored in the search tree by iteratively introducing constraints to cut the feasible region.

In general, the solvers may be classified using different criteria, e.g. convex/non-convex, heuristic/deterministic. There are in the literature more detailed surveys, e.g. Belotti et al. (2013), treating this topic, but what is relevant from the current paper's perspective is the following aspect. Some of the currently available and reliable solvers may employ heuristics in order to accelerate the standard algorithms. This is a necessary requirement especially for complex (large) problem formulations and real-time resolution. As a word of caution, the performances (particularly, computational times) may vary considerably from a solver to another due to the use of heuristics. Hence, the concept of "the best MIP solver" is pointless, in our opinion. On a more positive note, we have observed that for any given problem there can be found a solver, capable of handling it.

Besides these powerful commercial solvers, there exists a variety of non-commercial/open-source solvers able to provide reasonable performances, in some cases better

[22] Computational time, feasibility etc.

[23] Except AMPL.

Table 4
Software for MIP implementation-examples

Software		References(e.g.)
Programming Language	MATLAB	Bemporad & Mignone (2000); Culligan (2006); Janeček et al. (2017a)
	Python	Welder et al. (2018)
	Julia	Lubin et al. (2018)
Modelling Language	Yalmip	Mukai et al. (2017); Zhang et al. (2017b)
	GAMS	Lee & Grossmann (2000)
	CVX/CVXPY	Dueri et al. (2017)
	Pyomo	Legg et al. (2012)
	JuMP	Welder et al. (2018)
	AMPL	Richards & How (2005, 2002)
Solvers	CPLEX	Janeček et al. (2017a); Stoican et al. (2013)
	GUROBI	Janeček et al. (2017a)
	SCIP	Berthold et al. (2012)

than commercial ones. An important characteristic of this kind of solvers is that their adjustment and adaptation to the challenges encountered in the real-world applications and to the real-time control requirements can be done in a faster and more straightforward manner than commercial ones which, in addition, have to take into account commercial considerations, balance between free/payed functionalities, etc.. For instance, SCIP (`https://scip.zib.de`) (Achterberg, 2009) was at the beginning a MILP solver which implemented the branch-and-bound algorithm with various heuristics, while the later versions were able to solve MINLPs with quadratic objective, non-convex MINLPs or MISDPs (mixed integer semidefinite programs).

Moreover, a part of the control community has concentrated its attention on techniques to adapt the standard MIP resolution algorithms. For instance, Bemporad et al. (1999) propose an efficient branch-and-bound algorithm, enhancing the tree exploring strategy. The application therein involves the control and state-estimation for a MLD-system (Bemporad & Morari, 1999). Similarly, Bemporad (2015) provides an algorithm which combines the classical branch-and-bound with nonnegative-least-squares (NNLS) methods, in order to solve MIQP problems generated by the hybrid MPC applications. The idea is further developed in Naik & Bemporad (2017) where the NNLS is replaced by accelerated-dual-gradient projection algorithm.

There are also some other works which exploit the particularities of the problem structure within branch-and-bound algorithm. For example, Feng et al. (2017) propose a variation of the branch-and-bound algorithm, branch-and-lift algorithm, which has better performances in resolution of a classical obstacle avoidance problem. As well, Hespanhol et al. (2019) provide an iterative variation branch-and-bound algorithm which exploits the block-sparse optimal control structure of the problem and also the information at previous time steps.

As a side remark, besides the standard MIP resolution algorithms, there exist in the literature some heuristic techniques. Without being exhaustive, we mention only two[24] of these:

i) ADMM (alternating direction method for multipliers)-based methods Kanno & Kitayama (2018); Takapoui et al. (2020). Although an algorithm for solving convex optimization problems, ADMM turns out to be an effective method to approximately solve some nonconvex problems as well. The idea behind the heuristic method is to use multiple restarts of ADMM with random initial points, in most of cases this provides an acceptable solution with small computational cost. This technique is frequently used in optimal (power) flow problems, e.g. traffic signal control Timotheou et al. (2014) or network Abboud et al. (2015); Scott & Thiébaux (2014). However, Van Parys & Pipeleers (2016) employ ADMM in motion planning for a multi-vehicle systems. As well, for this technique there exist a multitude of variations and adaptations. For instance, Stellato et al. (2018) develop a novel branch-and-bound algorithm, tailored for a ADMM-based solver.

ii) FP (Feasibility Pump) Fischetti et al. (2005). A heuristic method for finding a feasible solution of a given MIP, FP aims to minimize the difference between the solution of the LP-relaxation and the one of the original MIP. For instance, Miertoiu & Dumitrescu (2019) use and adapt the algorithm for sparse representation.

Remark 16. *Apart of solvers and modeling languages, there exist in the literature works as Janeček et al. (2017a) which provide a toolbox for MPC-based control for obstacle*

[24] The most encountered.

Table 5
Alternative and heuristic techniques for MIP resolution

Alternative	References
Branch-and-bound variation	Bemporad (2015); Bemporad & Mignone (2000); Bemporad et al. (1999); Feng et al. (2017); Hespanhol et al. (2019); Naik & Bemporad (2017)
ADMM	Abboud et al. (2015); Kanno & Kitayama (2018); Scott & Thiébaux (2014); Takapoui et al. (2020); Timotheou et al. (2014)
FP	Fischetti et al. (2005); Fischetti & Salvagnin (2009); Miertoiu & Dumitrescu (2019)

avoidance problem. The toolbox is developed in an object-oriented manner, allowing to readily set up the associated MPC problem and, afterwards, providing an efficient formulation of the underlying optimization problem for any inexperienced user. In other words, the OPTIPLAN toolbox encapsulates all aspects related to formulation and resolution of the MIP problem. ♦

6.2. Hardware platforms

Before detailing, it is worth mentioning that some of the existing works takes into account the particularities of the hardware platforms developing specific methods, whereas most of the remaining presents generic methods able (or not) to adapt to the constructive constraints.

There exists a broad variety of robotic platforms which are used in academic and/or commercial applications: aerial, surface or underwater vehicles. Frequently, these robots are involved in activities which are unsafe or troublesome for humans. Even though different levels of autonomy are possible, we can easily remark that the control community has considered with predilection the unmanned vehicles. The main justification comes from the elimination/mitigation of human risk. This aspect has beneficial consequences on cost effectiveness and, in most of the cases, on accuracy of the operations.

