2017 IEEE 56th Annual Conference on Decision and Control (CDC)
December 12-15, 2017, Melbourne, Australia

Branch-and-Lift algorithm for obstacle avoidance control

Xuhui Feng*, Mario E. Villanueva, Benoît Chachuat, Boris Houska

Abstract— **Obstacle avoidance problems are a class of non-convex optimal control problems for which derivative-based optimization algorithms often fail to locate global minima. The goal of this paper is to provide a tutorial on how to apply Branch & Lift algorithms, a novel class of global optimal control methods, for solving such obstacle avoidance problems to global optimality. The focus of the technical developments is on how Branch & Lift methods can exploit the particular structure of Dubin models, which can be used to model a variety of practical obstacle avoidance problems. The global convergence properties of Branch & Lift in the context of obstacle avoidance is discussed from a theoretical as well as a practical perspective by applying it to a tutorial example.**

I. INTRODUCTION

Obstacle and collision avoidance control methods are needed in many fields of engineering of current interest such as mechatronics and robotics [11], [22] as well as traffic control of autonomous vehicles at intersections [6], [20]. In principle, such obstacle avoidance problems for dynamic devices can be formulated as optimal control problems [7], [11]. However, these optimal control problems are severely non-convex, especially, due to state constraints enforcing the distance of the controlled device to the closest obstacles to be positive [13].

Unfortunately, many state-of-the-art optimal control solvers [18] are based on local or derivative-based optimization methods, which often get trapped into local minima. One exception are algorithms based on dynamic programming [14], which can be used to solve generic optimal control problems to global optimality, but their application is limited to systems with a small number of states [2]. Similarly, one could apply deterministic global optimization methods [15] such as Branch & Bound to the discretized optimal control problem. Such approaches have for example been suggested in [5], [9], [10] for dynamic optimization problems with a small number of optimization variables. These methods scale reasonably well with the number of states. However, as soon as such methods are used in higher-dimensional search spaces, e.g., if we have problems with time-varying control inputs, Branch & Bound is basically not applicable. Thus, direct discretization methods are not suited for global optimization in obstacle avoidance control.

The lack of generic global optimal control solvers has led researchers to develop tailored methods for solving, or at least approximately solving, collision and obstacle avoidance problems to reasonable accuracies [3], [12], [11], [21], [24]. However, these methods are often based on various simplification or modeling assumptions, which often neglect the dynamics of the system in order to simplify the problem. Of course, this does not imply that these heuristics are not suited for practical use, but the main focus of this paper is on solving obstacle avoidance problems with moderately high numerical accuracies without restrictive modeling assumptions.

This paper analyzes and illustrates the effectiveness of Branch & Lift methods for solving obstacle avoidance control problems to global optimality. Branch & Lift methods have been developed recently by two of the authors [16]. The mathematical convergence properties of these methods have been analyzed by using functional analysis methods in a rather general setting in potentially infinite dimensional Hilbert spaces [17]. One principal goal of this paper is to make these abstract mathematical developments accessible to control engineers. Therefore, this paper briefly summarizes the main of idea of Branch & Lift, but then focusses on its application to a tutorial problem from the field of obstacle avoidance control. In this context, it is also discussed how the proposed method can cope with large-scale problems.

Section II introduces the problem formulation and Section II-A introduces a tutorial obstacle avoidance control problem, which is used as a guiding example to visualize all technical and algorithmic developments throughout this paper. Section III introduces the concept of moment-constrained reachable sets, which are in this paper identified as the basis for Branch & Lift algorithms. The corresponding developments are illustrated step-by-step for the tutorial problem in Section III-A. Section IV introduces the practical algorithm and summarizes its theoretical convergence properties. Section V discusses the application of Branch & Lift to the tutorial problem as well as to general obstacle avoidance control problems. Section VI concludes the paper.

Notation: Besides mathematical standard notation, we denote with $L_2[0,T]^n$ the set of n-dimensional square-integrable functions on the interval $[0,T]$. We use the syntax

$$\forall x, y \in L_2[0,T]^n, \qquad \langle x, y \rangle = \int_0^T x(t)^\mathsf{T} y(t) \, \mathrm{d}t$$

* Corresponding author.

