Systems & Control Letters 86 (2015) 34–40

Contents lists available at ScienceDirect

Systems & Control Letters

journal homepage: www.elsevier.com/locate/sysconle

A perspective-based convex relaxation for switched-affine optimal control

Nicholas Moehle [a,*], Stephen Boyd [b]

[a] Mechanical Engineering Department, Stanford University, United States
[b] Electrical Engineering Department, Stanford University, United States

ARTICLE INFO

Article history:
Received 8 December 2014
Received in revised form
13 May 2015
Accepted 2 September 2015

Keywords:
Switched-affine systems
Optimal control
Mixed-integer convex programming
Hybrid systems
Disjunctive programming

ABSTRACT

We consider the switched-affine optimal control problem, *i.e.*, the problem of selecting a sequence of affine dynamics from a finite set in order to minimize a sum of convex functions of the system state. We develop a new reduction of this problem to a mixed-integer convex program (MICP), based on perspective functions. Relaxing the integer constraints of this MICP results in a convex optimization problem, whose optimal value is a lower bound on the original problem value. We show that this bound is at least as tight as similar bounds obtained from two other well-known MICP reductions (via conversion to a mixed logical dynamical system, and by generalized disjunctive programming), and our numerical study indicates it is often substantially tighter. Using simple integer-rounding techniques, we can also use our formulation to obtain an upper bound (and corresponding sequence of control inputs). In our numerical study, this bound was typically within a few percent of the optimal value, making it attractive as a stand-alone heuristic, or as a subroutine in a global algorithm such as branch and bound. We conclude with some extensions of our formulation to problems with switching costs and piecewise affine dynamics.

1. Switched-affine control

A *switched-affine system* has the form

$$x_{t+1} = A^{u_t} x_t + b^{u_t}, \quad t = 0, 1, \dots,$$

where $x_t \in \mathbf{R}^n$ is the state at time t, $u_t \in \{1, \dots, K\}$ is the control input at time t, and $A^1, \dots, A^K$ and $b^1, \dots, b^K$ are given matrices and vectors. At each time period, the control input selects from a given finite set of affine dynamics. We assume, without loss of generality, that $(A^i, b^i) \neq (A^j, b^j)$ for $i \neq j$. Switched-affine systems arise in various engineering applications, for example as models of switched-mode power supplies and power conversion circuits.

The switched-affine control problem is

$$\begin{array}{ll} \text{minimize} & \sum_{t=0}^{T} g_t(x_t) \\ \text{subject to} & x_{t+1} = A^{u_t} x_t + b^{u_t} \\ & u_t \in \{1, \dots, K\}, \end{array} \qquad (1)$$

where the constraints must hold for $t = 0, \dots, T-1$. The problem variables are the system states $x_0, \dots, x_T \in \mathbf{R}^n$ and the control inputs $u_0, \dots, u_{T-1}$. The problem parameters are the dynamics (A^i, b^i) for $i = 1, \dots, K$ and the stage cost functions $g_0, \dots, g_T$. We assume the stage cost functions $g_t : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ are convex and extended valued, which allows us to represent convex state constraints in the stage cost function. We define the state constraint set as $\mathcal{X}_t = \{x \mid g_t(x) < \infty\}$, so the objective is infinite unless $x_t \in \mathcal{X}_t$ holds for $t = 0, \dots, T$. We can use g_0 to encode a given initial condition, so that $\mathcal{X}_0 = \{x_{\text{init}}\}$, for some $x_{\text{init}} \in \mathbf{R}^n$.

The switched-affine control problem (1) is NP-hard in general (this is proven by Egerstedt and Blondel [1] for a special case), and can be solved globally only at great computational cost in the worst-case. However, by reformulating it as a mixed-integer convex program (MICP), lower bounds on the optimal value can be obtained by relaxing the integer constraints, and upper bounds can be obtained by applying an integer-rounding heuristic to the relaxed solution. These bounds can be used as the basis for a global solver (using, *e.g.*, branch and bound), or alternatively, the rounding procedure can be used as a heuristic to produce a good, if not optimal, sequence of control inputs. The success of both methods (*i.e.*, the run-time of a global solution algorithm, or the quality of the heuristic control sequence) depends crucially on the MICP reformulation (and the tightness of the bounds it produces).

* Corresponding author.
E-mail addresses: moehle@stanford.edu (N. Moehle), boyd@stanford.edu (S. Boyd).

http://dx.doi.org/10.1016/j.sysconle.2015.09.002

In this paper, we give a new MICP formulation that achieves better bounds than those obtained from other popular reformulation techniques. Although we focus on the specific problem given in (1), we give some extensions of our approach to some related problems in Section 6.

1.1. Previous work

1.1.1. Switched-affine control

Many approaches exist for optimal control of switched systems; a summary can be found in [2]. Here we mention some particularly relevant techniques.

Mixed logical dynamical systems. Switched-affine systems are a special case of *hybrid systems*, *i.e.*, systems involving continuous and logical dynamics. A standard approach to solve (1), proposed by Torrisi and Bemporad [3], is to first convert the switched-affine system into an equivalent *mixed logical dynamical* (MLD) system, which expresses the system using a combination of linear and binary constraints on the original variables and some auxiliary variables (see [4] for details on MLD systems). Minimizing a sum of convex functions of the system states can therefore be expressed as an MICP. We will call this the MLD approach to solving (1), and will briefly describe it in Section 4.1.