In Table 6, we depicted the classes of unmanned vehicles which have been used as hardware platforms for MIP-based motion planning problems: UAV (unmanned aerial vehicle), USV (unmanned surface vehicle) or UGV (unmanned ground vehicle), UUV (unmanned underwater vehicle). The specific characteristics of each class (or, sometimes, sub-class) of vehicles lead to various challenges in the design of the motion planning strategy. For instance, the USVs/UGVs are moving in a bidimensional workspace, whereas the UAVs and UUV in a 3D workspace thus leading to a higher complexity of the control problem. Another classification is given by the ability to stop and/or go backwards, for example, fixed-wing UAV need to maintain a minimum velocity (to avoid stalling), but the quadcopters/helicopters (rotary wing UAVs), having more degrees of freedom, can maintain an arbitrary velocity (up to being stationary in midair, i.e. hovering).

For instance, in Culligan (2006) an experimental validation of the MILP framework is done: test flights on an indoor quadrotor testbed demonstrate the reliability of the approach for the optimal path planner. For example, using the MILP path planner to create a plan ten seconds into the future, the quadrotor can navigate through an obstacle-rich field with MILP solve times under one second. Simple plans in obstacle-spare environments are solved in less than 50ms. A multi-vehicle test is also used to show non-communicating deconfliction trajectory planning using MILP.

Many applications in precision agriculture, disaster management and target tracking assume a collaboration between an UAV and ground-based sensors. The UAV serves as a mobile sink: it prolongs the sensors' lifetime (by canceling their need to communicate with a base station Xu et al. (2018)) and reduces operational costs (by canceling the need of direct human supervision Jawad et al. (2017)). Such applications impose energy-based limitations in the motion planning procedure, either induced by path length Khan & Kumar (2016) or by communication requirements Xu et al. (2018). Furthermore, many works simplify the motion planning by assuming predefined path primitives (e.g., the UAV is constrained to move in straight, parallel lines Wang et al. (2015a) or in spirals Yue & Jiang (2018)). Not in the least, when the environment is cluttered or uneven, the communication links may be weakened or lost as a result of signal attenuation Jawad et al. (2017). Thus, bounds on the communication time at a waypoint have to be considered, which are difficult to handle by fixed-wing UAVs. As stated elsewhere, the result is a nonlinear (in cost and constraints) constrained optimization problem which is often impractical to solve. Even a relatively simple requirement as ensuring hovering at the waypoint leads to a MINLP formulation (Mathur et al., 2016).

7. Alternatives to MIP

This section briefly presents the existing alternatives to MIP which are extensively used in motion planning problems. We delineate in Table 7 the state-of-the-art references for each of these alternatives.

Table 6
Classification of the hardware platform

Experimental platform	References
UAVs	Chen et al. (2016a),Bellingham et al. (2002a,b); Culligan (2006); Papen et al. (2017); Ragi & Mittelmann (2017); Rey & Hijazi (2017); Richards & How (2002); Schouwenaars (2006)
UGVs	Bali & Richards (2018); Ballesteros-Tolosana et al. (2017); Fayazi et al. (2017); Molinari et al. (2017)

7.1. Alternative formulations

Alternative formulations for MIP range from graph-based approaches, potential field formulations to the corresponding optimization problems relaxations.

Graph-based approaches

In contrast to the MIP approaches where the discrete decisions are encoded in a mathematical formalism and are solved as such, the graph-based approaches reduce these discrete decision to the search of the shortest path between nodes in a graph. Although these techniques can be applied for any MIP-based motion planning problems (see Section 3), in this subsection we concentrate on collision-free path/trajectory planning.

Remark 17. *For finding of the shortest path in a graph there exists in the literature various algorithms (e.g., Latombe (2012); LaValle (2006)). The most influential ones are Dijkstra's , greedy or $A^\star$ search algorithms.* ♦

The graph-based approaches in path planning are classified by graph construction. A first category builds the graph based on the cell decomposition methods, considering an explicit representation of the (multi-obstacle) environment. Thus, the similarity with the MIP approach is undeniable, polytopic sets being a popular decomposition primitive. For instance, polytopic sets were already used in Lozano-Pérez & Wesley (1979). More precisely, the graph nodes are defined by the vertices of the polytopic obstacles, and by the initial and the final point of the path. This graph is named the *visibility graph* because the link between the nodes is given by a straight line which does not intersect any obstacles. In other words two linked nodes can "see" each other. Thus, the collision-free path is the shortest path through this graph between the initial and final position of the agent. This approach performs well from a theoretical perspective if the considered agent is a point. For the general case when the dimensions are not negligible, the above method is extended by considering an artificial "growing" of the obstacles (commonly, via Minkowski sum). The main drawback of the above presented method is the computational burden, specifically in a complex environment (large number of obstacles with complicated forms) where the number of vertices is exhaustive.

The second category tries to mitigate this issue avoiding an explicit representation of the environment. This type of approach is also referred in the literature as *sampled-based algorithms* (see LaValle (2006)) for a complete literature review on this topic). Hence, the explicit representation is eliminated and the focus is put on a collision checking module. This module checks the feasibility of the trajectories connecting random sampled points from the obstacle-free space. The graph is constructed such as the link between two nodes represents a feasible path in the (multi-obstacle) environment.

Remark 18. *An interesting notion for sampled-based algorithms is the* probabilistic completness *(Barraquand et al., 1997; Ladd & Kavraki, 2004). That is, the probability that the algorithm returns a feasible solution tends to one if the number of sample points is sufficiently large ($\rightarrow \infty$). This is empirically demonstrated in Hsu et al. (2007).* ♦

In the literature, there are two important sampled-based algorithms: PRM (Probabilistic RoadMaps) (Hsu et al., 2007) and RRT (Rapidly-exploring Random Tree) (Weiss et al., 2017). The difference between them is given by the method of constructing the graph. The former (PRM) is a multiple-query method in the sense that after the construction of the roadmap (a rich set of feasible paths) it answers queries by computing an optimal path through the graph. Henceforth, the PRM is an useful method if an awareness map of the environment is available (Karaman & Frazzoli, 2011). There are a plethora of variants for the PRM, each representing a valuable improvement Karaman & Frazzoli (2011); Ladd & Kavraki (2004).

For the case when the environment is not a priori known, the RRT method is more suitable. In this approach the construction of the graph is incremental, the algorithm stops when a large enough set of collision-free paths is attained. Thus, a collision-free sample is added as a node in the graph and is linked with the surrounding nodes. The obtained graph is actually a tree. As PRM, RRT has a variety of versions. Some take into account the equations of motion and generate attainable paths LaValle & Kuffner (2001), others generate only geometric paths which became reference trajectories for a lower level controller. Moreover, some versions are tailored for complex

and/or unstable dynamics Leonard et al. (2008) or for uncertain dynamics Weiss et al. (2017).