Xuhui Feng is with the School of Information Science and Technology, ShanghaiTech University and Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China. fengxh@shanghaitech.edu.cn

Mario E. Villanueva is with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China, and Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, USA. meduardov@shanghaitech.edu.cn

Benoît Chachuat is with Centre for Process Systems Engineering, Imperial College London, London, UK. b.chachuat@ic.ac.uk

Boris Houska is with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. borish@shanghaitech.edu.cn

978-1-5090-2873-3/17/$31.00 ©2017 IEEE

to denote the associated inner product. Moreover, $W_{1,2}[0,T]^n$ denotes the associated Sobolev space of weakly differentiable and square-integrable functions on $[0,T]$ with square-integrable weak derivatives. The diameter of a set $X \subseteq \mathbb{R}^n$ is denoted by

$$\operatorname{diam}(X) = \max_{x,y\in X} \|x-y\|_2 \ .$$

The complement of X in $\mathbb{R}^{n_x}$ is denoted by $\mathbb{R}^{n_x} \setminus X$.

II. Obstacle avoidance control problems

Open-loop obstacle avoidance problems can be cast as optimal control problems of the form

$$\mathcal{V} \ = \ \min_{x,u} \quad m(x(T)) \qquad \text{s.t.} \begin{cases} \forall t \in [0,T], \\ \dot{x}(t) = f(x(t)) + B(x(t))u(t), \\ x(0) = x_0, \\ x(t) \in \mathbb{X} \ , \ u(t) \in \mathbb{U} \ , \end{cases} \tag{1}$$

where $x \in W_{1,2}[0,T]^{n_x}$ denotes the states and $u \in L_2[0,T]^{n_u}$ the controls. Here, the functions $f : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, $B : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x \times n_u}$, and $m : \mathbb{R}^{n_x} \to \mathbb{R}$ are globally Lipschitz continuous and the initial value $x_0 \in \mathbb{R}^{n_x}$ is assumed to be given. Notice that robot and mechatronics models of practical relevance (e.g. based on Newton mechanics or descriptor forms) can be written in the above form. For example, control forces or torques of robot models enter affinely via Newton's equations of motion, i.e., the corresponding model equations are affine in u. Nonlinear effects such as centrifugal, non-trivial friction, or Coriolis forces depend on the state x only and can be modeled by choosing the function f and B appropriately. The control constraint set $\mathbb{U} \subseteq \mathbb{R}^{n_u}$ can be assumed to be compact and convex—for practical obstacle avoidance problems, $\mathbb{U}$ is a simple interval box modeling control bounds. This is in contrast to the set $\mathbb{X}$, which is non-convex for most obstacle avoidance problems of practical relevance. This is due to the fact that even if each obstacle is modeled as a convex set in the state space, the set of where our mechanical device is allowed to move, namely, the complement of the union of the obstacle set in $\mathbb{R}^{n_x}$, is non-convex. This implies in particular that local optimal control algorithms are not suitable for solving obstacle avoidance problems of the form (1). For simplicity of presentation, it is assumed here that the obstacles are static; $\mathbb{X}$ does not depend on t.

A. Tutorial example

One of the simplest possible models for 2D trajectory planning of mechanical devices is Dubin's model, which is obtained by setting

$$f(x) = \begin{pmatrix} \cos(x_3) \\ \sin(x_3) \\ 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \ . \tag{2}$$

Here, the states x_1 and x_2 can be interpreted as the 2D position coordinates of a moving body with unit mass and constant unit velocity. The state x_3 can be interpreted as an angle determining the direction in which the airplane is flying. The control input $u : [0,T] \to \mathbb{R}$ can be interpreted as the associated angular velocity. In order to set up a simple scenario, we introduce control bounds $\mathbb{U} = [\underline{u}, \overline{u}]$ and one obstacle,

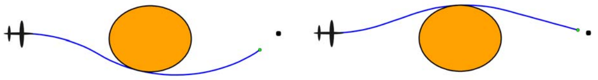

Fig. 1. Visualization of the two locally optimal trajectories on the (x_1, x_2) plane. LEFT: a locally optimal trajectory. RIGHT: the globally optimal trajectory.

$$\mathbb{X} = \left\{ x \mid (x_1 - c_1)^2 + (x_2 - c_2)^2 \geq r^2 \right\} \ .$$

Here, the obstacle is a disc in the (x_1, x_2)-plane with center (c_1, c_2) and radius r and the set $\mathbb{X}$ is defined to be the complement of the set that is occupied by this obstacle. Our goal is to come as close as possible to a target point (d_1, d_2),

$$m(x) = (x_1 - d_1)^2 + (x_2 - d_2)^2 \ ,$$

behind the obstacle. The numerical values for all these parameters are listed in the table below.