Disjunctive programming. Problem (1) can be cast as a *disjunctive program*, *i.e.*, an optimization problem in which the decision variables must lie in the union of some sets (see [5]). Ceria and Soares [6] show that minimizing a convex function over the union of convex sets can be equivalently formulated as an MICP, using lifted variables and perspective functions. This technique has seen much application in process engineering (see, *e.g.*, [7]); for some other applications, see [8]. Several works apply disjunctive programming to switched-affine optimal control; the first appears to be by Stursberg and Panek [9]; we refer to this approach as the GDP formulation, and we describe it in Section 4.2. Oldenburg and Marquardt [10,11] give a detailed account of how to formulate complex switched dynamic constraints using a disjunctive programming framework. Disjunctive programming techniques have also been suggested for deriving mixed logical dynamical systems; see [12]. Several strategies for finding an upper bounds, some with guaranteed suboptimality bounds, can be found in the work of Sager, Jung, and Reinelt [13,14].

Approximate dynamic programming. Wang, O'Donoghue, and Boyd [15] give a method for obtaining relaxations for several hard optimal control problems, including switched-affine systems. The bounds are obtained by maximizing a quadratic approximate value function, evaluated over some initial state distribution, while constraining it to be an under-estimator of the true value function (using a chain of Bellman inequalities).

1.2. Convex optimization

Convex optimization problems can be solved efficiently and reliably using standard techniques [16, Ch. 1]. In practice, this is often done by representing the functions involved in terms of a few standard convex cones, then using a conic optimization solver. Typical cones used in convex optimization include the positive orthant, second-order cone, semidefinite cone, exponential cone, and combinations thereof. Many functions and constraints are representable in terms of these cones; several examples are given in [17–20].

Mixed-integer optimization problems that are convex if the integrality constraints are relaxed are called *mixed-integer convex programs* (MICPs). Although mixed-integer convex programming is NP-hard, these problems can, in principle, be solved using simple branch-and-bound schemes; see [21] for details. Other techniques apply specifically mixed-integer linear programs (MILPs) and, more recently, mixed-integer second-order cone programs (MISOCPs); specialized solvers capable of handling MILPs and MISOCPs include the commercial solvers Mosek, Gurobi, and CPLEX, as well as ECOS-BB, an extension to the open-source, embedded second-order cone programming solver ECOS [22].

1.3. Contributions

In this paper, we give a new formulation of (1) as a mixed-integer convex program, based on perspective functions. We can then obtain a lower bound on (1) by relaxing the integer constraints and solving the resulting convex optimization problem. We show that this lower bound is at least as good as the lower bound obtained by relaxing the integer constraints of either the MLD or GDP formulations; our numerical study suggests that this difference can be substantial. We also show how to combine our formulation with a simple shrinking-horizon heuristic to get upper bounds on (1). Again, our numerical study suggests that this upper bound can be much tighter than the upper bound obtained using the same shrinking-horizon heuristic with the MLD or GDP formulation.

Our formulation is of course related to, and derivable from, several other approaches, although not in simple or obvious ways. Our formulation is derivable from the standard MICP reformulation procedure for (convex) disjunctive programs, as given in [6,7]. However, it differs from the "convex hull" approach followed in [9], which involves minimizing the original objective function over the convex hull of the disjunctive constraints. Instead, our formulation is obtained by first considering an epigraph formulation of (1), then treating all constraints as disjunctive constraints (even if the constraint is the same for all disjunctions); only then do we apply the convex hull relaxation.

Our lower bound can also be derived from the approach of Wang, O'Donoghue, and Boyd [15] (when modified to apply to a finite-horizon problem). In particular, if we take a chain of T Bellman inequalities, and restrict our search to value function under-estimators that are affine (instead of quadratic), then the problem of maximizing the value function under-estimator (evaluated at x_{init}) is the dual of our formulation.

1.4. Outline

In Section 2, we review some properties of perspectives of convex functions. In Section 3, we give an alternate MICP formulation based using perspective functions, and we prove its equivalence to (1). In Section 4, we review three other approaches to solving (1): by the standard conversion to a mixed logical dynamical system, by generalized disjunctive programming, and by approximate dynamic programming. We then compare these methods to our perspective-base formulation. In Section 5, we give an example with numerical results, and in Section 6, we give some extensions of our method to problems similar to (1).

2. Perspective of a function

Recall that the *perspective* of an extended-value convex function $g : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ is the function $p : \mathbf{R}^{n+1} \to \mathbf{R} \cup \{\infty\}$ defined by:

$$p(x, s) = \begin{cases} sg(x/s) & s > 0 \\ 0 & s = 0,\ x = 0 \\ \infty & \text{otherwise.} \end{cases}$$

Crucially, if g is convex, then so is p. (This can be shown by directly checking Jensen's inequality for all cases above.) For more details on perspective functions, see [16, Section 3.2.6] or [23, Section IV.2.2]. (Note that the definitions in these references differ for $s = 0$.)