Besides the classical graph-based approach, there exists in the literature a multitude of methods which combine the standard graph algorithms with advanced control strategies. For instance, Berntorp et al. (2018) presents a method for real-time integrated motion planning which uses feedback control, positive invariant sets, and equilibrium trajectories of the closed-loop system. In order to generate the collision-free trajectories the method employs, in an offline manner, a graph search over reference paths, each being associated with a constraint admissible positive invariant set. Next, they use pre-designed unconstrained linear quadratic controllers to track the reference paths.

As well, Altché & Fortelle (2017) addresses the problem of trajectory planning using an approach which consists in partitioning the feasible ("collision-free") region, while allowing to decompose the NP-hard problem as a path-finding problem in a well-designed graph followed by a (simple) optimization phase ("in MPC fashion") for a quadratic convex cost function. Also, Franzè & Lucia (2015) deals with the problem of obstacle avoidance in an unknown environment (considered agents are UGVs - autonomous ground vehicles). The proposed approach consists of two parts: an offline part which computes the ellipsoidal approximation of the one-step controllable sets for all possible scenarios (these approximations guarantee the existence of a feasible path through multi-obstacles environment) and a online part which involves the development of a MPC-based strategy in order to keep the agent in that sequence of ellipsoidal sets.

Potential field formulation

While MIP-based methods take explicitly into account the constraints and lead to a constrained optimization problem, the potential field-based formulations Chen et al. (2016b) relax the constraints by adding penalty terms in the cost. Essentially, the potential field approach relies on construction of a scalar function (so-called, the potential). This function takes high values when the agent stays within the interdicted zones. In the collision-free workspace the function is decreasing towards the goal configuration (i.e., the potential associated to the destination point is minimal). Thus, the agent may attain the final point moving in the direction of the negative gradient of the potential. Rimon & Koditschek (1992) provides an historical (and more detailed) review on the potential field formulation and how this approach is involved in motion planning. An interesting characteristics is that the potential field formulation is frequently used in decentralized or distributed control strategies Filotheou et al. (2018).

7.2. Optimization problem relaxation

As was stated above, one of the most significant capabilities of MIP is to handle non-convex constraints in non-convex optimal control problems. A natural way to tackle this type of problems is by extending the methods and techniques used for the convex optimal control. Often, the MI formulations are solved heuristically (e.g., Quaritsch et al. (2010) applies genetic algorithms) or relaxed by iterative solving (the optimization problem is broken into "reasonable" sub-problems which are solved iteratively). As an example, Xu et al. (2018) employs binary variables to model the link between an UAV and a ground sensor but relaxes the formulation through time allocation tactics and channel communication pre-scheduling. Thus, in the literature, a considerable effort was put in finding a technique which allows the translation/relaxation from non-convex to convex formulations without any major gap. This is referred as convex relaxation or convexification of the non-convex optimization problem. In the literature there are various works providing methods for convexification under several labels: e.g., convex relaxation, succesive convexification or time-varying constraints.

As its name suggests, the basic idea of successive convexification is to solve a non-convex optimal control problem via a sequence of convex sub-problems. The non-convexity comes either from having non-linear dynamics (Mao et al., 2016) and/or from non-convex state (and/or control) constraints (Mao et al., 2017). In both cases the same technique is applied: linearization, commonly, using a first order Taylor approximation (in a successive manner). Therefore, a preliminary condition on the functions which generate the non-convexity is necessary: they have to be differentiable. In an iterative manner, the linearization is done about the solution obtained at the previous step.

Although the linearization procedure leads to convex formulations, it also introduces two new issues, namely *artificial infeasibility* [25] and *approximation error*. These two drawbacks are addressed in the literature, a variety of algorithms was developed and, recently, a convergence analysis was elaborated Liu & Lu (2014).

For instance, Dueri et al. (2017) deals with the problem of the trajectory optimization for autonomous vehicles in an environment containing cylindrical and ellipsoidal obstacles. The approach employs the *Successive Convexification* technique which is used to solve the non-convex optimal control problem via a convergent sequence of convex optimization problems. It considers a discrete-time, finite horizon constrained optimization problem with a number of non-convex state constraints. Several assumptions are necessary in order to employ the technique at hand. The first assumption can be readily satisfied and involves that the obstacles boundaries should not be in contact with the ones of any other constraint. A second problematical assumption builds on a dynamic with a finite number of stationary points. Having the general formulation for the

[25] A solution of the original non-convex problem can become infeasible for the sequence of the convex sub-problems.

non-convex problem, the successive convexification technique is applied in order to obtain the sequence of convex sub-problems, each being linearized in an iterative procedure. This linearization results in a convex problem but, at the same time, introduces two drawbacks: approximation error and artificial infeasibility. To mitigate these two issues of the convexification, the authors introduce trust regions and penalty functions, respectively. The drawbacks of the convexification procedure and the ways to alleviate them are treated in-depth in the references such as Harris & Açıkmeşe (2014).

An interesting alternative for handling the non-convexity of the collision avoidance problem is based on time-varying constraints. The idea is mentioned in Frasch et al. (2013) and is used in, e.g., Janeček et al. (2017a) and Yu et al. (2014). In few words, the non-convex domain is decomposed in a sequence of convex regions and switching instants are introduced. At each moment the agent should stay in one of the convex regions. For instance, in Janeček et al. (2017a) the switching instants are steps of the prediction horizon in the MPC controller. Moreover, this approach is coupled with a heuristic black box which establishes the get-around direction and the sequence of convex sub-domains. This method can be seen as a particularization of the successive convexification. The difference comes from the way of obtaining the sequence of convex sub-problems/sub-domains. Time-varying constraints approaches consider an a priori known number of sub-domains, since the successive convexification is an iterative method, the sequence is growing until a feasible solution is obtained.

In Rey & Hijazi (2017) a different approach for convex relaxation is presented. For instance, therein the authors treat the aircraft conflict problem[26]. Basically, the provided formulation is based on complex numbers representation and it results in a tight convex relaxation for the inherently non-convex optimization problem. It is worth mentioning that the above reference includes a comprehensive literature review on the formulation of the air conflict problem as an optimal control problem using mixed-integer techniques (as MILP or as MINLP)[27]. Coming to the application, the aircraft separation condition is stated: the relative position of two aircraft should be greater than a certain threshold. As it was expected, this condition leads to a non-convex feasible domain which is modeled using a binary variable (actually, the feasible region is composed of two convex ones and the solution should be in one of these two convex subsets). The control actions (speed variation rate and heading deviation angle) admit a natural representation in the complex number form. Even so, the non-convexity is not eliminated (the disjunction constraint is maintained) but the considered formulation is useful towards the convex relaxation approach. This approach is extensively treated in Coffrin et al. (2017), where the non-convex constraint is tackled by deriving the corresponding convex hull, the problem being transformed in a MIQCP (mixed-integer quadratically constrained program). A further relaxation is possible by entirely omitting the non-convexity. An algorithm which included the relaxation is presented and tested (with excellent results) on two classical benchmark problems for conflict resolution.