Parameter	Symbol	Value
radius of the obstacle	r	0.32
center of the obstacle	(c_1, c_2)	$(1, -0.05)$
control bounds	$[\underline{u}, \overline{u}]$	$[-1, 1]$
target point	(d_1, d_2)	$(2, 0)$
duration	T	2
initial state	x_0	$(0,0,0)^\mathsf{T}$

The above obstacle avoidance optimal control problem has two local minima as visualized in Figure 1. Notice that this visualization uses—in order to better fit the format of this paper—different scales of the x_1 and x_2 coordinates such that the disc-shaped orange region appears as if it were an ellipse. Also notice that the trajectory in the right plot in Figure 1 corresponds to the globally optimal solution—it is slightly better to pass the obstacle on the left as $c_2 = -0.05$ is in this case study slightly negative.

III. Moment-constrained reachable sets

The construction of so-called Branch & Lift algorithms [16] for solving (1) to global optimality relies on the outer approximation of moment-constrained reachable sets $X_M(t, A) \subseteq \mathbb{R}^{n_x}$. For a given orthogonal function basis $\Phi_0, \Phi_1, \Phi_2, \ldots \in L_2[0,T]^n$ with squared norms $\sigma_i = \langle \Phi_i, \Phi_i \rangle$, and for any compact set $A \subset \mathbb{R}^{M+1}$, these sets are defined as

$$X_M(t,A) = \left\{ x(t) \;\middle|\; \begin{array}{l} \forall \tau \in [0,t], \\ \dot{x}(\tau) = f(x(\tau)) + B(x(\tau))u(\tau), \\ x(0) = x_0, \\ u(\tau) \in \mathbb{U}, \\ \left[\frac{\langle \Phi_0, u \rangle}{\sigma_0}, \ldots, \frac{\langle \Phi_M, u \rangle}{\sigma_M} \right]^{\mathsf{T}} \in A \end{array} \right\} .$$

The following theorem has originally been proven [17]. It lays the mathematical foundation for analyzing the convergence properties Branch & Lift, as discussed in Section IV.

Theorem 1: If the functions f and B are globally Lipschitz continuous, and if the set $\mathbb{U}$ is bounded, then there exist constants $C_1, C_2 < \infty$ such that

$$\operatorname{diam}(X_M(t,A)) \;\leq\; C_1 \operatorname{diam}(A) + C_2 \frac{1}{\sqrt{M}} \qquad (3)$$

for all $t \in [0,T]$, all $A \subset \mathbb{R}^{M+1}$, and all $M \in \mathbb{N}$.

In the following, a set $A \subset \mathbb{R}^{M+1}$ is called

1) **infeasible** if

$$\exists t \in [0,T]: \quad X_M(t,A) \subseteq \mathbb{R}^{n_x} \setminus \mathbb{X} \; ;$$

2) **sub-optimal** if

$$\min_{\xi \in X_M(T,A)} m(\xi) \geq \mathcal{U}$$

for a bound $\mathcal{U} \geq \mathcal{V}$ on the optimal value of (1), e.g., a locally optimal solution or any feasible solution of (1), and

3) **undecided** if A is neither sub-optimal nor infeasible.

Notice that if A is sub-optimal or infeasible, then any globally optimal control input u^* of (1) satisfies

$$\left[\frac{\langle \Phi_0, u^* \rangle}{\sigma_0}, \ldots, \frac{\langle \Phi_M, u^* \rangle}{\sigma_M} \right]^{\mathsf{T}} \notin A \; .$$

This statement is an immediate consequence of the above definition. Thus, the moments of a globally optimal control can only be contained in an undecided set.

Notice that there exist numerous tools for computing or outer-approximating the reachable sets of nonlinear differential equations, and we refer to [4], [25] for an overview. In the following, we use a variant of Taylor models with ellipsoidal remainder bounds, as proposed in [16] to describe non-convex convergent outer approximations $X_M(t,A) \subseteq Y_M(t,A)$ of the moment constrained reachable sets $X_M(t,A)$. This is sufficient in the sense that if the outer approximation Y_M satisfies

$$\exists t \in [0,T]: \quad Y_M(t,A) \subseteq \mathbb{R}^{n_x} \setminus \mathbb{X} \; ,$$

then A is infeasible. Similarly, if

$$\min_{\xi \in Y_M(T,A)} m(\xi) \geq \mathcal{U} \; ,$$

then A is sub-optimal. In summary, infeasibility and sub-optimality of a set A can be verified by using standard tools from the field of set based computing [4] such as Taylor- and Chebyshev arithmetics [23].