Closedness. If g is closed and proper, then its perspective p is closed if and only if g grows superlinearly in all directions, starting from any point in its domain (*i.e.*, $\lim_{s\to\infty}(g(x+sz)-g(x))/s = \infty$ for all nonzero z and $x \in \mathcal{X}$). For details, see [23, Section IV.2.2] or [24, Section 8]. Examples of functions that meet this condition are positive definite quadratic functions and functions with bounded domain.

For this reason, in the sequel we assume that g_t in (1) is closed, proper, and satisfies the superlinear growth condition given above. For some functions of interest, such as norms, this condition does not hold; a practical workaround is to include a small quadratic penalty in g_t, or to restrict the domain of g_t to be bounded.

Conic representation. A *conic representation* of g consists of a matrix C, vectors d and e, and a closed cone $\mathcal{K}$ such that

$$\mathbf{epi}\, g = \{(x,\lambda) \mid g(x) \le \lambda\} = \{(x,\lambda) \mid Cx + \lambda d + e \in \mathcal{K}\}, \qquad (2)$$

where **epi** g is the epigraph of g. Conic representations allow us to express nonsmooth functions (*e.g.*, perspective functions) in a smooth form, so they can be used in standard conic optimization software. Given a conic representation (2) of g, then if p is closed, a conic representation of p is

$$\begin{aligned}\mathbf{epi}\, p &= \{(x,s,\lambda) \mid p(x,s) \le \lambda\}\\ &= \{(x,s,\lambda) \mid Cx + \lambda d + se \in \mathcal{K}\}.\end{aligned} \qquad (3)$$

This fact has important practical consequences for solving convex optimization problems involving perspective functions, such as the perspective formulation of (1) given below. In particular, by using a smooth, conic representation of the perspective, we sidestep troublesome nondifferentiability and division-by-zero issues that could arise by attempting to directly implement perspective functions numerically.

3. Perspective formulation

The *perspective formulation* of (1) is the following MICP:

$$\begin{array}{ll} \text{minimize} & g_T(x_T) + \sum_{t=0}^{T-1}\sum_{i=1}^{K} p_t(z_t^i, s_t^i) \\ \text{subject to} & x_{t+1} = \sum_{i=1}^{K} A^i z_t^i + b^i s_t^i \\ & x_t = \sum_{i=1}^{K} z_t^i \\ & s_t^1 + \cdots + s_t^K = 1 \\ & s_t^i \in \{0,1\}, \end{array} \qquad (4)$$

where all constraints must hold for $t = 0,\dots,T-1$, and the last constraint also holds for $i = 1,\dots,K$. In addition to x_t, the variables are $z_t^i \in \mathbf{R}^n$ and s_t^i, for $t = 0,\dots,T-1$ and $i = 1,\dots,K$. The function p_t is the perspective of g_t.

Problem (4) can be solved using an MICP solver, which may require transforming the objective and constraints to conic form. This can be done by hand, or by modeling software such as CVX [25,26].

Proof of equivalence. To see the equivalence of (1) and (4), take any x_t, z_t^i and s_t^i (for appropriate t and i) that are feasible for (4) (*i.e.*, they satisfy all constraints, and the objective value is finite). For each t from 0 to $T-1$, we have $s_t^i = 1$ for exactly one value of i; denote this value as u_t. Because the objective is finite, and because $s_t^i = 0$ for $i \ne u_t$, we must also have $z_t^i = 0$ for $i \ne u_t$. The value of the objective is then

$$\begin{aligned} g_T(x_T) + \sum_{t=0}^{T-1}\sum_{i=1}^{K} p_t(z_t^i, s_t^i) &= g_T(x_T) + \sum_{t=0}^{T-1} p_t(x_t, 1) \\ &= \sum_{t=0}^{T} g_t(x_t), \end{aligned} \qquad (5)$$

and the first constraint implies

$$x_{t+1} = A^{u_t}x_t + b^{u_t}, \quad \text{for } t = 0,\dots,T-1.$$

Then x_t and u_t are a feasible point for (1) with the same objective value as our feasible point for (4).

Similarly, if x_t and u_t are feasible for (1), we define

$$(z_t^i, s_t^i) = \begin{cases} (x_t, 1) & u_t = i \\ (0,0) & \text{otherwise,} \end{cases}$$

for $t = 0,\dots,T-1$. It is easy to check that x_t, z_t^i, and s_t^i (for appropriate t and i) satisfy all constraints for (4). We can then apply (5) to show that the objective values of (1) and (4) agree.

Bounds and approximate solutions. To obtain a lower bound on the optimal value of (4), we can relax the integer constraints $s_t^i \in \{0,1\}$ to $s_t^i \in [0,1]$. This problem is convex (hence easily solved) and its optimal value is a lower bound on the optimal value of (1). We call this problem the *perspective relaxation* of (1).