As well, Huang & Peng (2017) solve the MI optimization problem using sequential convex optimization method (i.e., search the local optimum by forming convex sub-problems), avoiding the eventual curse of dimensionality. Likewise, Papen et al. (2017) solve a MILP by using a relaxation of the constraints and fine-tuning the complexity in order to limit the computation time w.r.t. a classical MILP resolution.

8. Conclusions and future challanges

In the preceding material we provided our evaluation on the state-of-the-art for MIP-based motion planning and also we aim to identify active topics and open problems in this field. It is important to mention that the valuable insights in the description of a non-convex feasible region represent an useful modeling tool not only for the motion planning but also for broader control fields.

As mentioned before, although the history of MIP starts almost 60 years ago, the interest of the control and robotic community on this topic is relatively recent and the research in this area is quite active. It is obvious that there exists a substantial progress in all of the topics mentioned in Section 3 and Section 4. However, there are many points that can be further enhanced since the developing of new and performing methods to provide exact solution for MIP is exponentially growing.

As open and active problems regarding mainly obstacle and collision avoidance topic we can mention the trade off between conservativeness and complexity in non-convex region modeling and representation. Even though there exist several valuable improvements on classical MIP formulations, the complexity remains an arduous issue imposing restrictions: only a small size problem can be solved in a real-time manner. The improvements may be accomplished by exploiting the underlying combinatorial structure of the MIP formulation. Nevertheless, whenever the problems are inherently non-convex and/or they involve alternative choices, then mixed-integer representations provide an useful and powerful tool, but we have to proceed with caution evaluating the structural properties that may lead to compact formulations. As well, important progress may be expected in the direction of the MIP resolution algorithms, which can be adapted and improved in order to attenuate the inherent complexity of the mixed-integer approaches by capitalizing on the different particularities

[26] According to the air traffic rules, the aircraft have to be separated by at least 5 nautical miles horizontally and 1000 ft vertically, otherwise they are in conflict.

[27] A more detailed review on the air conflict detection and resolution can be found in Kuchar & Yang (2000).

Table 7
Classification of the alternatives of MIP in motion planning

Method/Technique	References
Graph-based algorithms	Hsu et al. (2007); Karaman & Frazzoli (2011); Ladd & Kavraki (2004); LaValle (2006); Lozano-Pérez & Wesley (1979); Weiss et al. (2017)
Potential field formulation	Chen & Wang (2005); Filotheou et al. (2018); Rimon & Koditschek (1992); Vlantis et al. (2018); Vrohidis et al. (2018)
Convex relaxation	Mao et al. (2017, 2016), Liu & Lu (2014) Harris & Açıkmeşe (2014), Rey & Hijazi (2017) Frasch et al. (2013); Janeček et al. (2017a); Yu et al. (2014)

of the problems to be solved. Additionally, another aspect that can be viewed as an open issue and, in some sense, is generated by the previous ones is that there are not so many works in the field of MIP-based motion planning which validate the theoretical results over, at least, experimental platforms, even though there are many applications able to benefit from their use, e.g., search and rescue, environmental measurements tasks, area coverage and the like.

This article has reviewed some recent research and development in MIP-based motion planning. In addition to the results reviewed above, there are many other publications treating MIP formulations in feedback decision making that could not find a direct link with the scope of the motion planning in spite of our best effort. Covering all the variety of application and often inhomogenous problem formulations involving MIP is a task that can be challenging even for a review.

ACKNOWLEDGMENT

The work of Daniel Ioan is financially supported by the Ministry of the Armed Forces - Defence Procurement Agency (DGA) - no.2017352. The authors acknowledge Jacques Blanc-Talon, expert DGA, for the fruitful discussions.

The work of Florin Stoican was supported by a grant of the Romanian Ministry of Education and Research, CNCS - UEFISCDI, project number PN-III-P1-1.1-TE-2019-1614, within PNCDI III.

The authors would like to thank the reviewers for all of their valuable and insightful comments.

References

Abboud, A., Başтuğ, E., Hamidouche, K., & Debbah, M. (2015). Distributed caching in 5g networks: An alternating direction method of multipliers approach. In *2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)* (pp. 171–175). IEEE.

Achterberg, T. (2009). Scip: solving constraint integer programs. *Mathematical Programming Computation*, *1*, 1–41.

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*, 3662–3669.

Afonso, R. J., Maximo, M. R., & Galvão, R. K. (2020). Task allocation and trajectory planning for multiple agents in the presence of obstacle and connectivity constraints with mixed-integer linear programming. *International Journal of Robust and Nonlinear Control*, *30*, 5464–5491.

Afonso, R. J. M., Galvão, R. K. H., & Kienitz, K. H. (2013). Waypoint trajectory planning in the presence of obstacles with a tunnel-MILP approach. In *2013 European Control Conference (ECC)* (pp. 1390–1397).

Alighanbari, M., Kuwata, Y., & How, J. P. (2003). Coordination and control of multiple UAVs with timing constraints and loitering. In *2003 American Control Conference (ACC)* (pp. 5311–5316).

Altché, F., & Fortelle, A. d. L. (2017). Partitioning of the free space-time for on-road navigation of autonomous ground vehicles. In *2017 IEEE 56th Conference on Decision and Control (CDC)* (pp. 2126–2133).

Bahiense, L., Oliveira, G. C., Pereira, M., & Granville, S. (2001). A mixed integer disjunctive model for transmission network expansion. *IEEE Transactions on Power Systems*, *16*, 560–565.

Bajestani, M. A., & Beck, J. C. (2013). Scheduling a Dynamic Aircraft Repair Shop with Limited Repair Resources. *Journal of Artificial Intelligence Research*, *47*, 35–70.

Bali, C., & Richards, A. (2018). Merging vehicles at junctions using mixed-integer model predictive control. In *2018 European Control Conference (ECC)* (pp. 1740–1745). IEEE.

Ballesteros-Tolosana, I., Olaru, S., Rodriguez-Ayerbe, P., Pita-Gil, G., & Deborne, R. (2017). Collision-free trajectory planning for overtaking on highways. In *2017 IEEE 56th Conference on Decision and Control (CDC)* (pp. 2551–2556).

Barraquand, J., Kavraki, L., Latombe, J.-C., Motwani, R., Li, T.-Y., & Raghavan, P. (1997). A random sampling scheme for path planning. *The International Journal of Robotics Research*, *16*, 759–774.

Beard, R. W., & McLain, T. W. (2012). *Small unmanned aircraft: Theory and practice*. Princeton University press.

