Safe Schedulability of Bounded-Rate Multi-Mode Systems

Rajeev Alur
University of Pennsylvania,
Philadelphia, USA
alur@seas.upenn.edu

Vojtěch Forejt
Dept. of Computer Science,
University of Oxford, UK
vojfor@cs.ox.ac.uk

Salar Moarref
University of Pennsylvania,
Philadelphia, USA
moarref@seas.upenn.edu

Ashutosh Trivedi
Indian Institute of Technology,
Bombay
trivedi@cse.iitb.ac.in

ABSTRACT

Bounded-rate multi-mode systems (BMS) are hybrid systems that can switch freely among a finite set of modes, and whose dynamics is specified by a finite number of real-valued variables with mode-dependent rates that can vary within given bounded sets. The schedulability problem for BMS is defined as an infinite-round game between two players—the scheduler and the environment—where in each round the scheduler proposes a time and a mode while the environment chooses an allowable rate for that mode, and the state of the system changes linearly in the direction of the rate vector. The goal of the scheduler is to keep the state of the system within a pre-specified safe set using a non-Zeno schedule, while the goal of the environment is the opposite. Green scheduling under uncertainty is a paradigmatic example of BMS where a winning strategy of the scheduler corresponds to a robust energy-optimal policy. We present an algorithm to decide whether the scheduler has a winning strategy from an arbitrary starting state, and give an algorithm to compute such a winning strategy, if it exists. We show that the schedulability problem for BMS is co-NP complete in general, but for two variables it is in PTIME. We also study the discrete schedulability problem where the environment has only finitely many choices of rate vectors in each mode and the scheduler can make decisions only at multiples of a given clock period, and show it to be EXPTIME-complete.

Categories and Subject Descriptors

I.2.8 [**Problem Solving, Control Methods, and Search**]: Scheduling; B.5.2 [**Design Aids**]: Verification, Optimization; D.4.7 [**Organization and Design**]: Real-time systems and embedded systems

General Terms

Theory, Verification

Keywords

Multi-Mode Systems, Hybrid Automata, Game Theory, Green Scheduling, Cyber-Physical Systems

1. INTRODUCTION

There is a growing trend towards multi-mode compositional design frameworks [8, 13, 9] for the synthesis of cyber-physical systems where the desired system is built by composing various modes, subsystems, or motion primitives—with well-understood performance characteristics—so as to satisfy certain higher level control objectives. A notable example of such an approach is *green scheduling* proposed by Nghiem et al. [11, 12] where the goal is to compose different modes of heating, ventilation, and air-conditioning (HVAC) installations in a building so as to keep the temperature surrounding each installation in a given comfort zone while keeping the peak energy consumption under a given budget. Under the assumption that the state of the system grows linearly in each mode, Nghiem et al. gave a polynomial algorithm to decide the green schedulability problem. Alur, Trivedi, and Wojtczak [1] studied general constant-rate multi-mode systems and showed, among others, that the result of Nghiem et al. holds for arbitrary multi-mode systems with constant rate dynamics as long as the scheduler can switch freely among the finite set of modes.

In this paper we present *bounded-rate multi-mode systems* that generalize constant-rate multi-mode systems by allowing non-constant mode-dependent rates that are given as bounded polytopes. Our motivations to study bounded-rate multi-mode schedulability are twofold. First, it allows one to model a conservative approximation of green schedulability problem in presence of more complex inter-mode dynamics. Second motivation is theoretical and it stems from the desire to characterize decidable problems in context of design and analysis of cyber-physical systems. In particular, we view a bounded-rate multi-mode system as a two-player extension of constant-rate multi-mode system, and show the decidability of schedulability game for such systems.

Before discussing bounded-rate multi-mode system (BMS) in any further detail, let us review the definition, relevant results, and limitations of constant-rate multi-mode system (CMS). A CMS is specified as a finite set of variables whose dynamics in a finite set of modes is given as mode-dependent constant rate vector. The *schedulability problem* for a CMS and a bounded convex safety set of states is to decide whether there exists an infinite sequence (schedule) of modes and

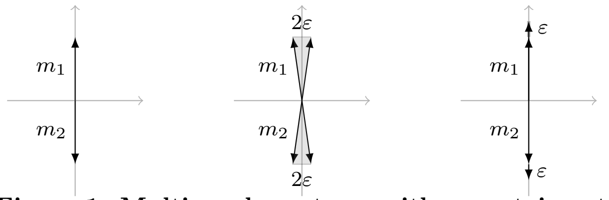


Figure 1: Multi-mode systems with uncertain rates

time durations such that choosing modes for corresponding time durations in that sequence keeps the system within the safety set forever. Moreover such schedule is also required to be physically implementable, i.e. the sum of time durations must diverge (the standard non-Zeno requirement [6]). Alur et al. [1] showed that, for the starting states in the interior of the safety set, the necessary and sufficient condition for safe schedulability is the existence of an assignment of dwell times to modes such that the sum of rate vectors of various modes scaled by corresponding dwell time is zero. Intuitively, if it is possible using the modes to loop back to the starting state, i.e. to go to some state other than the starting state and then to return to the starting state, then the same schedule can be scaled appropriately and repeated forever to form a *periodic schedule* that keeps the system inside the interior of any convex safety set while ensuring time divergence. On the other hand, if no such assignment exists then Farkas' lemma implies the existence of a vector such that choosing any mode the system makes a positive progress in the direction of that vector, and hence for any non-Zeno schedule the system will leave any bounded safety set in a finite amount of time. Also, due to constant-rate dynamics such condition can be modeled as a linear program feasibility problem, yielding a polynomial-time algorithm.

EXAMPLE 1. *Consider the 2-dimensional CMS shown in Figure 1 (left) with two modes m_1 and m_2 with rates of the variables as $\vec{r}_1 = (0,1)$ in mode m_1 and $\vec{r}_2 = (0,-1)$ in mode m_2. It is easy to see that the system is schedulable for any starting state (x_0, y_0) in the interior of any bounded convex set S as $\vec{r}_1 + \vec{r}_2 = (0,0)$. The safe schedule consists of the periodic schedule $(m_1, t), (m_2, t)$ for a carefully selected $t \in \mathbb{R}_{>0}$ such that $(x_0, y_0) + \vec{r}_1 t$ stays inside S.*

However, the schedules constructed in this manner are not robust as an arbitrarily small change in the rate can make the schedule unsafe as shown in the following example.