An upper bound for (1) can be found by choosing an initial condition x_0 and a sequence of switching controls $u_0,\dots,u_{T-1}$, simulating the dynamical system according to $x_{t+1} = A^{u_t}x_t + b^{u_t}$, and evaluating the objective function $\sum_{t=0}^{T} g_t(x_t)$. The relax-and-round method for choosing a sequence of switching controls starts from a solution $\tilde{x}_t$, $\tilde{z}_t^i$, $\tilde{s}_t^i$ of the integer relaxation of (4). We then take $x_0 = \tilde{x}_0$ and $u_t \in \operatorname{argmax}_{i\in\{1,\dots,K\}} \tilde{s}_t^i$. A more sophisticated (and typically much better) upper bound can be found by taking $u_0 \in \operatorname{argmax}_{i\in\{1,\dots,K\}} \tilde{s}_0^i$, as in the relax-and-round method. We then compute $x_1 = A^{u_0}x_0 + b^{u_0}$, and repeat the procedure, solving a new relaxed problem with initial state x_1 and horizon length $T-1$. This technique requires solving T convex optimization problems of decreasing size. We refer to this as the *shrinking-horizon bound* (as in [27]). It often produces a good, if not optimal, choice of switching controls, as well as an upper bound on the optimal value of the switching control problem. (Of course, these heuristics may also fail to find a feasible point, even if one exists.)

4. Comparison to other formulations

In this section we compare the perspective formulation to three other solution techniques for (1) from the literature.

4.1. Mixed logical dynamical formulation

A standard approach to solve (1) is by optimizing over an equivalent MLD system, as described in [28]. (For simplicity, we refer to this as the MLD formulation, although other methods for converting switched-affine systems to MLD systems are possible.) Here we make the assumption that the dynamics update expressions are bounded over $\mathcal{X}_t$, *i.e.*,

$$m_t^i \le A^i x_t + b^i \le M_t^i, \quad \text{for all } x_t \in \mathcal{X}_t$$

for some known vectors $m_t^i, M_t^i \in \mathbf{R}^n$ (the inequalities are taken to be elementwise). (When this assumption does not hold, standard practice is to take m_t^i to be sufficiently small, and M_t^i sufficiently large, so that they can reasonably be expected not to affect the problem solution. This is often called a *big-M method*.) Under this assumption, the *MLD formulation* of (1) is:

$$\begin{array}{ll} \text{minimize} & \sum_{t=0}^{T} g_t(x_t) \\ \text{subject to} & m_t^i s_t^i \le y_t^i \le M_t^i s_t^i \\ & A^i x_t + b^i - M_t^i(1-s_t^i) \le y_t^i \\ & \le A^i x_t + b^i - m_t^i(1-s_t^i) \\ & x_{t+1} = \sum_{i=1}^{K} y_t^i \\ & s_t^1 + \cdots + s_t^K = 1 \\ & s_t^i \in \{0,1\}, \end{array} \tag{6}$$

where all constraints must hold for $t = 0, \ldots, T-1$ and $i = 1, \ldots, K$. In addition to x_t, the variables include $y_t^i \in \mathbf{R}^n$ and s_t^i, for $t = 0, \ldots, T-1$ and $i = 1, \ldots, K$. Similar to (4), this problem can be solved using an MICP solver.

The same procedures used to bound the optimal value of (4) can be used to bound the optimal value of (6), or produce an approximate solution. To obtain a lower bound, we again relax the constraints $s_t^i \in \{0,1\}$ to $s_t^i \in [0,1]$; we call this the *MLD relaxation* of (1). The same relax-and-round and shrinking-horizon methods can be used to find an approximately optimal choice of switching controls, and therefore also an upper bound on the optimal value of (1).

4.1.1. Comparison of lower bounds

Here we prove that the lower bound obtained from the integer relaxation (4) is at least as tight as the bound from the integer relaxation of (6). To do this, we will show that, given an arbitrary feasible point for the relaxation of (4), we can construct a feasible point for the relaxation of (6) with lower objective value. We only treat the case in which the assumption of Section 4.1 holds; otherwise, the MLD method cannot be used.

Constraint satisfaction. Suppose x_t, z_t^i, and s_t^i (for appropriate values of t and i) are a feasible point for integer relaxation of (4). From the definition of perspective, for p_t to be finite (and thus for our point to be feasible), we must have $z_t^j/s_t^j \in \mathcal{X}_t$ if $s_t^j > 0$, and $z_t^j = 0$ if $s_t^j = 0$. Combining this with the assumption of Section 4.1, we have

$$m^i s_t^j \le A^i z_t^j + b^i s_t^j \le M^i s_t^j \tag{7}$$

for all i and j, and all $t = 0, \ldots, T-1$.

Now we show that by defining $y_t^i = A^i z_t^i + b^i s_t^i$, we have that x_t, y_t^i, and s_t^i satisfy all (non-integrality) constraints of (6). The first constraint is obtained by applying (7) with $j = i$, and noting that the middle term is equal to y_t^i. We now consider the second constraint. By summing (7) over all $j \ne i$, and using $\sum_{i=1}^K s_t^i = 1$ and $\sum_{i=1}^K z_t^i = x_t$ we have

$$m^i(1-s_t^i) \le A^i(x_t - z_t^i) + b^i(1-s_t^i) \le M_i(1-s_t^i).$$

Rearranging these inequalities yields

$$A^i x_t + b^i - M^i(1-s_t^i) \le A^i z_t^i + b^i s_t^i \le A^i x_t + b^i - m^i(1-s_t^i),$$

and substituting y_t^i for $A^i z_t^i + b^i s_t^i$ gives the desired result. The third constraint of (6) is obtained by substituting $A^i z_t^i + b^i s_t^i$ for y_t^i, and noting equivalence with the first constraint of (4). Finally, the fourth constraint is equivalent to the fourth constraint of (4).