A. Visualization of moment-constrained reachable sets

In order to visualize moment-constrained reachable sets for our tutorial example from Section II-A, we use the colors red, blue, and green in order to visualize infeasible, sub-optimal, and undecided sets, respectively. Moreover, we use the scalar Legendre polynomial basis

$$\forall k \in \mathbb{N}, \quad \Phi_k(t) = (-1)^k \sum_{l=0}^{k} \begin{pmatrix} k \\ l \end{pmatrix} \begin{pmatrix} k+l \\ l \end{pmatrix} \left(-\frac{t}{T} \right)^l ,$$

since the control input $u \in L_2[0,T]^1$ is scalar for this example. Notice that these polynomials are orthogonal by construction and satisfy $\sigma_k = \langle \Phi_k, \Phi_k \rangle = \frac{T}{2k+1}$. This implies in particular that the zeroth-order moment of the input u, given by

$$a_0 = \frac{1}{\sigma_0} \langle \Phi_0, u \rangle = \frac{1}{T} \int_0^T u(\tau) \, \mathrm{d}\tau \; ,$$

matches the average value of the function u on $[0,T]$. Since the set $\mathbb{U} = [-1,1]$ is convex, we must have $a_0 \in \mathbb{U}$. Next, we consider the exhaustive moment partition $\mathcal{A} = \{A_1, A_2, \ldots, A_5\}$ consisting of 5 adjacent intervals that cover the interval $[-1,1]$, see Figure 2. The upper left part shows the moment-constrained reachable set $\cup_{t \in [0,T]} X_0(t, A_1)$ of $A_1 = [-1,-0.6]$, the first set in the partition $\mathcal{A}$, on the (x_1,x_2)-plane. In words, $\cup_{t \in [0,T]} X_0(t,A_1)$ corresponds to the set of positions that the airplane can reach at times $t \in [0,T]$ by applying a control input $u(t) \in [-1,1]$, which additionally satisfies the moment constraint

$$\frac{1}{T} \int_0^T u(\tau) \, \mathrm{d}\tau \in A_1 = [-1,-0.6] \; .$$

It can be seen from Figure 2 that this means that the airplane may turn right initially, but then it has to turn left on average, as we need to choose a control function whose average is negative. Clearly, the distance of all points in

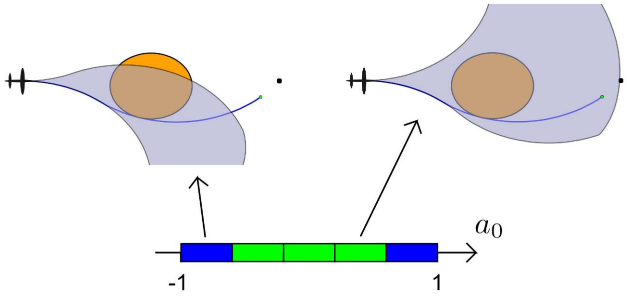

Fig. 2. BOTTOM: Visualization of a one-dimensional moment partition, $\mathcal{A} = \{A_1, A_2, \ldots, A_5\}$, consisting of 5 intervals with respect to the first moment $a_0 = \frac{1}{\sigma_0} \langle u, \Phi_0 \rangle$ for the first Legendre polynomial $\Phi_0(t) = 1$. The intervals $A_1 = [-1,-0.6]$ and $A_5 = [0.6,1]$ are sub-optimal (blue) with respect to the current locally optimal solution trajectory. The intervals $A_2 = [-0.6,-0.2]$, $A_3 = [-0.2,0.2]$, and $A_4 = [0.2,0.6]$ are undecided and therefore colored green. UPPER-LEFT: visualization of the moment constrained reachable set $\cup_{t \in [0,T]} X_0(t,A_1)$ on (x_1,x_2)-plane (gray). UPPER-RIGHT: visualization of the moment constrained reachable set $\cup_{t \in [0,T]} X_0(A_4)$ on (x_1,x_2)-plane (gray).

the set $X_0(t, A_1)$ are sub-optimal compared to the locally optimal solution of (1), which is indicated as the blue trajectory in the upper left part of Figure 2. Consequently, the interval A_1 is colored blue in the bottom of Figure 2 indicating that this interval is sub-optimal. The upper right part of Figure 1 shows the moment-constrained reachable set $\cup_{t\in[0,T]} X_0(t, A_4)$ of the fourth set in the partition, $A_4 = [0.2, 0.6]$, on (x_1, x_2)-plane, which happens to be undecided. Thus, the interval A_4 is colored green. The sets A_2, A_3 are undecided, too, but the set A_5 is sub-optimal compared to the locally optimal current upper bound. A discussion and visualization of moment-constrained reachable sets for our tutorial example for $M > 0$ can be found further below.