EXAMPLE 2. *Consider a multi-mode system where some environment related fluctuations [6] cause the rate vectors in modes m_1 and m_2 to differ from those in Example 1 by an arbitrarily small $\varepsilon > 0$ as shown in Figure 1 (middle). Here, m_1 can have rate-vectors from $\{(0+\delta, 1) : -\varepsilon \leq \delta \leq \varepsilon\}$, while rate-vectors of m_2 are from $\{(0+\delta, -1) : -\varepsilon \leq \delta \leq \varepsilon\}$. First we show that the periodic schedule $(m_1, t), (m_2, t)$ proposed in Example 1 is not safe for any t. Consider the case when the rate vector in modes m_1 and m_2 are fixed to $(\varepsilon, 1)$ and $(\varepsilon, -1)$. Starting from the state (x_0, y_0) and following the periodic schedule $(m_1, t), (m_2, t)$ for k steps the state of the system will be $(x_0 + kt\varepsilon, y_0)$ after k steps. Hence it is easy to see that for any bounded safety set the state of the system will leave the safety set after finitely many steps. In fact, for this choice of rate vectors no non-Zeno safe schedule exists at all, since by choosing any mode for a positive time the system makes a positive progress along the X axis.*

We formalize modeling of such multi-mode system under uncertainty as bounded-rate multi-mode systems (BMS). BMSs can also approximate [3] the effect of more complex non-linear, and even time-varying, mode dynamics over a bounded safety set. Formally, a BMS is specified as a finite set of variables whose dynamics in a finite set of modes is given as a mode-dependent bounded convex polytopes of rate vectors. We present the schedulability problem on BMS as an infinite-round zero-sum game between two players, *the scheduler* and *the environment*; at each round scheduler chooses a mode and a time duration, the environment chooses a rate vector from the allowable set of rates for that mode, and the state of the system is evolved accordingly. The recipe for selecting their choices, or *moves*, is formalized in the form of a strategy that is a function of the history of the game so far to a move of the player. A strategy is called *positional* if it a function only of the current state. We say that the scheduler wins the schedulability game, or has a winning strategy, from a given starting state if there is a scheduler strategy such that irrespective of the strategies of the environment the state of the system stays within the safety set and time does not converge to any real number. Similarly, we say that the environment has a winning strategy if she has a strategy such that for any strategy of the scheduler the system leaves the safety set in a finite amount of time, or the time converges to some real number. One of the central results of this paper is that the schedulability games on BMS are *determined*, i.e. for each starting state exactly one of the player has a winning strategy. Note that the determinacy of these games could be proved using more general results on determinacy, e.g. [10], however our proof is direct and shows the existence of positional winning strategies.

We distinguish between two kind of strategies of scheduler–the *static strategies*, where scheduler can not observe the decisions of the environment, and the *dynamic strategies*, where scheduler can observe the decisions of the environment so far before choosing a mode and a time. Static strategies correspond precisely to schedules, and we often use these two terms interchangeably. A key challenge in the schedulability analysis of BMS is that static strategies are not sufficient as is clear from the following example.

EXAMPLE 3. *Consider the BMS of Figure 1 (right) where the rates in mode m_1 and m_2 lie in $\{(0, 1+\delta) : 0 \leq \delta \leq \varepsilon\}$ and $\{(0, -(1+\delta)) : 0 \leq \delta \leq \varepsilon\}$, respectively. We hint that there is no static winning strategy of scheduler in this BMS (the formal conditions on where the static winning strategy exists will be analyzed later in the paper). Let us assume, for example, that $\sigma = (m_1, t_1), (m_2, t_2), \ldots$ is a static non-Zeno winning strategy of the scheduler. Moreover consider two strategies π and π' of the environment that differ only in mode m_1 where they propose rates $(1,0)$ and $(1+\varepsilon, 0)$ respectively. Let ϱ and ϱ' be the sequences of system states and player's choices—what we subsequently refer to as runs—as the game progresses from a starting state (x_0, y_0) where the environment uses strategy π and π', respectively, against scheduler's strategy σ. Let $T_1(i)$ and $T_2(i)$ be the time spent in mode m_1 and m_2, resp., till the i-th round in runs ϱ and ϱ', while T_1 and T_2 be total time spent in mode m_1 and m_2, resp. The state of the system in the runs ϱ and ϱ' after i rounds will be $(x_0, y_0 + T_1(i) - T_2(i))$ and $(x_0, y_0 + T_1(i) - T_2(i) + T_1(i)\varepsilon)$. Hence the distance $T_1(i)\varepsilon$ between states reached after i-rounds in runs ϱ and ϱ' tends to $T_1\varepsilon$ as i tends to ∞. It is easy to see that if σ is a winning strategy then $T_1 = \infty$; since if $T_1 < \infty$ and $T_2 = \infty$ then the sys-*

tem will move in the direction of rates of mode m_2, while if both T_1 and T_2 are finite then the strategy is not non-Zeno. Hence system will eventually leave any bounded safety set, contradicting our assumption on σ being a winning strategy.

The techniques used for schedulability analysis and schedule construction for CMS cannot be generalized to BMS since in a BMS, the scheduler may not have a strategy to loop back to the starting state. In fact, in general scheduler does not have a strategy to revisit any state as is clear from Figure 1 (right)—here the environment can always choose a rate vector in both mode m_1 and m_2 to avoid any previously visited state. However, from our results on BMS it follows that if the scheduler has a winning strategy then he has a strategy to restrict the future states of the system to a ball of arbitrary diameter centered around the starting state.

In order to solve schedulability game for BMS we exploit the following observation: the scheduler has a winning strategy, from all the starting states in the interior of the safety set S, if and only if there is a polytope $P \subseteq S$, such that for every vertex $\overline{v}$ of P there is a mode $m(\overline{v})$ and time $t(\overline{v})$ such that choosing mode $m(\overline{v})$ for time $t(\overline{v})$ from the vertex $\overline{v}$, the line $\overline{v} + \vec{r}t(\overline{v})$ stays within polytope P for all allowable rates $\vec{r}$ of $m(\overline{v})$. In other words, for any vertex of P there is a mode and a time duration such that if the system evolves with any rate vector of that mode for such amount of time, the system stays in P. For a BMS $\mathcal{H}$ we call such a polytope *$\mathcal{H}$-closed*. The *$\mathcal{H}$-closed* polytope is similar to controlled invariant set in control theory literature (see [2] for a comprehensive review). We show how such a polytope can be constructed for a BMS based on its characteristics. We also analyze the complexity of such a construction. The existence of an $\mathcal{H}$-closed polytope immediately provides a non-Zeno safe dynamic strategy for the scheduler for any starting state in P: find the convex coefficient $(\lambda_1, \lambda_2, \ldots, \lambda_k)$ of the current state $\overline{x}$ with respect to the finite set of vertices $(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_k)$ of P and choose the mode $m(\overline{x}_i)$ for time $t(\overline{x}_i)\lambda_i$ that maximizes $t(\overline{x}_i)\lambda_i$. Then, for some choice $\vec{r}$ of the environment for $m(\overline{x}_i)$ the system will progress to $\overline{x}'{=}\overline{x}{+}t(\overline{x}_i)\lambda_i\vec{r}$. One can repeat this dynamic strategy from the next state $\overline{x}'$ as the current state. We prove that such strategy is both non-Zeno and safe.

An extreme-rate CMS of a BMS $\mathcal{H}$ is obtained by preserving the set of modes, and for each mode assigning a rate which is a vertex of the available rate-set of that mode. The main result of the paper is that an $\mathcal{H}$-closed polytope exists for a BMS $\mathcal{H}$ iff all *extreme-rate* CMSs of $\mathcal{H}$ are schedulable. The "only if" direction of the above characterization is immediate as if some extreme-rate CMS is not schedulable then the environment can fix those rate vectors and win the schedulability game in the BMS. We show the "if" direction by explicitly constructing the $\mathcal{H}$-closed polytope.