Objective bound. We now show the objective value of the new point for the integer relaxation of (6) is lower than that of the original point for (4). The objective of (6) is

$$\sum_{t=0}^{T} g_t(x_t) = g_T(x_T) + \sum_{t=0}^{T-1} g_t\left(\sum_{i\in\mathcal{I}_t} s_t^i(z_t^i/s_t^i)\right),$$

where $\mathcal{I}_t = \{i \mid s_t^i \ne 0\}$. We are justified in replacing x_t with this sum because $x_t = \sum_{i=1}^K z_t^i$, and z_t^i vanishes if s_t^i does. Using Jensen's inequality for g_t, the right-hand side is bounded above by

$$g_T(x_T) + \sum_{t=0}^{T-1} \sum_{i\in\mathcal{I}_t} s_t^i g_t(z_t^i/s_t^i).$$

Because $p_t(z_t^i, s_t^i) = 0$ for $s_t^i = 0$ and $z_t^i = 0$, this is in fact equal to the objective of (4).

4.2. Generalized disjunctive programming formulation

Here we introduce the *generalized disjunctive programming* (GDP) formulation, which first appeared in [9]. (This name common in the literature, but is somewhat unfortunate in this context, as our approach can also be derived from disjunctive programming techniques.)

Define the perspective $\mathcal{P}$ of a set $\mathcal{X}$ as

$$\mathcal{P} = \{(x,s) \mid s > 0,\ x/s \in \mathcal{X}\} \cup \{(0,0)\}.$$

The perspective of a convex set is convex. (To see this, take g to be the indicator function over $\mathcal{X}$, and note that the perspective of g is the indicator function of the perspective of $\mathcal{X}$.)

The GDP formulation of (1) is

$$\begin{array}{ll} \text{minimize} & \sum_{t=0}^{T} g_t(x_t) \\ \text{subject to} & x_{t+1} = \sum_{i=1}^{K} A^i z_t^i + b^i s_t^i \\ & x_t = \sum_{i=1}^{K} z_t^i \\ & s_t^1 + \cdots + s_t^K = 1 \\ & (z_t^i, s_t^i) \in \mathcal{P}_t \\ & s_t^i \in \{0,1\}. \end{array} \tag{8}$$

The variables and constraints are the same of those of (4), except the added constraint $(z_t^i, s_t^i) \in \mathcal{P}_t$, which holds for $i = 1, \ldots, K$ and $t = 1, \ldots, T-1$. The sets $\mathcal{P}_t$ are the perspectives of the sets $\mathcal{X}_t$. For $\mathcal{X}_t$ bounded, this problem is equivalent to (1).

As with the previous formulations, a lower bound on (1) can be obtained by relaxing the integer constraints of (8) and solving the resulting convex optimization problem, which we call the *GDP relaxation*. Note that the objectives of the GDP and MLD formulations are the same, as are the feasible sets of the GDP and perspective formulations, making the GDP formulation a hybrid between the two formulations. In fact, by following the same arguments given in Section 4.1.1, any feasible point for the MLD relaxation can be used to generate a feasible point for the GDP relaxation with the same objective value. Similarly, any feasible point for the GDP relaxation is also a feasible point for the perspective relaxation, and attains a greater or equal objective value. This establishes a hierarchy of relaxations: the MLD relaxation is weaker than the GDP relaxation, which is in turn weaker than the perspective relaxation.

4.3. Approximate dynamic programming

The dual of (4) can be written as

$$\begin{array}{ll} \text{maximize} & \mu_0 \\ \text{subject to} & \lambda_t^T x + \mu_t \le g_t(x) + \lambda_{t+1}^T(A^k x + b^k) + \mu_{t+1} \end{array} \tag{9}$$

with variables λ_t for $t = 1, \ldots, T$ and μ_t for $t = 0, \ldots, T$. The constraint holds for $i = 1, \ldots, K$, $t = 0, \ldots, T-1$ and all $x \in \mathbf{R}^n$. (We take λ_0, λ_{T+1}, and μ_{T+1} to be zero, for notational convenience.)

This problem can be interpreted as an approximate dynamic programming (ADP) method. Recall that if a (time-dependent) approximate value function $\hat{V}_t$ satisfies $\hat{V}_T(x) \le g_T(x)$, as well as the chain of Bellman inequalities

$$\hat{V}_t(x) \le \inf_k g_t(x) + \hat{V}_{t+1}(A^k x + b^k)$$

for all x and $t = 0, \ldots, T-1$, then we have $\hat{V}_t(x) \le V_t(x)$ for all x and t, where V_t is the optimal value function for (1). Problem (9) can therefore be interpreted as the problem of finding the affine value function underestimator $\hat{V}_t(x) = \lambda_t^T x + \mu_t$ (with $\lambda_0 = 0$) that achieves the highest cost at time $t = 0$. This approach to ADP is related to the linear programming solution of finite Markov decision problems given by de Farias and Van Roy [29] and the semidefinite programming approaches to ADP given by Rantzer [30], and Wang, O'Donoghue, and Boyd [15], which are applicable to switched-affine dynamics and quadratically representable costs.