IV. Branch-and-Lift Algorithm

Branch & Lift algorithms [16], [17] maintain a set of intervals $\mathcal{A}$ by applying three main algorithmic operations, named branching, lifting, and fathoming.

1) **Branching** refers to the process of substituting an interval $A \in \mathcal{A}$ with smaller intervals whose union contains A.
2) **Lifting** refers to the process of replacing every interval $A \in \mathcal{A}$ with its associated lifted interval $A \times [\underline{a}, \overline{a}]$ and setting $M \leftarrow M + 1$.
3) **Fathoming** refers to the process of deleting all infeasible (red) and sub-optimal (blue) intervals from the current partition $\mathcal{A}$.

The implementation of the branching and fathoming operations is analogous to standard Branch & Bound methods [15]. In order to implement the lifting operation the bounds $[\underline{a}, \overline{a}]$ are computed as the minimizer/maximizer of the infinite dimensional convex optimization problem

$$\begin{array}{rcl} \underline{a} \,/\, \overline{a} & = & \min_u \,/\, \max_u \;\; \frac{\langle \Phi_{M+1}, u\rangle}{\sigma_{M+1}} \\ & & \text{s.t.} \; \left\{ \begin{array}{l} \left[\frac{\langle \Phi_0, u\rangle}{\sigma}, \ldots, \frac{\langle \Phi_M, u\rangle}{\sigma_M} \right]^{\mathsf{T}} \in A \\ \forall t \in [0,T], \quad u(t) \in \mathbb{U} \,, \end{array} \right. \end{array} \tag{4}$$

which happens to be a linear programming problem if $\mathbb{U}$ is an interval. This construction ensures that if the first $M + 1$ moments of a globally optimal control u^* are in an interval A, then the first $M + 2$ moments of u^* are in the lifted interval $A^+ = A \times [\underline{a}, \overline{a}]$. Notice that there exist standard tools for solving the infinite dimensional linear programming problems (4) with high numerical accuracy [1], [8]. In terms of computational effort, another demanding operation is the fathoming step, which requires the outer-approximation of moment-constrained reachable sets as discussed in the previous section.

In order to arrive at a practical implementation, the above fathoming, branching, and lifting steps need to be called in a loop. Here, lifting operations are only applied if the diameter of the outer approximation of the moment-constrained reachable set cannot be improved much further by applying branching operations. This procedure can—at least qualitatively—also be motivated by Theorem 1: if the term $C_1 \mathrm{diam}(A)$ on the right hand of inequality (3) is already small, the diameter of $X_M(t, A)$ can only be improved further by increasing M. Details on how to implement decision heuristics about when to branch and to lift can be found in [16]. Moreover, in order to start and terminate the algorithm, the fathoming, branching, and lifting operations need to be augmented by the following additional steps.

4) **Initialization** of Branch & Lift is implemented setting $M = 0$ and $\mathcal{A} = \{A_0\}$, where the initial interval $A_0 \subset \mathbb{R}$ is such that $\frac{\langle u, \Phi_0\rangle}{\sigma_0} \in A_0$ for all feasible u.
5) **Termination** of Branch & Lift is implemented by stopping the algorithm whenever the termination criterion

$$\mathcal{U} - \min_{A\in\mathcal{A}, \xi\in A} m(\xi) \le \epsilon$$

is satisfied for a user tolerance $\epsilon > 0$ or if $\mathcal{A}$ is empty. In the first case, the algorithm returns a proof of ϵ-optimality, while, in the second case, a certificate of infeasibility is returned.

Notice that the interval $A_0 = [\underline{a}, \overline{a}]$ in the initialization step can be found by setting

$$\begin{array}{rcl} \underline{a} \,/\, \overline{a} & = & \min_u \,/\, \max_u \;\; \frac{\langle \Phi_0, u\rangle}{\sigma_{M+1}} \\ & & \text{s.t.} \qquad \forall t \in [0,T], \quad u(t) \in \mathbb{U} \,. \end{array} \tag{5}$$

In order to implement the termination criterion, we use in this paper a local optimal control solver from ACADO Toolkit [18], which yields upper bounds on $\mathcal{V}$. Here, the local solver is initialized at the mid-point, $\mathrm{mid}(A)$, of each undecided interval $A \in \mathcal{A}$, i.e., by initializing ACADO at the control input $u = \sum_{i=0}^{M} a_i \Phi_i$ with $a = \mathrm{mid}(A)$. The best local minimum found after initializing ACADO at all of these mid-points is used as the current upper bound $\mathcal{U}$ in order to fathom sub-optimal intervals in $\mathcal{A}$. If the local solver does not return a feasible solution, we set $\mathcal{V} = \infty$. A proof of the following worst-case convergence statement can be found in [17].