Bellingham, J., Richards, A., & How, J. P. (2002a). Receding horizon control of autonomous aerial vehicles. In *2002 American Control Conference (ACC)* (pp. 3741–3746). IEEE volume 5.

Bellingham, J. S., Tillerson, M., Alighanbari, M., & How, J. P. (2002b). Cooperative path planning for multiple UAVs in dynamic and uncertain environments. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.* (pp. 2816–2822).

Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., & Mahajan, A. (2013). Mixed-integer nonlinear optimization. *Acta*

Numerica, *22*, 1–131.

Bemporad, A. (2015). Solving mixed-integer quadratic programs via nonnegative least squares. *IFAC-PapersOnLine*, *48*, 73–79.

Bemporad, A., & Mignone, D. (2000). miqp. m: A matlab function for solving mixed integer quadratic programs version 1.02 user guide. *Institut für Automatik, ETH–Swiss Federal Institute of Technology*, .

Bemporad, A., Mignone, D., & Morari, M. (1999). An efficient branch and bound algorithm for state estimation and control of hybrid systems. In *1999 European Control Conference (ECC)* (pp. 557–562). IEEE.

Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, *35*, 407–427.

Bemporad, A., Roll, J., & Ljung, L. (2001). Identification of hybrid systems via mixed-integer programming. In *40th IEEE Conference on Decision and Control* (pp. 786–792). IEEE.

Berntorp, K., Danielson, C., Weiss, A., & Di Cairano, S. (2018). Positive invariant sets for safe integrated vehicle motion planning and control. In *2018 IEEE Conference on Decision and Control (CDC)* (pp. 6957–6962). IEEE.

Berthold, T., Gamrath, G., Gleixner, A., Heinz, S., Koch, T., & Shinano, Y. (2012). *Solving mixed integer linear and nonlinear problems using the SCIP Optimization Suite*. Technical Report 12-27, ZIB,Berlin.

Bethke, B., Valenti, M., & How, J. P. (2008). UAV Task Assignment. *IEEE Robotics Automation Magazine*, *15*, 39–44.

Cao, Y., Yu, W., Ren, W., & Chen, G. (2012). An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, *9*, 427–438.

Cetin, B., Bikdash, M., & Hadaegh, F. (2007). Hybrid mixed-logical linear programming algorithm for collision-free optimal path planning. *IET Control Theory & Applications*, *1*, 522–531.

Chen, M., Shih, J. C., & Tomlin, C. J. (2016a). Multi-vehicle collision avoidance via hamilton-jacobi reachability and mixed integer programming. In *2016 IEEE 55th Conference on Decision and Control (CDC)* (pp. 1695–1700).

Chen, Y.-b., Luo, G.-c., Mei, Y.-s., Yu, J.-q., & Su, X.-l. (2016b). UAV path planning using artificial potential field method updated by optimal control theory. *International Journal of Systems Science*, *47*, 1407–1420.

Chen, Y. Q., & Wang, Z. (2005). Formation control: a review and a new consideration. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3181–3186).

Coffrin, C., Hijazi, H., & Hentenryck, P. V. (2017). The QC relaxation: A theoretical and computational study on optimal power flow. In *2017 IEEE Power Energy Society General Meeting* (pp. 1–1).

CPLEX, I. I. (2009). V12. 1: User's manual for cplex. *International Business Machines Corporation*, *46*, 157.

Culligan, K. F. (2006). *Online trajectory planning for UAVs using mixed integer linear programming*. Thesis Massachusetts Institute of Technology. URL: http://dspace.mit.edu/handle/1721.1/37952.

Dantzig, G., Fulkerson, R., & Johnson, S. (1954). Solution of a Large-Scale Traveling-Salesman Problem. *Journal of the Operations Research Society of America*, *2*, 393–410.

Deits, R., & Tedrake, R. (2015a). Computing large convex regions of obstacle-free space through semidefinite programming. In *Algorithmic foundations of robotics XI* (pp. 109–124). Springer.

Deits, R., & Tedrake, R. (2015b). Efficient mixed-integer planning for UAVs in cluttered environments. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 42–49).

Diamond, S., & Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, *17*, 1–5.

Diehl, M. (2014). Lecture notes on optimal control and estimation.

Dueri, D., Mao, Y., Mian, Z., Ding, J., & Açikmeşe, B. (2017). Trajectory optimization with inter-sample obstacle avoidance via successive convexification. In *2017 IEEE 56th Conference on Decision and Control (CDC)* (pp. 1150–1156).

Dunning, I., Huchette, J., & Lubin, M. (2017). JuMP: A Modeling Language for Mathematical Optimization. *SIAM Review*, *59*, 295–320.

Earl, M. G., & D'Andrea, R. (2002). Modeling and control of a multi-agent system using mixed integer linear programming. In *41st IEEE Conference on Decision and Control, 2002.* (pp. 107–111 vol.1). volume 1.

Earl, M. G., & D'Andrea, R. (2005a). Iterative MILP methods for vehicle-control problems. *IEEE Transactions on Robotics*, *21*, 1158–1167.

Earl, M. G., & D'Andrea, R. (2005b). Multi-Vehicle Cooperative Control Using Mixed Integer Linear Programming. *arXiv:cs/0501092*, . URL: http://arxiv.org/abs/cs/0501092.

Fayazi, S. A., Vahidi, A., & Luckow, A. (2017). Optimal scheduling of autonomous vehicle arrivals at intelligent intersections via MILP. In *2017 American Control Conference (ACC)* (pp. 4920–4925).

Feng, X., Villanueva, M. E., Chachuat, B., & Houska, B. (2017). Branch-and-lift algorithm for obstacle avoidance control. In *2017 IEEE 56th Conference on Decision and Control (CDC)* (pp. 745–750).

Filotheou, A., Nikou, A., & Dimarogonas, D. V. (2018). Decentralized control of uncertain multi-agent systems with connectivity maintenance and collision avoidance. In *2018 European Control Conference (ECC)* (pp. 8–13). IEEE.

Fischetti, M., Glover, F., & Lodi, A. (2005). The feasibility pump. *Mathematical Programming*, *104*, 91–104.

Fischetti, M., & Salvagnin, D. (2009). Feasibility pump 2.0. *Mathematical Programming Computation*, *1*, 201–222.

Fourer, R., Gay, D. M., & Kernighan, B. W. (1987). *AMPL: A mathematical programming language*. AT & T Bell Laboratories Murray Hill, NJ 07974.

Franzè, G., & Lucia, W. (2015). The obstacle avoidance motion planning problem for autonomous vehicles: A low-demanding receding horizon control scheme. *Systems & Control Letters*, *77*, 1–10.