Example 4. *Consider the BMS $\mathcal{H}$ from Figure 1 (right) with $\varepsilon = 0.5$. The safety set is given as a shaded area in Figure 2 (left) and $\bar{x}_0 = (-1, -0.5)$ is the initial state. Observe that all extreme-rate combinations are schedulable and hence we show a winning strategy. An $\mathcal{H}$-closed polytope for this BMS is the line-segment between the points $(0, 2.5)$ and $(0, -2.5)$ (we explain the construction of such polytope in Section 3). After translating this line-segment to x_0 and scaling it to fit inside the safety set, we will get the line-segment connecting $\bar{x}_1 = (-1, 1)$ to $\bar{x}_2 = (-1, -2)$, as shown in Figure 2 (left). At vertices $\bar{x}_1$ and $\bar{x}_2$ modes m_2*

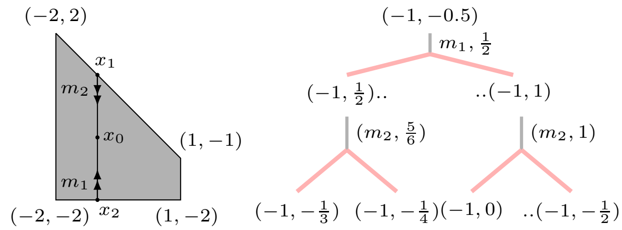


Figure 2: $\mathcal{H}$-closed polytope and dynamic strategy

and m_1, respectively, can be used for 1 time unit. A winning strategy of scheduler is to keep the system's state along the line segment. Our strategy observes the current state $\overline{x}$ and finds the mode to choose by computing convex coefficient $\lambda{\in}[0, 1]$ s.t. $\overline{x}{=}\lambda\overline{x}_1{+}(1{-}\lambda)\overline{x}_2$. For instance, at state $\bar{x}_0{=}\frac{1}{2}\bar{x}_1{+}\frac{1}{2}\bar{x}_2$ the scheduler can choose any of the modes for $\frac{1}{2}$ time units. Assume that it chooses m_1. Based on environment's choice the state of system after $\frac{1}{2}$ time units will be in the set $\{-1, 0.5 + \delta : 0 \leq \delta \leq 0.5\}$. The scheduler observes this new state after $\frac{1}{2}$ time-unit, and chooses mode and time accordingly. For example, if the environment chooses $(0, 1.25)$ and so the next state is $\bar{x} = (-1, 0.75) = \frac{1}{12}\bar{x}_1{+}\frac{11}{12}\bar{x}_2$, scheduler can choose mode m_2 for $\frac{11}{12}$ time units. In Figure 2 (right) we show first two rounds of the game. Since, for any point on our line segment scheduler can choose a mode for at least 0.5 time unit and stay on the line segment, such strategy is both safe and non-Zeno.

We also extend the above result to decide the winner starting from arbitrary states, i.e. including those states that lie on the boundary of the safety set. Here we show that the existence of a safe scheduler implies the existence of a safe scheduler which only allows to move from lower-dimensional faces to higher-dimensional ones and not the other way around; this allows us to use an algorithm which traverses the face lattice of the safety set and analyses each face one by one. We also prove co-NP completeness of the schedulability problem, showing the hardness by giving a reduction from 3-SAT to the non-schedulability problem. On a positive note, we show that if the number of variables is two, then the schedulability game can be decided in polynomial time. This is because in such a case we can prove that there is only polynomially many candidates for falsifiers we need to consider, and hence we can check each of them one by one. Finally, we study a discrete version of schedulability games where scheduler can choose time delays only at multiples of a given clock period, while the environment can choose rate vectors from a finite set. We show that discrete schedulability games on BMS are EXPTIME-complete, and that the maximal clock period for which scheduler has a winning strategy can be computed in exponential time. If the system is a CMS, we get a PSPACE algorithm, improving the result of [1] where only an approximation of the maximal clock period for CMS was studied.

We refer to [12, 11] and [1] for a review of related work on CMS and green scheduling. Heymann et al. [6] considered scheduling problem on BMS where rate-vectors are given as upper and lower rate matrices and the safety set as the entire non-negative orthant. They showed that the scheduler wins if he wins in the CMS of the lower rate matrix, and wins only if he wins in the CMS of the upper rate matrix. We study more general BMS and safety sets, and characterize necessary and sufficient condition for schedulability. To complete the picture, we remark that games on hybrid automata [5, 4], that corresponds to BMS with local invariants and guards, have undecidable schedulability problem.

2. PROBLEM DEFINITION

Points and Vectors. Let $\mathbb{R}$ be the set of real numbers. We represent the states in our system as points in $\mathbb{R}^n$ that is equipped with the standard *Euclidean norm* $\|\cdot\|$. We denote points in this state space by $\overline{x}, \overline{y}$, vectors by $\vec{r}, \vec{v}$, and the i-th coordinate of point $\overline{x}$ and vector $\vec{r}$ by $\overline{x}(i)$ and $\vec{r}(i)$, respectively. We write $\vec{0}$ for a vector with all its coordinates equal to 0; its dimension is often clear from the context. The distance $\|\overline{x}, \overline{y}\|$ between points $\overline{x}$ and $\overline{y}$ is defined as $\|\overline{x} - \overline{y}\|$. For two vectors $\vec{v}_1, \vec{v}_2 \in \mathbb{R}^n$, we write $\vec{v}_1 \cdot \vec{v}_2$ to denote their dot product defined as $\sum_{i=1}^{n} \vec{v}_1(i) \cdot \vec{v}_2(i)$.

Boundedness and Interior. We denote a *closed ball* of radius $d \in \mathbb{R}_{\geq 0}$ centered at $\overline{x}$ as $B_d(\overline{x}) = \{\overline{y} \in \mathbb{R}^n \ : \ \|\overline{x}, \overline{y}\| \leq d\}$. We say that a set $S \subseteq \mathbb{R}^n$ is *bounded* if there exists $d \in \mathbb{R}_{\geq 0}$ such that for all $\overline{x}, \overline{y} \in S$ we have $\|\overline{x}, \overline{y}\| \leq d$. The *interior* of a set S, $\text{int}(S)$, is the set of all points $\overline{x} \in S$ for which there exists $d > 0$ s.t. $B_d(\overline{x}) \subseteq S$.

Convexity. A point $\overline{x}$ is a *convex combination* of a finite set of points $X = \{\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_k\}$ if there are $\lambda_1, \lambda_2, \ldots, \lambda_k \in [0, 1]$ such that $\sum_{i=1}^{k} \lambda_i = 1$ and $\overline{x} = \sum_{i=1}^{k} \lambda_i \cdot \overline{x}_i$. The *convex hull* of X is then the set of all points that are convex combinations of points in X. We say that $S \subseteq \mathbb{R}^n$ is *convex* iff for all $\overline{x}, \overline{y} \in S$ and all $\lambda \in [0, 1]$ we have $\lambda \overline{x} + (1 - \lambda)\overline{y} \in S$ and moreover, S is a *convex polytope* if it is bounded and there exists $k \in \mathbb{N}$, a matrix A of size $k \times n$ and a vector $\vec{b} \in \mathbb{R}^k$ such that $\overline{x} \in S$ iff $A\overline{x} \leq \vec{b}$. We write *rows*$(M)$ for the number of rows in a matrix M, here *rows*$(A) = k$.

A point $\overline{x}$ is a *vertex* of a convex polytope P if it is not a convex combination of two distinct (other than $\overline{x}$) points in P. For a convex polytope P we write $\mathsf{vert}(P)$ for the finite set of points that correspond to the vertices of P. Each point in P can be written as a convex combination of the points in $\mathsf{vert}(P)$, or in other words, P is the *convex hull* of $\mathsf{vert}(P)$. From standard properties of polytopes, it follows that for every convex polytope P and every vertex $\overline{c}$ of P, there exists a vector $\vec{v}$ such that $\vec{v} \cdot \overline{c} = d$ and $\vec{v} \cdot \overline{x} > d$ for all $\overline{x} \in P \setminus \{\overline{c}\}$ for some d. We call such a vector $\vec{v}$ a *supporting hyperplane* of the polytope P at $\overline{c}$.