Theorem 2: If the functions f, B, and m are globally Lipschitz continuous, if $\mathbb{U}$ is compact, and if the local solver finds feasible locally optimal upper bounds, then the outlined Branch & Lift algorithms terminates after at most $\mathbf{O}\left(\exp\left(\epsilon^{-2}\log\left(\epsilon^{-1}\right)\right)\right)$ iterations.

Notice that the complexity estimate in Theorem 2 is a worst-case bound. In practice, especially in obstacle avoidance problems, Branch & Lift performs much better than what can be expected from this exponential worst case bound. Also notice that better complexity estimates are possible if additional structure is available. For example, if the optimal control input is known to be smooth, the worst case complexity of Branch & Lift algorithms reduces to

$$\mathbf{O}\left(\exp\left(\log\left(\epsilon^{-1}\right)^2\right)\right) \,,$$

which has also been proven in [17].

V. Branch & Lift for obstacle avoidance control

In this section we first apply Branch & Lift to the tutorial obstacle avoidance problem from Section II-A in order to explain and visualize the main steps of the algorithm. Applications to more challenging obstacle avoidance control problems are discussed further below.

A. Application of Branch & Lift to the tutorial example

The iterations of Branch & Lift for the problem in Section II-A can be worked out explicitly. Recall that Legendre basis polynomials are used as explained in Section III-A.

- **Iteration 1:** The initialization step selects $A_0 = [-1, 1]$ as explained in Section III-A.
- **Iteration 2:** The second iteration applies a branching operation. In this example, the initial partition is divided into five scalar sub-intervals, $\mathcal{A} = \{A_1, \ldots, A_5\}$.
- **Iteration 3:** The third step applies a fathoming operation, which deletes the sub-optimal intervals A_1 and A_5 from the current partition, i.e., we now have $\mathcal{A} = \{A_2, A_3, A_4\}$. The corresponding computations are visualized in Figure 2.
- **Iteration 4:** The fourth iteration applies a lifting operation. The associated lifted partition $\mathcal{A} = \left\{A_2^+, A_3^+, A_4^+\right\}$ contains three 2-dimensional interval boxes.
- **Iteration 5:** The next branching operation divides all 3 lifted intervals into 5 sub-intervals. The result of this branching operation is visualized in the lower left corner of Figure 3 in the (a_0, a_1)-space, where the variables

$$\begin{array}{rcl} a_0 & = & \frac{1}{T}\int_0^T u(t)\,\mathrm{d}t \\ \text{and} \quad a_1 & = & \frac{3}{T}\int_0^T (2(t/T)-1)u(t)\,\mathrm{d}t \end{array}$$

  corresponds to the zeroth- and first order moments of the steering input u in the Legendre basis.
- **Iteration 6:** All 15 intervals in the current partition are checked for sub-optimality and infeasibility. During this process the local solver, i.e., the upper bounding routine finds a better local minimizer, which in this example happens to be globally optimal. However, the algorithm has no proof of global optimality at this stage and, consequently, keeps iterating. Thereafter, the current fathoming step detects 2 infeasible as well as 8 sub-optimal intervals and deletes all of them from the current partition. The result of this process is shown in the left part of Figure 4: the updated partition contains only 5 remaining undecided intervals.

  Notice that the main topology of the obstacle avoidance problem is reflected in the two dimensional moment space in the sense that the union of the 5 undecided intervals in $\mathcal{A}$ consists of two connected sub-regions (see the left part of Figure 4). These sub-regions correspond to the two main possibilities, namely, passing the obstacle on either the left or right side, although, the algorithm is still undecided about which of these options is better at this stage.
- **Iteration 7:** The result of the next lifting step is shown in the right part of Figure 4. Notice that the associated lifted intervals are three dimensional, after we introduce the new variable

$$a_2 = \frac{5}{T}\int_0^T (6(t/T)^2 - 6(t/T) + 1)u(t)\,\mathrm{d}t\ ,$$

  which corresponds to the second order moment of the steering input u.
- **Iteration 8-20:** If we keep on branching, lifting, and fathoming for a few iterations, termination is achieved. For example, if the termination tolerance is set to $\epsilon = 10^{-3}$ the algorithm stops after 20 iterations and returns a (numerical) proof that the found local solution found in step 6 is ϵ-sub-optimal.