Frasch, J. V., Gray, A., Zanon, M., Ferreau, H. J., Sager, S., Borrelli, F., & Diehl, M. (2013). An auto-generated nonlinear mpc algorithm for real-time obstacle avoidance of ground vehicles. In *Control Conference (ECC), 2013 European* (pp. 4136–4141). IEEE.

Garg, D., Patterson, M. A., Francolin, C., Darby, C. L., Huntington, G. T., Hager, W. W., & Rao, A. V. (2011). Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a radau pseudospectral method. *Computational Optimization and Applications*, *49*, 335–358.

Haghighi, I., Sadraddini, S., & Belta, C. (2016). Robotic swarm control from spatio-temporal specifications. In *2016 IEEE 55th Conference on Decision and Control (CDC)* (pp. 5708–5713).

Harris, M. W., & Açıkmeşe, B. (2014). Lossless convexification of non-convex optimal control problems for state constrained linear systems. *Automatica*, *50*, 2304–2311.

Hart, W. E., Laird, C. D., Watson, J.-P., Woodruff, D. L., Hackebeil, G. A., Nicholson, B. L., & Siirola, J. D. (2017). *Pyomo–optimization modeling in python* volume 67. (2nd ed.). Springer Science & Business Media.

Herceg, M., Kvasnica, M., Jones, C., & Morari, M. (2013). Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference* (pp. 502–510). Zürich, Switzerland. http://control.ee.ethz.ch/~mpt.

Hespanhol, P., Quirynen, R., & Di Cairano, S. (2019). A structure exploiting branch-and-bound algorithm for mixed-integer model predictive control. In *2019 18th European Control Conference (ECC)* (pp. 2763–2768). IEEE.

Hooker, J. N. (2011). *Integrated methods for optimization* volume 170. Springer Science & Business Media.

Hsu, D., Latombe, J.-C., & Kurniawati, H. (2007). On the probabilistic foundations of probabilistic roadmap planning. In *Robotics Research* (pp. 83–97). Springer.

Huang, X., & Peng, H. (2017). Speed trajectory planning at signalized intersections using sequential convex optimization. In *2017 American Control Conference (ACC)* (pp. 2992–2997).

Inc., G. (2014). Gurobi optimizer reference manual. *URL: http://www. gurobi. com*, .

Ioan, D., Prodan, I., Stoican, F., Olaru, S., & Niculescu, S.-I. (2019). Complexity bounds for obstacle avoidance within a zonotopic framework. In *2019 American Control Conference (ACC)* (pp. 335–340). IEEE.

Janeček, F., Klaučo, M., Kalúz, M., & Kvasnica, M. (2017a). OPTIPLAN: A Matlab Toolbox for Model Predictive Control with Obstacle Avoidance. *IFAC-PapersOnLine*, *50*, 531–536.

Janeček, F., Klaučo, M., & Kvasnica, M. (2017b). Trajectory planning and following for uavs with nonlinear dynamics. In *2017 21st International Conference on Process Control (PC)* (pp. 333–338). IEEE.

Jawad, H., Nordin, R., Gharghan, S., Jawad, A., & Ismail, M. (2017). Energy-efficient wireless sensor networks for precision agriculture: A review. *Sensors*, *17*, 1781.

Jünger, M., Liebling, T. M., Naddef, D., Nemhauser, G. L., Pulleyblank, W. R., Reinelt, G., Rinaldi, G., & Wolsey, L. A. (2009). *50 Years of integer programming 1958-2008: From the early years to the state-of-the-art*. Springer Science & Business Media.

Kanno, Y., & Kitayama, S. (2018). Alternating direction method of multipliers as a simple effective heuristic for mixed-integer nonlinear optimization. *Structural and Multidisciplinary Optimization*, *58*, 1291–1295.

Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, *30*, 846–894.

Khan, T. F., & Kumar, D. S. (2016). Mobile collector aided energy reduced (mcer) data collection in agricultural wireless sensor networks. In *2016 IEEE 6th International Conference on Advanced Computing (IACC)* (pp. 629–633). IEEE.

Kirk, D. E. (2012). *Optimal Control Theory: An Introduction*. Courier Corporation.

Kuchar, J. K., & Yang, L. C. (2000). A review of conflict detection and resolution modeling methods. *IEEE Transactions on Intelligent Transportation Systems*, *1*, 179–189.

Kuhn, W. (1998). Rigorously computed orbits of dynamical systems without the wrapping effect. *Computing*, *61*, 47–67.

Ladd, A. M., & Kavraki, L. E. (2004). Measure theoretic analysis of probabilistic path planning. *IEEE Transactions on Robotics and Automation*, *20*, 229–242.

Latombe, J.-C. (2012). *Robot motion planning* volume 124. Springer Science & Business Media.

LaValle, S. M. (2006). *Planning algorithms*. Cambridge University press.

LaValle, S. M., & Kuffner, J. J. (2001). Randomized Kinodynamic Planning. *The International Journal of Robotics Research*, (p. 23).

Lee, S., & Grossmann, I. E. (2000). New algorithms for nonlinear generalized disjunctive programming. *Computers & Chemical Engineering*, *24*, 2125–2141.

Legg, S., Benavides-Serrano, A., Siirola, J. D., Watson, J.-P., Davis, S., Bratteteig, A., & Laird, C. D. (2012). A stochastic programming approach for gas detector placement using cfd-based dispersion simulations. *Computers & Chemical Engineering*, *47*, 194–201.

Leonard, J., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., Fletcher, L., Frazzoli, E., Huang, A., Karaman, S. et al. (2008). A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, *25*, 727–774.

Liu, X., & Lu, P. (2014). Solving Nonconvex Optimal Control Problems by Convex Optimization. *Journal of Guidance, Control, and Dynamics*, *37*, 750–765.

Liu, Z., Dai, J., Wu, B., & Lin, H. (2017). Communication-aware motion planning for multi-agent systems from signal temporal logic specifications. In *2017 American Control Conference (ACC)* (pp. 2516–2521).

Lofberg, J. (2004). Yalmip: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on* (pp. 284–289). IEEE.

Lozano-Pérez, T., & Wesley, M. A. (1979). An Algorithm for Planning Collision-free Paths Among Polyhedral Obstacles. *Commun. ACM*, *22*, 560–570.

Lubin, M., Yamangil, E., Bent, R., & Vielma, J. P. (2018). Polyhedral approximation in mixed-integer convex optimization. *Mathematical Programming*, *172*, 139–168.

Maestre, J. M., Negenborn, R. R. et al. (2014). *Distributed model predictive control made easy* volume 69. Springer.

Maia, M. H., & Galvao, R. K. (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.

Mao, Y., Dueri, D., Szmuk, M., & Açıkmeşe, B. (2017). Successive Convexification of Non-Convex Optimal Control Problems with State Constraints. *IFAC-PapersOnLine*, *50*, 4063–4069.