2.1 Multi-Mode Systems

A multi-mode system is a hybrid system equipped with finitely many *modes* and finitely many real-valued *variables*. A configuration is described by values of the variables, which change, as the time elapses, at the rates determined by the modes being used. The choice of rates is nondeterministic, which introduces a notion of adversarial behavior. Formally,

Definition 1 (Multi-Mode Systems). *A multi-mode system is a tuple $\mathcal{H} = (M, n, \mathcal{R})$ where: M is the finite nonempty set of* modes, *n is the number of continuous variables, and $\mathcal{R} : M \to 2^{\mathbb{R}^n}$ is the rate-set function that, for each mode $m \in M$, gives a set of vectors.*

We often write $\vec{r} \in m$ for $\vec{r} \in \mathcal{R}(m)$ when $\mathcal{R}$ is clear from the context. A finite *run* of a multi-mode system $\mathcal{H}$ is a finite sequence of states, timed moves and rate vector choices $\varrho = \langle \overline{x}_0, (m_1, t_1), \vec{r}_1, \overline{x}_1, \ldots, (m_k, t_k), \vec{r}_k, \overline{x}_k \rangle$ s.t. for all $1 \leq i \leq k$ we have $\vec{r}_i \in \mathcal{R}(m_i)$ and $\overline{x}_i = \overline{x}_{i-1} + t_i \cdot \vec{r}_i$. For such a run ϱ we say that $\overline{x}_0$ is the *starting state*, while $\overline{x}_k$ is its *last state*. An infinite run is defined in a similar manner. We write *Runs* and *FRuns* for the set of infinite and finite runs of $\mathcal{H}$, while *Runs*$(\overline{x})$ and *FRuns*$(\overline{x})$ for the set of infinite and finite runs starting from $\overline{x}$.

An infinite run $\langle \overline{x}_0, (m_1, t_1), \vec{r}_1, \overline{x}_1, (m_2, t_2), \vec{r}_2, \ldots \rangle$ is *Zeno* if $\sum_{i=1}^{\infty} t_i < \infty$. Given a set $S \subseteq \mathbb{R}^n$ of safe states, we say that a run $\langle \overline{x}_0, (m_1, t_1), \vec{r}_1, \overline{x}_1, (m_2, t_2), \vec{r}_2, \ldots \rangle$ is S-safe if for all $i \geq 0$ we have that $\overline{x}_i \in S$ and $\overline{x}_i + t \cdot \vec{r}_{i+1} \in S$ for all $t \in [0, t_{i+1}]$, assuming $t_0 = 0$. Notice that if S is a convex set and $\overline{x}_i \in S$ for all $i \geq 0$, then for all $i \geq 0$ and for all $t \in [0, t_{i+1}]$ we have that $\overline{x}_i + t \cdot \vec{r}_{i+1} \in S$. The concept of S-safety for finite runs is defined in a similar manner. Sometimes we simply call a run safe when the safety set and the starting state is clear from the context.

We formally give the semantics of a multi-mode system $\mathcal{H}$ as a turn-based two-player game between the players, *scheduler* and *environment*, who choose their moves to construct a run of the system. The system starts in a given starting state $\overline{x}_0 \in \mathbb{R}^n$ and at each turn scheduler chooses a timed move, a pair $(m, t) \in M \times \mathbb{R}_{>0}$ consisting of a mode and a time duration, and the environment chooses a rate vector $\vec{r} \in \mathcal{R}(m)$ and as a result the system changes its state from $\overline{x}_0$ to the state $\overline{x}_1 = \overline{x}_0 + t \cdot \vec{r}$ in t time units following the linear trajectory according to the rate vector $\vec{r}$. From the next state $\overline{x}_1$ the scheduler again chooses a timed move and the environment an allowable rate vector, and the game continues forever in this fashion. The focus of this paper is on safe-*schedulability game*, where the goal of the scheduler is to keep the states of the system within a given safety set S, while ensuring that the time diverges (non-Zenoness requirement). The goal of the environment is the opposite, i.e. to visit a state out of the safety set or make the time converge to some finite number.

Given a bounded and convex safety set S, we define *(safe) schedulability objective* $\mathcal{W}^S_{\text{Safe}}$ as the set of S-safe and non-Zeno runs of $\mathcal{H}$. In a schedulability game the winning objective of the scheduler is to make sure that the constructed run of a system belongs to $\mathcal{W}^S_{\text{Safe}}$, while the goal of the environment is the opposite. The choice selection mechanism of the players is typically defined as strategies. A *strategy* σ of scheduler is function $\sigma{:}\textit{FRuns}{\to}M{\times}\mathbb{R}_{\geq 0}$ that gives a timed move for every history of the game. A strategy π of the environment is a function $\pi : \textit{FRuns} \times (M \times \mathbb{R}_{\geq 0}) \to \mathbb{R}^n$ that chooses an allowable rate for a given history of the game and choice of the scheduler. We say that a strategy is *positional* if it suggests the same action for all runs with common last state. We write Σ and Π for the set of strategies of the scheduler and the environment, respectively.

Given a starting state $\overline{x}_0$ and a strategy pair $(\sigma, \pi) \in \Sigma \times \Pi$ we define the unique run *Run*$(\overline{x}_0, \sigma, \pi)$ starting from $\overline{x}_0$ as

$$\textit{Run}(\overline{x}_0, \sigma, \pi) = \langle \overline{x}_0, (m_1, t_1), \vec{r}_1, \overline{x}_1, (m_2, t_2), \vec{r}_2, \ldots \rangle$$

where for all $i \geq 1$, $(m_i, t_i) = \sigma(\langle \overline{x}_0, (m_1, t_1), \vec{r}_1, \overline{x}_1, \ldots, \overline{x}_{i-1} \rangle)$ and $\vec{r}_i = \pi(\langle \overline{x}_0, (m_1, t_1), \vec{r}_1, \overline{x}_1, \ldots, \overline{x}_{i-1}, m_i, t_i \rangle)$ and $x_i = x_{i-1} + t_i \cdot \vec{r}_i$. The scheduler wins the game if there is $\sigma \in \Sigma$ such that for all $\pi \in \Pi$ we get *Run*$(\overline{x}_0, \sigma, \pi) \in \mathcal{W}^S_{\text{Safe}}$. Such a strategy σ is *winning*. Similarly, the environment wins the game if there is $\pi \in \Pi$ such that for all $\sigma \in \Sigma$ we have *Run*$(\overline{x}_0, \sigma, \pi) \notin \mathcal{W}^S_{\text{Safe}}$. Again, π is called *winning* in this case. If a winning strategy for scheduler exists, we say that $\mathcal{H}$ is schedulable for S and $\overline{x}_0$ (or simply *schedulable* if S and $\overline{x}_0$ are clear from the context). The following is the main algorithmic problem studied in this paper.