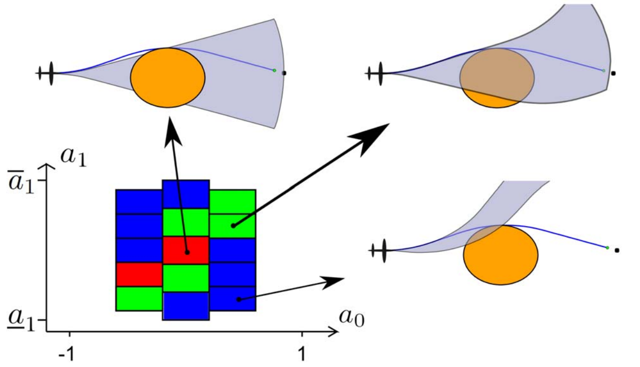

Fig. 3. The lower left plot shows the partition $\mathcal{A}$ in Iteration 6. The red and blue intervals correspond to infeasible and sub-optimal intervals, respectively. The other plots show moment-constrained reachable sets of the infeasible interval in middle (upper left plot), an undecided interval (upper right plot), and a sub-optimal interval (lower right plot) as indicated by the arrows.

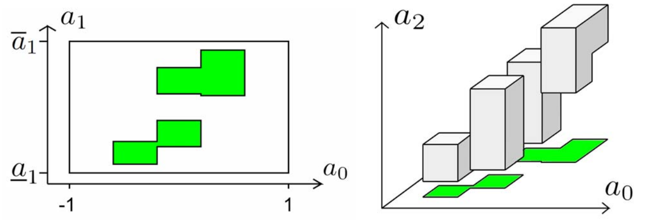

Fig. 4. LEFT: visualization of the remaining 6 undecided intervals in the partition after the fathoming operation in Iteration 6 has been completed. RIGHT: The same partition after a lifting operation has been applied, see Iteration 7 for more details.

B. Performance of Branch & Lift

The above tutorial example has been solved with three different methods, where the implementation of the presented Branch & Lift method uses various set arithmetics from the library MC++ [4]:

Algorithm	Run-Time
Single Shooting + Branch & Bound	$\approx$ 2 days
Dynamic Programming	$\approx$ 1 h
Branch & Lift	$<$ 1 min

We have used the same numerical tolerance of $\epsilon = 10^{-3}$ for all implementations. Branch & Lift outperforms standard

Branch & Bound as well as exhaustive dynamic programming, although the state space is only 3-dimensional in this tutorial example. Notice that if we replace the single obstacle in our tutorial example with, for example, 100 obstacles with random location, Branch & Lift often converges in less than 1min, although the practical performance varies from case-to-case. This is due to the fact that more obstacles lead to more infeasible sub-intervals, which expedites convergence, as the fathoming process helps to accumulate less intervals in the working partition. This effect can be observed independently of the size of the current lifting dimension M. However, one limitation of the algorithm becomes apparent if we increase the length of the time horizon of the maneuver. If the time horizon is prolonged to $T = 10$ (instead of $T = 2$) the Branch & Lift algorithm takes up to 20min. If the horizon is made even longer, the partition sometimes becomes too big, while dynamic programming methods scale linearly with respect to the duration of the maneuver. This current issue with Branch & Lift is mostly due to the fact that the conservatism of many state-of-the-art set propagation integrators [25] increases on longer time horizons due to wrapping effects. As one might expect, the quality of the outer approximation of the reachable sets has a major influence on the performance of the proposed algorithm. Tailored intergators for moment-constrained reachable sets for problems with longer horizon are under investigation [19].

VI. Conclusion

This paper has discussed the application of Branch & Lift algorithms for solving obstacle avoidance control problems to global optimality. A theoretical convergence guarantee and an associated complexity estimate for this algorithm for general non-convex optimal controls problems has been summarized in Theorem 1. We have illustrated the main algorithmic steps by applying Branch & Lift to a tutorial obstacle avoidance problem based on Dubin's model. Numerical experiments for problems with more than one obstacle indicate superior performance of Branch & Lift over other methods such as single-shooting with Branch & Bound or dynamic programming, although one current bottleneck of the implementation of Branch & Lift, is that the method scales less favorably with respect to the time horizon duration. This problem will be overcome in future implementations by developing tailored moment-constrained reach-set integrators.