5. Example

In this section we give numerical results for a specific example, linear–quadratic switching control, with stage cost function

$$g_t(x) = \begin{cases} x^T Q x & \|x\|_\infty \le x_{\max} \\ \infty & \text{otherwise,} \end{cases}$$

for $t = 1, \ldots, T$, where $Q \in \mathbf{S}^n_{++}$. The perspective of g_t is

$$p_t(z, s) = \begin{cases} z^T Q z / s & \|z\|_\infty \le x_{\max} s, \ s > 0 \\ 0 & z = 0, \ s = 0 \\ \infty & \text{otherwise.} \end{cases}$$

The function g_0 is used to encode an initial condition, so that

$$g_0(x) = \begin{cases} 0 & x = x_{\text{init}} \\ \infty & \text{otherwise.} \end{cases}$$

The perspective of g_0 is

$$p_0(z, s) = \begin{cases} 0 & z = s x_{\text{init}}, \ s \ge 0 \\ \infty & \text{otherwise.} \end{cases}$$

In this case (4) is a mixed-integer second-order cone program, and can be solved using several available solvers, such as Gurobi, Mosek [31], and ECOS-BB (an extension of ECOS [22]),

Tightness of bounds. To test the tightness of the various bounds, we generated 200 random instances of the linear–quadratic switching control problem, with state dimension $n = 3$, $K = 5$ different switched dynamics, horizon length $T = 20$, stage cost matrix $Q = I$, and state bound $x_{\max} = 5$. (We chose relatively small problems so they could be solved globally in reasonable time. All relaxations, however, scale to problems with far larger dimensions.) The dynamics matrices were randomly chosen as $A^i = I + 0.1\tilde{A}^i$ and $b^i = 0.1\tilde{b}^i$, with the elements of $\tilde{A}^i$, $\tilde{b}^i$, and x_{init} sampled from a standard normal distribution.

The values of m_t^i and M_t^i (used in the MLD method) were chosen to give the tightest bounds on the dynamics functions over the set $\mathcal{X}_t$ (*i.e.*, we took $m_0^i = M_0^i = A^i x_{\text{init}} + b^i$ and $m_t^i = b^i - x^{\max} a^i$, $M_t^i = b^i + x^{\max} a^i$, for $t = 1, \ldots, T$, where each element of the vector a^i is the ℓ_1-norm of the corresponding row of A^i).

For each random instance we first computed the optimal value by solving the mixed integer problem (4) globally, using CVX [25,26], with Gurobi as the solver. All 200 instances were feasible. We then computed six bounds on the optimal value: three lower bounds from the perspective relaxation, the MLD relaxation, and the GDP relaxation (*i.e.*, the integer relaxations of (6), (4), and (8)) and three upper bounds from shrinking-horizon heuristic based on the three formulations (6), (4), and (8). For each problem instance, the bounds were scaled by the optimal value, so that the lower bounds are between 0 and 1, and the upper bounds are greater than or equal to 1.

Table 1
The mean and median of the ratio of each bound to the optimal value, as well as the percentage of instances for which the bound is infinite, for the randomly generated instances of the linear–quadratic switching control example.

Bound	Mean	Median	% inf.
Shrinking-horizon, perspective	1.11	1.04	0%
Shrinking-horizon, MLD	∞	1.14	1%
Shrinking-horizon, GDP	1.38	1.13	0%
Relaxation, perspective	0.76	0.79	–
Relaxation, MLD	0.19	0.17	–
Relaxation, GDP	0.20	0.19	–

The means and medians of the four (scaled) bounds across the 200 instances are shown in Table 1. The shrinking horizon heuristic based on the MLD method did not always find a feasible point, even though all problem instances were feasible, so the mean is infinite; we also show in Table 1 the fraction of instances for which each upper bound is infinite. Note that the median of the shrinking-horizon bound using the perspective formulation is 1.04, meaning that in the majority of instances, this heuristic produced a bound within four percent of the optimal value. The histograms of all six bounds are shown in Fig. 1.

Solve time. We also performed a simple comparison of the time required to globally solve the perspective formulation (4), the MLD formulation (6), the GDP formulation (8). We used the same numerical parameter values that we used for comparing the tightness of the bounds; however, due to the difficulty of solving (6) and (8) globally, we consider only the first 50 of the 200 instances, and we terminated the solver if an instance took longer than five hours. We used CVX with Gurobi, running on a Linux machine with an Intel Xeon processor. Using the perspective formulation, the mean solve time for was around 7 min, and the maximum solve time was 1 h. Using the GDP formulation, the mean solve time was (at least) 52 min, with six instances terminated after five hours. Using the MLD formulation, the mean solve time was (at least) 1.5 h, with 12 of the instances terminated after five hours.

6. Extensions

We conclude with some extensions of the perspective-based reformulation for problems that are not in the form of problem (1).

Switching costs. To incorporate switching costs, we add $\sum_{t=0}^{T-1} h_t(u_{t-1}, u_t)$ to the objective of (1), where $h_t(i, j) > 0$ is the cost of transitioning from dynamics i to dynamics j. We take $u_{-1} \in \{1, \ldots, K\}$ to be a known parameter. (Intuitively, u_{-1} gives the dynamics applied just before our problem begins.) An equivalent MICP is obtained by adding

$$\sum_{i=1}^{K} h_0(u_{-1}, s_0) + \sum_{t=1}^{T-1} \sum_{i=1}^{K} h_t(s_{t-1}^i, s_t^i)$$

to the objective of (4).