Definition 2 (Schedulability). *Given a multi-mode system $\mathcal{H}$, a safety set S, and a starting state $\overline{x}_0 \in S$, the (safe)* schedulability problem *is to decide whether there exists a winning strategy of the scheduler.*

2.2 Bounded-Rate Multi-Mode Systems

To algorithmically decide schedulability problem, we need to restrict the range of $\mathcal{R}$ and the domain of safety set S in a schedulability game on a multi-mode system. The most general model that we consider is the bounded-rate multi-mode systems (BMS) that are multi-mode systems $(M, n, \mathcal{R})$ such that $\mathcal{R}(m)$ is a convex polytope for every $m \in M$. We also assume that the safety set S is specified as a convex polytope. In our proofs we often refer to another variant of multi-mode systems in which there are only a fixed number of different rates in each mode (i.e. $\mathcal{R}(m)$ is finite for all $m \in M$). We call such a multi-mode system *multi-rate multi-mode systems* (MMS). Finally, a special form of MMS are *constant-rate multi-mode systems* (CMS) [1] in which $\mathcal{R}(m)$ is a singleton for all $m \in M$. We sometimes use $\mathcal{R}(m)$ to refer to the unique element of the set $\mathcal{R}(m)$ in a CMS. The concepts for the schedulability games for BMS and MMS are already defined for multi-mode systems. Similar concepts also hold for CMS but note that the environment has no real choice in this case. For this reason, we can refer to a schedulability game on CMS as a one-player game.

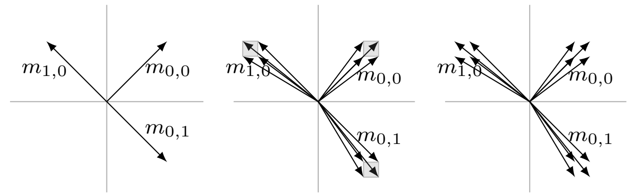
The prime [1] practical motivation for studying CMS was to generalize results on green scheduling problem by Nghiem et al. [12]. We argue that BMS are a suitable abstraction to study green scheduling problem when various rates of temperature change are either uncertain or follow a complex and time-varying dynamics, as shown in the following example.

Example 5 (Green Scheduling). *Consider a building with two rooms A and B. HVAC units in each zone can be in one of the two modes 0 (OFF) and 1 (ON). We write the mode of the combined system as $m_{i,j}$ to represent the fact that rooms A and B are in mode $i \in \{0, 1\}$ and $j \in \{0, 1\}$, respectively. The rate of temperature change and the energy usage for each room is given below.*

Zones	*ON*	*OFF*
A (temp. change rate/ usage)	*-2/2*	*2/1*
B (temp. change/ usage)	*-2/2*	*2/1*

Following [1] we assume that the energy cost is equal to energy usage if peak energy usage at any given point in time is less than or equal to 3 units, otherwise energy cost is 10 times of that standard rate. It follows that to minimize energy cost the peak usage, if possible, must not be higher than 3 units at any given time. We can model the system as a CMS with modes $m_{0,0}$, $m_{0,1}$, and $m_{1,0}$, because these are the only ones that have peak usage at most 3. The variables of the CMS are the temperature of the rooms, while the safety set is the constraint that temperature of both zones should be between 65^oF to 75^oF. The existence of a winning strategy in CMS implies the existence of a switching schedule with energy peak demand less than or equal to 4 units. In Figure 3.(a) we show a graphical representation of such CMS with three modes $m_{0,0}, m_{0,1}$ and $m_{1,0}$ and two variables (corresponding to the two axes). The rate of the variables in mode $m_{0,0}$ is $(2, 2)$, in mode $m_{0,1}$ is $(2, -2)$, and in mode $m_{1,0}$ is $(-2, 2)$.

Now assume that the rate of temperature change in a mode is not constant and can vary within a given margin $\varepsilon > 0$.



(a) Constant-Rate (b) Bounded-Rate (c) Multi-Rate

Figure 3: Restricted Multi-mode Systems

Schedulability problem for such system can best be modeled as a BMS as shown in Figure 3.(b) where the polytope of possible rate vectors is shown as a shaded region. In Figure 3.(c) we show a MMS where variables can only change with the extreme rates of the BMS in Figure 3.(b).

We say that a CMS $H = (M, n, R)$ is an instance of a multi-mode system $\mathcal{H} = (M, n, \mathcal{R})$ if for every $m \in M$ we have that $R(m) \in \mathcal{R}(m)$. For example, the CMS shown in Figure 3.(a) is an instance of BMS in Figure 3.(b). We denote the set of instances of a multi-mode system $\mathcal{H}$ by $[\![\mathcal{H}]\!]$. Notice that for a BMS $\mathcal{H}$ the set $[\![\mathcal{H}]\!]$ of its instances is uncountably infinite, while for a MMS $\mathcal{H}$ the set $[\![\mathcal{H}]\!]$ is finite whose size is exponential in the size of $\mathcal{H}$. We say that a MMS $(M, n, \mathcal{R}')$ is the *extreme-rate* MMS of a BMS $(M, n, \mathcal{R})$ if $\mathcal{R}'(m) = \mathsf{vert}(\mathcal{R}(m))$. The MMS in Figure 3.(c) is the extreme-rate MMS for the BMS in Figure 3.(b) We write $Ext(\mathcal{H})$ for the extreme-rate MMS of the BMS $\mathcal{H}$.

Notice that for every starting state and winning objective at most one player can have a winning strategy. We say that a game is not *determined* if no player has a winning strategy for some starting state. In the next section we give an algorithm to decide the winner in a schedulability game for an arbitrary starting state. Since for every starting state we can decide the winner, it gives a direct proof of determinacy of schedulability games on BMS. Moreover, it follows from our results that whenever a player has a winning strategy, he has a positional such strategy. These two results together yield the first key results of this paper.

Theorem 1 (Determinacy). *Schedulability games on BMS with convex safety polytopes are positionally determined.*

In Section 4 we analyze the complexity of deciding the winner in a schedulability game. Using a reduction from SAT problem to non-schedulability for a MMS, we prove the following main contribution of the paper.

Theorem 2. *Schedulability problems for BMS and MMS are co-NP complete.*

On a positive note, we also show that schedulability games can be solved in polynomial time for BMS and MMS with two variables.

3. SOLVING SCHEDULABILITY GAMES

In this section we discuss the decidability of the schedulability problem for BMS. We first present a solution for the case when the starting state is in the interior of a safety set, and generalize it to arbitrary starting states in Section 3.2.

3.1 Starting State in the Interior of Safety Set

Alur et al. [1] presented a polynomial-time algorithm to decide if the scheduler has a winning strategy in a schedulability game on a CMS for an arbitrary starting state. In particular, for starting states in the interior of the safety set, they characterized a necessary and sufficient condition.

THEOREM 3 ([1]). *The scheduler has a winning strategy in a* CMS (M, n, R), *with convex safety set* S *and starting state* $\overline{x}_0$ *in the interior of* S, *iff there is* $\vec{t} \in \mathbb{R}_{\geq 0}^{|M|}$ *satisfying:*

$$\sum_{i=1}^{|M|} R(i)(j) \cdot \vec{t}(i) = 0 \text{ for } 1 \leq j \leq n \text{ and } \sum_{i=1}^{|M|} \vec{t}(i) = 1. \quad (1)$$

We call a CMS *safe* if it satisfies (1) and we call H unsafe otherwise. The intuition behind Theorem 3 is that the scheduler has a winning strategy if and only if it is possible to return to the starting state in strictly positive time units. From the results of [1] it also follows that whenever a winning strategy exists, there is a strategy which does not look at a history or even the current state, but only uses a bounded counter of size $\ell \leq |M|-1$ and after after a history of length k makes a decision only based on the number k modulo ℓ. Such strategies are called *periodic*.