References

[1] D. Bampou and D. Kuhn. Polynomial approximations for continuous linear programs. *SIAM Journal on Optimization*, 22(2):628–648, 2012.

[2] D.P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

[3] J. Borenstein and Y. Koren. The vector field histogram—fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, 1991.

[4] B. Chachuat, B. Houska, R. Paulen, N. Perić, J. Rajyaguru, and M.E. Villanueva. Set-theoretic approaches in analysis, estimation and control of nonlinear systems. *IFAC-PapersOnLine*, 48(8):981–995, 2015.

[5] B. Chachuat, A.B. Singer, and P.I. Barton. Global methods for dynamic optimization and mixed-integer dynamic optimization. *Industrial & Engineering Chemistry Research*, 45(25):8373–8392, 2006.

[6] L. Chen and C. Englund. Cooperative intersection management: a survey. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):570–586, 2016.

[7] F.L. Chernousko. Optimization in control of robots. *International Series of Numerical Mathematics*, page 19, 1994.

[8] O. Devolder, F. Glineur, and Y. Nesterov. Solving infinite-dimensional optimization problems by polynomial approximation. In *Recent Advances in Optimization and its Applications in Engineering*, pages 31–40. Springer, 2010.

[9] H. Diedam and S. Sager. Global optimal control with the direct multiple shooting method. *Optimal Control Applications and Methods*, DOI. 10.1002/oca.2324, 2017.

[10] W.R. Esposito and C.A. Floudas. Deterministic global optimization in nonlinear optimal control problems. *Journal of Global Optimization*, 17(1):97–126, 2000.

[11] M. Gerdts, R. Henrion, D. Hömberg, and C. Landry. Path planning and collision avoidance for robots. *Numerical Algebra, Control and Optimization*, 2(3):437–463, 2012.

[12] M. Gerdts and I. Xausa. Avoidance trajectories using reachable sets and parametric sensitivity analysis. In *System Modelling and Optimization*, pages 491–500. Springer, 2011.

[13] E.G. Gilbert and D.W. Johnson. Distance functions and their application to robot path planning in the presence of obstacles. *IEEE Journal on Robotics and Automation*, 1(1):21–30, 1985.

[14] L. Grüne and W. Semmler. Using dynamic programming with adaptive grid scheme to solve nonlinear dynamic models in economics. Technical report, Society for Computational Economics, 2002.

[15] R. Horst and H. Tuy. *Global optimization: Deterministic approaches*. Springer Science & Business Media, 2013.

[16] B. Houska and B. Chachuat. Branch-and-lift algorithm for deterministic global optimization in nonlinear optimal control. *Journal of Optimization Theory and Applications*, 162(1):208–248, 2014.

[17] B. Houska and B. Chachuat. Global optimization in Hilbert space. *Mathematical Programming. (in revision) Available from: http://www.optimization-online.org/DB_HTML/2016/08/5581.html. Accessed February*, 6, 2017.

[18] B. Houska, H.J. Ferreau, and M. Diehl. Acado Toolkit—An open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.

[19] B. Houska, M.E. Villanueva, and B. Chachuat. Stable set-valued integration of nonlinear dynamic systems using affine set-parameterizations. *SIAM Journal on Numerical Analysis*, 53(5):2307–2328, 2015.

[20] R. Hult, M. Zanon, S. Gros, and P. Falcone. Primal decomposition of the optimal coordination of vehicles at traffic intersections. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 2567–2573. IEEE, 2016.

[21] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

[22] A.D. Luca and F. Flacco. Integrated control for phri: Collision avoidance, detection, reaction and collaboration. In *4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 288–295. IEEE, 2012.

[23] J. Rajyaguru, M.E. Villanueva, B. Houska, and B. Chachuat. Chebyshev model arithmetic for factorable functions. *Journal of Global Optimization*, 68(2):413–438, 2017.

[24] T. Schouwenaars, B. DeMoor, E. Feron, and J. How. Mixed integer programming for safe multi-vehicle cooperative path planning. In *Proceedings of the European Control Conference*, pages 2603–2608, 2001.

[25] M.E. Villanueva, B. Houska, and B. Chachuat. Unified framework for the propagation of continuous-time enclosures for parametric nonlinear ODEs. *Journal of Global Optimization*, 62(3):575–613, 2015.