Mao, Y., Szmuk, M., & Acikmese, B. (2016). Successive Convexification of Non-Convex Optimal Control Problems and Its Convergence Properties. *arXiv:1608.05133 [math]*, (pp. 3636–3641). ArXiv: 1608.05133.

Marafioti, G., Stoican, F., Bitmead, R. R., & Hovd, M. (2012). Persistently exciting model predictive control for siso systems. *IFAC Proceedings Volumes*, *45*, 448–453.

Martin, P., Murray, R. M., & Rouchon, P. (2002). *Flat systems*. Technical Report No.INIS-XA–855.

Matai, R., Singh, S. P., & Mittal, M. L. (2010). Traveling salesman problem: an overview of applications, formulations, and solution approaches. *Traveling salesman problem, theory and applications*, *1*.

Mathur, P., Nielsen, R. H., Prasad, N. R., & Prasad, R. (2016). Data collection using miniature aerial vehicles in wireless sensor networks. *IET Wireless Sensor Systems*, *6*, 17–25.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*, 789–814.

Mercier, A., & Soumis, F. (2007). An integrated aircraft routing, crew scheduling and flight retiming model. *Computers & Operations Research*, *34*, 2251–2265.

Miertoiu, F. I., & Dumitrescu, B. (2019). Feasibility pump algorithm for sparse representation under laplacian noise. *Mathematical Problems in Engineering*, *vol. 2019*. URL: https://doi.org/10.1155/2019/5615243.

Molinari, F., Anh, N. N., & Re, L. D. (2017). Efficient mixed integer programming for autonomous overtaking. In *2017 American Control Conference (ACC)* (pp. 2303–2308).

Mosek, A. (2015). The mosek optimization toolbox for matlab manual. *Version 7.1 (Revision 28)*, (p. 17).

Mukai, M., Natori, H., & Fujita, M. (2017). Model predictive control with a mixed integer programming for merging path generation on motor way. In *2017 IEEE Conference on Control Technology and Applications (CCTA)* (pp. 2214–2219).

Murray, R. M. et al. (2009). Optimization-based control. *California Institute of Technology, CA*, (pp. 111–128).

Naik, V. V., & Bemporad, A. (2017). Embedded mixed-integer quadratic optimization using accelerated dual gradient projection. *IFAC-PapersOnLine*, *50*, 10723–10728.

Papen, A., Vandenhoeck, R., Bolting, J., & Defay, F. (2017). Collision-Free Rendezvous Maneuvers for Formations of Unmanned Aerial Vehicles. *IFAC-PapersOnLine*, *50*, 282–289.

Prodan, I., Olaru, S., Stoica, C., & Niculescu, S.-I. (2012). On the tight formation for multi-agent dynamical systems. In *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications* (pp. 554–565). Springer.

Prodan, I., Stoican, F., Olaru, S., & Niculescu, S.-I. (2015). *Mixed-integer representations in control design: Mathematical foundations and applications*. Springer.

Qu, Z. (2009). *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles*. London: Springer-Verlag.

Quaritsch, M., Kruggl, K., Wischounig-Strucl, D., Bhattacharya, S., Shah, M., & Rinner, B. (2010). Networked uavs as aerial sensor network for disaster management applications. *Elektrotechnik und Informationstechnik*, *127*, 56–63.

Ragi, S., & Mittelmann, H. D. (2017). Mixed-integer nonlinear programming formulation of a UAV path optimization problem. In *2017 American Control Conference (ACC)* (pp. 406–411).

Raman, R., & Grossmann, I. E. (1994). Modelling and computational

techniques for logic based integer programming. *Computers & Chemical Engineering*, *18*, 563–578.

Rawlings, J. B., & Mayne, D. Q. (2009). *Model predictive control: Theory and design*. Nob Hill Pub. Madison, Wisconsin.

Rey, D., & Hijazi, H. (2017). Complex number formulation and convex relaxations for aircraft conflict resolution. In *2017 IEEE 56th Conference on Decision and Control (CDC)* (pp. 88–93).

Richards, A., & How, J. (2003). Performance evaluation of rendezvous using model predictive control. In *AIAA Guidance, Navigation, and Control Conference and Exhibit* (p. 5507).

Richards, A., & How, J. (2005). Mixed-integer programming for control. In *2005 American Control Conference* (pp. 2676–2683 vol. 4).

Richards, A., & How, J. P. (2002). Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *2002 American Control Conference* (pp. 1936–1941). IEEE volume 3.

Richards, A., & Turnbull, O. (2015). Inter-sample avoidance in trajectory optimizers using mixed-integer linear programming. *International Journal of Robust Nonlinear Control*, *25*, 521–526. doi:10.1002/rnc.3101.

Richards, A. G., & Turnbull, O. (2013). Inter-sample avoidance in trajectory optimizers using mixed-integer linear programming. In *AIAA Guidance, Navigation, and Control (GNC) Conference* (p. 4634).

Rimon, E., & Koditschek, D. E. (1992). Exact robot navigation using artificial potential functions. *Departmental Papers (ESE)*, (p. 323).

Ritter, T., Euler, J., Ulbrich, S., & Stryk, O. v. (2014). Adaptive Observation Strategy for Dispersion Process Estimation Using Cooperating Mobile Sensors. *IFAC Proceedings Volumes*, *47*, 5302–5308.

Roll, J., Bemporad, A., & Ljung, L. (2004). Identification of piecewise affine systems via mixed-integer programming. *Automatica*, *40*, 37–50.

Schouwenaars, T. (2006). *Safe trajectory planning of autonomous vehicles*. Thesis Massachusetts Institute of Technology. URL: http://dspace.mit.edu/handle/1721.1/35298.

Schouwenaars, T., De Moor, B., Feron, E., & How, J. (2001a). Mixed integer programming for multi-vehicle path planning. In *2001 European control conference (ECC)* (pp. 2603–2608). IEEE.

Schouwenaars, T., Richards, A., Feron, E., & How, J. (2001b). Plume avoidance maneuver planning using mixed integer linear programming. In *AIAA Guidance, Navigation, and Control Conference and Exhibit* (p. 4091).

Scott, P., & Thiébaux, S. (2014). Dynamic optimal power flow in microgrids using the alternating direction method of multipliers. *arXiv preprint arXiv:1410.7868*, .

Smith, J. C., & Taskin, Z. C. (2008). A tutorial guide to mixed-integer programming models and solution techniques. *Optimization in Medicine and Biology*, (pp. 521–548).

Sontag, E. D. (2013). *Mathematical control theory: deterministic finite dimensional systems* volume 6. Springer Science & Business Media.