It is natural to ask whether the approach of [1] can be generalized to BMS. Unfortunately, Example 3 shows that in a BMS although a winning strategy may exist, it may not be possible to return to the initial state, or indeed visit any state twice. Another natural question to ask is whether a suitable generalization of periodic strategies suffice for BMS. *Static* strategies are BMS analog of periodic strategies that behave in the same manner irrespective of the choices of the environment, i.e. for a static strategy σ we have that $\sigma(\rho) = \sigma(\rho')$ for all runs $\rho = \langle \overline{x}_0, (m_1, t_1), \vec{r}_1, \overline{x}_1, \ldots, (m_k, t_k), \vec{r}_k, \overline{x}_k \rangle$ and $\rho' = \langle \overline{x}_0, (m_1, t_1), \vec{r}'_1, \overline{x}'_1, \ldots, (m_k, t_k), \vec{r}'_k), \overline{x}'_k \rangle$. Static strategies are often desirable in the settings where scheduler can not observe the state of the system. However, as we show in Appendix A, except for the degenerate cases when the BMS contains a subset of modes which induce a safe CMS, scheduler can never win a game on BMS using static strategies. We saw an example of this phenomenon in the Introductory section as Figure 1.(c).

This negative observations imply that to solve the schedulability games for BMS one needs to take a different approach. In the rest of this section, we define the notion of $\mathcal{H}$-closed polytope and show that if such a polytope exists, then for any convex set S we can construct a winning *dynamic* strategy which takes its decisions only based on the last state. We also extend the notion of safety of a CMS to BMS. We say that a BMS $\mathcal{H}$ is *safe* if all instances of its extreme-rate MMS $Ext(\mathcal{H})$ are safe i.e. all $H \in [\![Ext(\mathcal{H})]\!]$ satisfy (1). Finally, we connect (Lemmas 5 and 6) the existence of $\mathcal{H}$-closed polytope with the safety of the BMS.

Dynamic Scheduling Algorithm. For a BMS $\mathcal{H}$ we call a convex polytope P $\mathcal{H}$*-closed*, if for every vertex of P there exists a mode m such that all the rate vectors of m keep the system in P, i.e. for all $\overline{c} \in \mathsf{vert}(P)$ there exists $m \in M$ and $\tau \in \mathbb{R}_{>0}$ such that for all $\vec{r} \in \mathcal{R}(m)$ we have that $\overline{c} + \vec{r} \cdot t \in P$ for all $t \in [0, \tau]$. An example of a $\mathcal{H}$-closed polytope is given in the Example 4.

Assume that for any $\gamma > 0$ and $\overline{x}_0$ we are able to compute a $\mathcal{H}$-closed polytope which is fully contained in $B_\gamma(\overline{x}_0)$ and contains $\overline{x}_0$. If this is the case, we can use Algorithm 1 to compute a dynamic scheduling strategy. The idea of the algorithm is to build a $\mathcal{H}$-closed polytope which contains the initial state and is fully contained within S, and then construct the strategy based on the modes safe at the vertices of the polytope. The correctness of the algorithm is established by the following proposition.

Algorithm 1: Dynamic scheduling algorithm

Input: BMMS $\mathcal{H}$, starting state $\overline{x}_0$
Output: non-Terminating Scheduling Algorithm
1 $\gamma :=$ the shortest distance of $\overline{x}_0$ from borders of S;
2 $P := \mathcal{H}$-closed polytope s.t. $P \subseteq B_\gamma(\overline{x}_0)$ and $\overline{x}_0 \in P$;
3 **foreach** $\overline{c} \in \mathsf{vert}(P)$ **do**
4   **foreach** *mode* $m \in M$ **do**
5   **foreach** *extreme rate vector* $\vec{r} \in m$ **do**
6             $t_{\vec{r}} = \max\{t : \overline{c} + \vec{r} \cdot t \in P\}$;
7         $\delta_m = \min_{\vec{r} \in m} t_{\vec{r}}$;
8     $m_* = \arg\max_{m \in M} \delta_m$; $\quad \Delta_{\overline{c}} = \delta_{m_*}$; $\quad m_{\overline{c}} = m_*$;
9 **while** *true* **do**
10   Store current state as $\overline{x}$;
11   Find $(\lambda_{\overline{c}} \geq 0)_{\overline{c} \in \mathsf{vert}(P)}$ where $\overline{x} = \sum_{\overline{c} \in \mathsf{vert}(P)} \lambda_{\overline{c}} \cdot \overline{c}$;
12   Find $\overline{c}_* = \arg\max_{\overline{c} \in \mathsf{vert}(P)} \lambda_{\overline{c}} \cdot \Delta_{\overline{c}}$;
13   Schedule mode $m_{\overline{c}_*}$ for $\lambda_{\overline{c}_*} \cdot \Delta_{\overline{c}_*}$;

PROPOSITION 4. *If there exists an* $\mathcal{H}$*-closed polytope and it can be effectively computed then Algorithm 1 implements a winning dynamic strategy for the scheduler.*

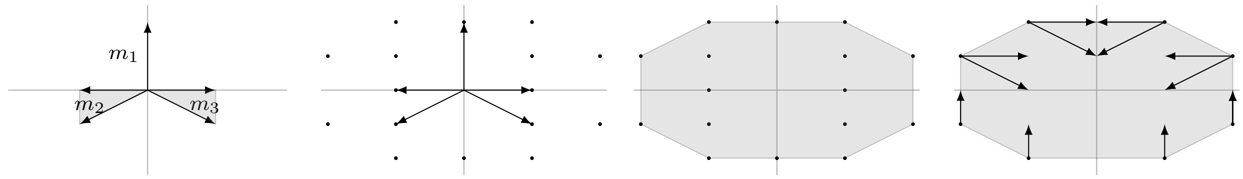
PROOF. Assume that there exists an $\mathcal{H}$-closed polytope and we have an algorithm to compute it. Observe that the strategy is non-Zeno, because $\lambda_{\overline{c}_*} \cdot \Delta_{\overline{c}_*}$ on line 13 is bounded from below by $\frac{1}{|\mathsf{vert}(P)|} \cdot \min_{\overline{c} \in \mathsf{vert}(P)} \Delta_{\overline{c}}$ for any point of P, and $\Delta_{\overline{c}}$ are positive by their construction and the definition of the $\mathcal{H}$-closed polytope. Next, we need to show that under the computed strategy we never leave the convex polytope P. For a state $\overline{x}$ which is of the form $\sum_{\overline{c} \in \mathsf{vert}(P)} \lambda_{\overline{c}} \cdot \overline{c}$, the successor state will be $\overline{x}' = (\sum_{\overline{c} \in \mathsf{vert}(P)} \lambda_{\overline{c}} \cdot \overline{c}) + \lambda_{\overline{c}_*} \cdot \Delta_{\overline{c}_*} \cdot \vec{r}$ where $\vec{r}$ is the rate picked by the environment. We can rewrite $\overline{x}'$ as $(\sum_{\overline{c} \in \mathsf{vert}(P) \setminus \{\overline{c}_*\}} \lambda_{\overline{c}} \cdot \overline{c}) + \lambda_{\overline{c}_*} \cdot (\overline{c}_* + \vec{r} \cdot \Delta_{\overline{c}_*})$. Since $\overline{c}_* + \vec{r} \cdot \Delta_{\overline{c}_*} \in P$, we get that $\overline{x}'$ is a convex combination of points in P and hence lies in P. $\square$

Constructing $\mathcal{H}$-Closed Polytope. We will next show how to implement line 2 of Algorithm 1. We give necessary and sufficient conditions for existence of $\mathcal{H}$-closed polytopes in the following two lemmas. The first lemma shows that an $\mathcal{H}$-closed polytope exists if and only if for any hyperplane (given by its normal vector $\vec{v}$) there exists a mode m such that all its rates stay at one side of the hyperplane.