Switch-dependent stage costs. In some applications, the stage cost function may depend on u_t, so that the objective of (1) becomes

$$g_T(x_T) + \sum_{t=0}^{T-1} g_t^{u_t}(x_t),$$

where $g_t^{u_t}(x_t)$ is convex in x_t for each value of u_t. An equivalent problem is formed from (4), with the objective replaced by

$$g_T(x_T) + \sum_{t=0}^{T-1} \sum_{i=1}^{K} p_t^i(z_t^i, s_t^i),$$

where p_t^i is the perspective of g_t^i.

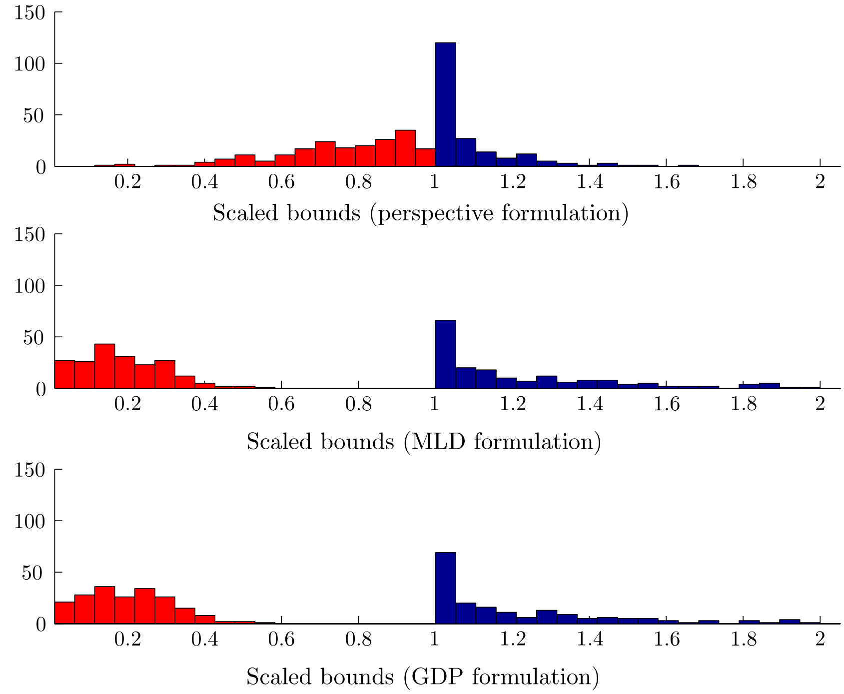

Fig. 1. Histograms of the scaled upper and lower bounds (a bound of 1 is the tightest bound possible). The top shows the lower bounds from integer relaxation (red) and upper bounds from the shrinking-horizon heuristic (blue), both obtained using the MLD formulation (6). The bottom shows the same bounds obtained using the perspective reformulation (4). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Piecewise affine systems. A *piecewise affine system* has the form

$$x_{t+1} = \begin{cases} A^1 x_t + Bv_t & f^1(x_t) \le 0 \\ \vdots & \vdots \\ A^K x_t + Bv_t & f^K(x_t) \le 0 \end{cases}$$

for $t = 0, 1, \ldots$. The (continuous) control input is $v_t \in \mathbf{R}^m$. We assume the functions f^i are convex, and also that the sublevel sets $\{x \mid f^i(x) \le 0\}$ for $i = 1, \ldots, K$ have disjoint interior, that their union is $\mathbf{R}^n$, and that on the intersection of any two of these sets, the piecewise dynamics agree.

Using switch-dependent constraints (and adding a continuous input), we can minimize $g_T(x_T) + \sum_{t=0}^{T-1} \big(g_t(x_t) + l_t(v_t)\big)$, where l_t is convex, over a piecewise affine system, by solving

$$\begin{array}{ll} \text{minimize} & g_T(x_T) + \sum_{t=0}^{T-1} \Big(l_t(v_t) + \sum_{i=1}^{K} p_t(z_t^i, s_t^i) \Big) \\ \text{subject to} & x_{t+1} = \Big(\sum_{i=1}^{K} A^i z_t^i \Big) + Bv_t \\ & x_t = \sum_{i=1}^{K} z_t^i \\ & s_t^1 + \cdots + s_t^K = 1 \\ & s_t^i \in \{0, 1\} \\ & q^i(z_t^i, s_t^i) \le 0, \end{array} \qquad (10)$$

where q^i is the perspective of f^i. The first three constraints must hold for $t = 0, \ldots, T-1$, and the last two constraints must hold for $t = 0, \ldots, T-1$ and $i = 1, \ldots, K$. The variables are the same as those of (4), with the addition of $v_t \in \mathbf{R}^m$ for $t = 0, \ldots, T-1$.

7. Conclusion

In this paper, we presented a formulation of the switched-affine optimal control problem as an MICP, allowing us to obtain bounds on the optimal problem value using convex optimization, and to use standard MICP solvers to solve the problem. We compared our MICP formulation to some other popular reformulation techniques, and showed that our formulation provides very competitive bounds, both theoretically and numerically.