Stellato, B., Naik, V. V., Bemporad, A., Goulart, P., & Boyd, S. (2018). Embedded mixed-integer quadratic optimization using the osqp solver. In *2018 European Control Conference (ECC)* (pp. 1536–1541). IEEE.

Stoican, F., Olaru, S., Seron, M. M., & De Doná, J. A. (2012). Reference governor design for tracking problems with fault detection guarantees. *Journal of Process Control*, *22*, 829–836.

Stoican, F., Prodan, I., & Grøtli, E. I. (2018). Exact and overapproximated guarantees for corner cutting avoidance in a multiobstacle environment. *International Journal of Robust and Nonlinear Control*, *28*, 4528–4548.

Stoican, F., Prodan, I., & Olaru, S. (2013). Hyperplane arrangements in mixed-integer programming techniques. collision avoidance application with zonotopic sets. In *Control Conference (ECC), 2013 European* (pp. 3155–3160). IEEE.

Stubbs, R. A., & Mehrotra, S. (1999). A branch-and-cut method for 0-1 mixed convex programming. *Mathematical programming*, *86*, 515–532.

Sun, X., & Cassandras, C. G. (2015). Optimal dynamic formation control of multi-agent systems in environments with obstacles. In *2015 54th IEEE Conference on Decision and Control (CDC)* (pp. 2359–2364).

Taccari, L. (2016). Integer programming formulations for the elementary shortest path problem. *European Journal of Operational Research*, *252*, 122–130.

Takapoui, R., Moehle, N., Boyd, S., & Bemporad, A. (2020). A simple effective heuristic for embedded mixed-integer quadratic programming. *International Journal of Control*, *93*, 2–12.

Testa, A., Rucco, A., & Notarstefano, G. (2017). A finite-time cutting plane algorithm for distributed mixed integer linear programming. In *2017 IEEE 56th Conference on Decision and Control (CDC)* (pp. 3847–3852).

Timotheou, S., Panayiotou, C. G., & Polycarpou, M. M. (2014). Distributed traffic signal control using the cell transmission model via the alternating direction method of multipliers. *IEEE Transactions on Intelligent Transportation Systems*, *16*, 919–933.

Van Leeuwen, J., & Leeuwen, J. (1990). *Handbook of theoretical computer science: Algorithms and complexity* volume 1. Elsevier.

Van Parys, R., & Pipeleers, G. (2016). Online distributed motion planning for multi-vehicle systems. In *2016 European Control Conference (ECC)* (pp. 1580–1585). IEEE.

Vielma, J. (2015). Mixed Integer Linear Programming Formulation Techniques. *SIAM Review*, *57*, 3–57.

Vielma, J. P., & Nemhauser, G. L. (2011). Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming*, *128*, 49–72.

Vitus, M., Pradeep, V., Hoffmann, G., Waslander, S., & Tomlin, C. (2008). Tunnel-milp: Path planning with sequential convex polytopes. In *AIAA guidance, navigation and control conference and exhibit* (p. 7132).

Vlantis, P., Vrohidis, C., Bechlioulis, C. P., & Kyriakopoulos, K. J. (2018). Robot navigation in complex workspaces using harmonic maps. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1726–1731). doi:10.1109/ICRA.2018.8460695.

Vrohidis, C., Vlantis, P., Bechlioulis, C. P., & Kyriakopoulos, K. J. (2018). Prescribed time scale robot navigation. *IEEE Robotics and Automation Letters*, *3*, 1191–1198. doi:10.1109/LRA.2018.2794616.

Vujanic, R., Esfahani, P. M., Goulart, P. J., Mariéthoz, S., & Morari, M. (2016). A decomposition method for large scale milps, with performance guarantees and a power system application. *Automatica*, *67*, 144–156.

Wang, C., Ma, F., Yan, J., De, D., & Das, S. K. (2015a). Efficient aerial data collection with uav in large-scale wireless sensor networks. *International Journal of Distributed Sensor Networks*, *11*, 286080.

Wang, X., Kloetzer, M., Mahulea, C., & Silva, M. (2015b). Collision avoidance of mobile robots by using initial time delays. In *2015 54th IEEE Conference on Decision and Control (CDC)* (pp. 324–329).

Weiss, A., Danielson, C., Berntorp, K., Kolmanovsky, I., & Cairano, S. D. (2017). Motion planning with invariant set trees. In *2017 IEEE Conference on Control Technology and Applications (CCTA)* (pp. 1625–1630).

Welder, L., Ryberg, D. S., Kotzur, L., Grube, T., Robinius, M., & Stolten, D. (2018). Spatio-temporal optimization of a future energy system for power-to-hydrogen applications in Germany. *Energy*, *158*, 1130–1149.

Williams, H. P. (2013). *Model building in mathematical programming*. John Wiley & Sons.

Xu, Y., Xiao, L., Yang, D., Cuthbert, L., & Wang, Y. (2018). Energy-efficient uav communication with multiple GTs based on trajectory optimization. *Mobile Information Systems*, *2018*.

Yang, L., Qi, J., Xiao, J., & Yong, X. (2014). A literature review of UAV 3d path planning. In *Proceeding of the 11th World Congress on Intelligent Control and Automation* (pp. 2376–2381).

Yu, H., Wang, Y., Bortoff, S. A., & Ueda, K. (2014). Energy-Efficient

Trajectory Planning for a Mobile Agent by Using a Two-Stage Decomposition Approach. *IFAC Proceedings Volumes*, *47*, 3851–3856.

Yue, W., & Jiang, Z. (2018). Path planning for uav to collect sensors data based on spiral decomposition. *Procedia computer science*, *131*, 873–879.

Zhang, Y., Su, R., Gao, K., & Zhang, Y. (2017a). Traffic light scheduling for pedestrians and vehicles. In *2017 IEEE Conference on Control Technology and Applications (CCTA)* (pp. 1593–1598).

Zhang, Y., Su, R., Sun, C., & Zhang, Y. (2017b). Modelling and traffic signal control of a heterogeneous traffic network with signalized and non-signalized intersections. In *2017 IEEE Conference on Control Technology and Applications (CCTA)* (pp. 1581–1586).

Zidek, R. A., Bemporad, A., & Kolmanovsky, I. V. (2017a). Optimal and receding horizon drift counteraction control: Linear programming approaches. In *2017 American Control Conference (ACC)* (pp. 2636–2641). IEEE.

Zidek, R. A., Petersen, C. D., Bemporad, A., & Kolmanovsky, I. V. (2017b). Receding horizon drift counteraction and its application to spacecraft attitude control. In *27th AAS/AIAA Space flight mechanics meeting*.

Ziegler, G. M. (2012). *Lectures on polytopes* volume 152. Springer Science & Business Media.