LEMMA 5. *For a* BMS $\mathcal{H}$, *a state* $\overline{x}_0$ *and* $\gamma > 0$, *there is a* $\mathcal{H}$*-closed polytope* $P \subseteq B_\gamma(\overline{x}_0)$ *with* $\overline{x}_0 \in P$ *if and only if for every* $\vec{v}$ *there is a mode* m *such that* $\vec{v} \cdot \vec{r} \geq 0$ *for all* $\vec{r} \in m$.

PROOF. Let us fix a BMS $\mathcal{H} = (M, n, \mathcal{R})$. The proof is in two parts. For $\Rightarrow$, assume that the system is schedulable but there exists a vector $\vec{v}$ such that for all modes $m \in M$ there is a rate $\vec{r}_m \in m$ where $\vec{v} \cdot \vec{r}_m < 0$. It implies that if the adversary fixes the rates $\vec{r}_m$ whenever the scheduler chooses m, then the system moves in the direction of vector $-\vec{v}$ (i.e. for all d a state $\overline{x}$ will be reached such that $\vec{v} \cdot \overline{x} < d$), and hence for any bounded safety set and non-Zeno strategy system will leave the safety set. This contradicts with existence of $\mathcal{H}$-closed polytope implying winning scheduler strategy.

To prove the other direction, let $R = \{\vec{r}_1, \ldots, \vec{r}_N\}$ be the set of rates occurring in modes of the extreme-rate MMS of



(a) (b) (c) (d)

Figure 4: Constructing closed convex polytope

$\mathcal{H}$, i.e. $R = \{\mathcal{R}'(m) : (M, n, \mathcal{R}') \in [\![Ext(\mathcal{H})]\!], m \in M\}$. We claim the following to be the $\mathcal{H}$-closed polytope:

$$P := \{\overline{x}_0 + D \cdot \sum_{i=1}^{N} \vec{r}_i \cdot p_i \mid p_i \in [0,1]\}, \tag{2}$$

where $D = \gamma / \sum_{i=1}^{N} \|\vec{r}_i\|$. Notice that P is a convex polytope since it is a convex hull of points $\overline{x}_0 + D \cdot \sum_{i=1}^{N} \vec{r}_i \cdot p_i$ where $p_i \in \{0,1\}$. Also, due to our choice of D, $P \subseteq B_\gamma(\overline{x}_0)$, and $\overline{x}_0 \in P$. For the sake of contradiction we assume that for every $\vec{v}$ there is a mode m such that all rates $\vec{r}$ of m satisfy $\vec{v} \cdot \vec{r} \geq 0$, but at least one corner $\overline{c}$ of P does not satisfy the defining condition of $\mathcal{H}$-closed polytope, i.e. for all modes i there is a rate vector $\vec{r}_i$ satisfying

$$\overline{c} + t \cdot \vec{r}_i \notin P \text{ for all } t > 0 \tag{3}$$

Let us fix such corner $\overline{c}$. By the supporting hyperplane theorem there is a vector $\vec{v}$ such that, for some d:

$$\vec{v} \cdot \overline{c} \;=\; d \tag{4}$$
$$\vec{v} \cdot \overline{x} \;>\; d, \text{ for all } \overline{x} \in P \setminus \{\overline{c}\} \tag{5}$$

i.e. $\vec{v}$ is supporting P on $\overline{c}$. Let us fix some mode m such that for all rates $\vec{r}$ of m we have $\vec{v} \cdot \vec{r} \geq 0$. Notice that this exists by the assumption. Let $\vec{r}_i$ be a rate of m satisfying (3).

By the definition of P the point $\overline{c}$, a corner of P, is of the form $\overline{x}_0 + D \cdot \sum_{j=1}^{N} \vec{r}_j \cdot p_j$ for some $p_j \in [0,1]$ where $1 \leq j \leq N$ and $\vec{r}_j \in R$. We necessarily have $p_i = 1$, because if $p_i = 1 - \delta$ for some $\delta > 0$, then $\overline{c} + D \cdot \varepsilon \cdot \vec{r}_i \in P$ for any $\varepsilon \leq \delta$ and that will contradict with (3). Notice that for all $k \in [0,1]$ the points $\overline{y}_k = \overline{x}_0 + D \cdot \sum_{j=1}^{N} p_j^k \cdot \vec{r}_j$, where $p_j^k = p_j$ if $j \neq i$ and $p_j^k = k$ otherwise, are all in P. Also notice that point $\overline{y}_1 = \overline{c}$ and for each $k \in [0,1]$ we have that $\overline{y}_k = \overline{y}_0 + D \cdot k \cdot \vec{r}_i$. In particular, $\overline{c} = \overline{y}_1 = \overline{y}_0 + D \cdot \vec{r}_i$. It follows that $\overline{c} - D \cdot \vec{r}_i = \overline{y}_0 \in P$. W.l.o.g. we assume $\vec{r}_i \neq \vec{0}$. Hence, from (5) we get $\vec{v} \cdot (\overline{c} - D \cdot \vec{r}_i) > d$. By rearranging we get $\vec{v} \cdot \overline{c} - D \cdot \vec{v} \cdot \vec{r}_i > d$, and because $\vec{v} \cdot \overline{c} = d$, we get $D \cdot \vec{v} \cdot \vec{r}_i < 0$ which contradicts that $\vec{v} \cdot \vec{r}_i \geq 0$. ☐

Figures 4.(b)-(c) show how to construct $\mathcal{H}$-closed polytope from (2) for the BMS in Figure 4.(a), while Figure 4.(d) shows that for every corner of the constructed polytope there is a mode that keeps the system inside the polytope.

The following lemma finally gives an algorithmically checkable characterization of existence of $\mathcal{H}$-closed polytope.

LEMMA 6. *Let $\mathcal{H} = (M, n, \mathcal{R})$ be a BMS. We have that for every $\vec{v}$ there is a mode m such that $\vec{v} \cdot \vec{r} \geq 0$ for all $\vec{r} \in m$ if and only if $\mathcal{H}$ is safe.*

PROOF. In one direction, let us assume that $(M, n, R) \in [\![Ext(\mathcal{H})]\!]$ is not safe, and let $Q = \{R(m) \mid m \in M\}$. Then

---

Algorithm 2: Schedulability Problem for Interior Starting States.