We conclude by noting that unlike the MLD and GDP formulations, the perspective formulation crucially depends on reformulating the objective of (1) in addition to the feasible set, even though the objective is already a convex function. We believe that this principle could be fruitfully applied to other types of hybrid optimal control problems and this may be the subject of future research.

Acknowledgments

This research was supported by the DARPA XDATA program. We thank the reviewers, as well as Alberto Bemporad, for helpful comments and suggestions.

References

[1] M. Egerstedt, V.D. Blondel, How hard is it to control switched systems? in: Proceedings of the 2002 American Control Conference, Vol. 3, IEEE, 2002, pp. 1869–1873.
[2] S. Sager, Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control, J. Process Control 19 (8) (2009) 1238–1247.
[3] F.D. Torrisi, A. Bemporad, HYSDEL—a tool for generating computational hybrid models for analysis and synthesis problems, IEEE Trans. Control Syst. Technol. 12 (2) (2004) 235–249.
[4] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, Automatica 35 (3) (1999) 407–427.
[5] E. Balas, Disjunctive programming, Ann. Discrete Math. 5 (1979) 3–51.
[6] S. Ceria, J. Soares, Convex programming for disjunctive convex optimization, Math. Program. 86 (3) (1999) 595–614.
[7] I.E. Grossmann, F. Trespalacios, Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming, Am. Inst. Chem. Eng. J. 59 (9) (2013) 3276–3295.
[8] O. Günlük, J. Linderoth, Perspective reformulation and applications, in: Mixed Integer Nonlinear Programming, Springer, 2012, pp. 61–89.
[9] O. Stursberg, S. Panek, Control of switched hybrid systems based on disjunctive formulations, in: Proceedings of 5th International Workshop on Hybrid Systems: Computation and Control, Springer, 2002, pp. 421–435.

[10] J. Oldenburg, W. Marquardt, Optimization of discrete–continuous dynamic systems based on disjunctive programming, Proc. Appl. Math. Mech. 5 (1) (2005) 51–54.
[11] J. Oldenburg, W. Marquardt, Disjunctive modeling for optimal control of hybrid systems, Comput. Chem. Eng. 32 (10) (2008) 2346–2364.
[12] A. Bemporad, N. Giorgetti, Logic-based solution methods for optimal control of hybrid systems, IEEE Trans. Automat. Control 51 (6) (2006) 963–976.
[13] S. Sager, M. Jung, C. Kirches, Combinatorial integral approximation, Math. Methods Oper. Res. 73 (3) (2011) 363–380.
[14] M.N. Jung, G. Reinelt, S. Sager, The Lagrangian relaxation for the combinatorial integral approximation problem, Optim. Methods Softw. 30 (1) (2015) 54–80.
[15] Y. Wang, B. O'Donoghue, S. Boyd, Approximate dynamic programming via iterated Bellman inequalities, Internat. J. Robust Nonlinear Control (2014).
[16] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[17] A. Ben-Tal, A. Nemirovski, Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications, Vol. 2, SIAM, 2001.
[18] F. Alizadeh, D. Goldfarb, Second-order cone programming, Math. Program. 95 (1) (2003) 3–51.
[19] L. Vandenberghe, S. Boyd, Semidefinite programming, SIAM Rev. 38 (1) (1996) 49–95.
[20] M. Udell, K. Mohan, D. Zeng, J. Hong, S. Diamond, S. Boyd, Convex optimization in Julia, in: Proceedings of the First Workshop for High Performance Technical Computing in Dynamic Languages, IEEE Press, 2014, pp. 18–28.
[21] S. Boyd, J. Mattingley, Branch and bound methods, in: Notes for EE364b, Stanford University, 2007, pp. 2006–2007.
[22] A. Domahidi, E. Chu, S. Boyd, ECOS: An SOCP solver for embedded systems, in: Proceedings of the 12th European Control Conference, IEEE, 2013, pp. 3071–3076.
[23] J. Hiriart-Urruty, C. Lemaréchal, Convex Analysis and Minimization Algorithms I: Part 1: Fundamentals, Springer, 1996.
[24] R.T. Rockafellar, Convex Analysis, Princeton University Press, 1970.
[25] M. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, version 2.1, March 2014. http://cvxr.com/cvx.
[26] M. Grant, S. Boyd, Graph implementations for nonsmooth convex programs, in: V. Blondel, S. Boyd, H. Kimura (Eds.), Recent Advances in Learning and Control, in: Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, 2008, pp. 95–110.
[27] J. Skaf, S. Boyd, A. Zeevi, Shrinking-horizon dynamic programming, Internat. J. Robust Nonlinear Control 20 (17) (2010) 1993–2002.
[28] A. Bemporad, F.D. Torrisi, M. Morari, Discrete-time hybrid modeling and verification of the batch evaporator process benchmark, Eur. J. Control 7 (4) (2001) 382–399.
[29] D.P. de Farias, B. Van Roy, The linear programming approach to approximate dynamic programming, Oper. Res. 51 (6) (2003) 850–865.
[30] A. Rantzer, Relaxed dynamic programming in switching systems, IEE Proc. Control Theory Appl. 153 (5) (2006) 567–574.
[31] E.D. Andersen, K.D. Andersen, The Mosek interior point optimizer for linear programming: An implementation of the homogeneous algorithm, in: High Performance Optimization, Springer, 2000, pp. 197–232.