Input: BMS $\mathcal{H}$, $\overline{x} \in \mathbb{R}^n$ and $\gamma > 0$
Output: $\mathcal{H}$-closed polytope P contained in $B_\gamma(\overline{x})$ s.t. $\overline{x} \in P$, No if there is no $\mathcal{H}$-closed polytope.

foreach *CMS* $H = (M, n, R)$ *of* $[\![Ext(\mathcal{H})]\!]$ **do**
  Check if there is a satisfying assignment for:

$$\begin{aligned} \textstyle\sum_{m \in M} R(m) \cdot t_m &= \vec{0} \\ \textstyle\sum_{m \in M} t_m &= 1 \\ t_m &\geq 0 \text{ for all } m \in M. \end{aligned} \tag{6}$$

  if *no satisfying assignment exists* **then return** NO
$R := \{\vec{r}_1, \vec{r}_2, ..., \vec{r}_N\}$ be the set of rate vectors of $[\![Ext(\mathcal{H})]\!]$;
return the polytope given as convex hull of the points $\overline{x} + \frac{\gamma}{\sum_{i=1}^{N} \|\vec{r}_i\|} \cdot \sum_{i=1}^{N} \cdot p_i \vec{r}_i$ where $p_i \in \{0,1\}$;

$\vec{0}$ is not a convex combination of points in Q, and so by supporting hyperplane theorem applied to $\vec{0}$ and the convex hull of Q there is $\vec{v}$ and $d > 0$ such that $\vec{v} \cdot R(m) \geq d$ for all $m \in M$. Since $R(m) \in \mathcal{R}(m)$, this direction of the proof is finished. In the other direction, let $\vec{v}$ be such that there is $\vec{r} \in \mathcal{R}(m)$ for all $m \in M$ such that $\vec{v} \cdot \vec{r} < 0$. Then by convexity of $\mathcal{R}(m)$ there is $\vec{r}_m \in \mathsf{vert}(\mathcal{R}(m))$ with the same properties, and we can create a CMS $(M, n, R) \in [\![Ext(\mathcal{H})]\!]$ by putting $R(m) = \vec{r}_m$. This CMS is not safe, because for any strategy, for a sufficiently large time bound a point $\overline{x}$ will be reached such that $(-\vec{v}) \cdot \overline{x}$ is arbitrarily large, and hence any convex polytope will be left eventually. ☐

Combining Proposition 4 with Lemmas 5 and 6 we get the following main result.

THEOREM 7. *For every BMS $\mathcal{H}$ and the starting state in the interior of a convex and bounded safety set we have that scheduler has a winning strategy if and only if $\mathcal{H}$ is safe.*

Theorem 7 allows us to devise Algorithm 2 and at the same time give its correctness. The reader may have noticed that Theorem 7 bears a striking resemblance to Theorem 3 for CMS, since the former boils down to checking safety of exponentially many CMS instances. Note, however, that the proof here is much more delicate. While in the case of CMS satisfiability of (1) gives immediately a periodic winning strategy, for BMS this is not the case: even when every instance in $[\![Ext(\mathcal{H})]\!]$ is safe, we cannot immediately see which modes should be used by the winning strategy; this requires the introduction of $\mathcal{H}$-closed polytopes.

3.2 General Case

Algorithm 3: Schedulability Problem For Arbitrary Starting State

Input: BMS $\mathcal{H}$, a safety set S given by inequalities $A\vec{x} \le \vec{b}$, and a starting state $\overline{x}_0$.
Output: Yes, if the scheduler wins, No otherwise.

```
Compute the sequence ℐ = ⟨I_1, I_2, ..., I_ℓ⟩;
SCHEDULABLE = ∅, UNSCHEDULABLE = ∅;
foreach I in ℐ do
    if I' ⊆ I and I' ∈ UNSCHEDULABLE then
        UNSCHEDULABLE := UNSCHEDULABLE ∪ {I};
    if ∃m ∈ M with only internal rates then
        SCHEDULABLE := SCHEDULABLE ∪ {(I, ⊥)};
    else
        Construct ℋ_I;
        if ℋ_I is safe and P_I is ℋ_I-closed polytope then
            SCHEDULABLE := SCHEDULABLE ∪ {(I, P_I)} ;
        else UNSCHEDULABLE:=UNSCHEDULABLE∪{I};
if ∃I ∈ SCHEDULABLE and x̄_0 ⊨ S|_I then return Yes;
else return No;
```

In this section we present Algorithm 3 that analyses schedulability of arbitrary starting states in S. Notice that a starting state on the boundary of the safety polytope may lie on various faces (planes, edges etc.) of different dimensions. The scheduler may have a winning strategy using modes that let the system stay on some lower dimension face, or there may exists a winning strategy where scheduler first reaches a face of higher dimension where it may have a winning strategy. Before we describe steps of our algorithm, we need to formalize a notion of (open) faces of a convex polytope, a concept critical in Algorithm 3.

Let $Ax \le b$ be the linear constraints specifying a convex polytope S. We specify a face of S by a set $I \subseteq \{1, \ldots, rows(A)\}$. We write $\overline{x} \models S|_I$, and we say that $\overline{x}$ satisfies $S|_I$, if and only if $A_{1,j}x(1) + \cdots A_{n,j}x(n) = b_j$ for all $j \in I$, and $A_{1,j}x(1) + \cdots A_{n,j}x(n) < b_j$ for all $j \notin I$, i.e. exactly the inequalities indexed by numbers from I are satisfied tightly. Note that every point of S satisfies $S|_I$ for exactly one I. Although there are potentially uncountably many states in every face of S the following Lemma implies that it is sufficient to analyze only one state in every face.

Lemma 8. *For a BMS, a convex polytope S, and for all faces I of S, either none or all states satisfying $S|_I$ are schedulable. Moreover, if $I' \subseteq I$ and no point satisfying $S|_{I'}$ is schedulable, then no point satisfying $S|_I$ is schedulable.*

Let $\mathcal{I} = \langle I_1, I_2, \ldots \rangle$ be the sequence of all faces such that $S|_{I_i}$ is satisfied by some state, ordered such that if $I_i \subseteq I_j$, then $i \le j$. We call a mode m *unusable for I* if there is $\overline{x} \models S|_I$ and $\vec{r} \in \mathcal{R}(m)$ such that $\overline{x} + \vec{r} \cdot \delta \notin S$ for all $\delta > 0$. The rate $\vec{r}$ satisfying this condition is called *external.* A rate $\vec{r}$ is called *internal* if for any $\overline{x}$ such that $\overline{x} \models S|_I$ there is $\delta > 0$ and j such that $I_j \subseteq I$ and $\overline{x} + \vec{r} \cdot \varepsilon \models S|_{I_j}$ for all $0 < \varepsilon \le \delta$. For a BMS $\mathcal{H}$ and face I we define a BMS $\mathcal{H}_I = (M', n, \mathcal{R}')$ where M' contains all modes of M which are not unusable for I, and $\mathcal{R}'(m)$ is the set of non-internal rates of $\mathcal{R}(m)$.

Theorem 9. *For every BMS $\mathcal{H}$, a convex polytope safety set S, and a starting state $\overline{x}_0 \in S$, Algorithm 3 decides schedulability problem for $\mathcal{H}$. Moreover, one can construct a dynamic winning strategy using the set* Schedulable.

Proof. (Sketch.) Let $\langle I_1, I_2, \ldots \rangle$ be all sets such that $S|_{I_i}$ is satisfied by some state, ordered such that if $I_i \subseteq I_j$, then $i \le j$. Algorithm 3 analyzes the sets I_i, determining whether the points satisfying $S|_{I_i}$ are schedulable (in which case we call I_i schedulable), or not. Let us assume that I is the first element of the sequence $\langle I_1, I_2 \ldots \rangle$ which has not been analyzed yet. If there is I' such that $I' \subseteq I$ and I' is already marked as not schedulable, then by Lemma 8 one can immediately mark I as non-schedulable. If all modes are unusable, then no point $\overline{x}$ such that $S|_I$ is schedulable. Notice that if there exists an internal rate to face I_j then it must necessarily be the case that I_j is schedulable. If there is a mode m which only has internal rates, there is a winning strategy σ for the scheduler which starts by picking m and a sufficiently small time interval t. This will make sure that after one step a point is reached which is already known to be schedulable and scheduler has a winning strategy.

If none of the previous cases match, the algorithm creates a BMS $\mathcal{H}_I$ and applies Theorem 7 to the system $\mathcal{H}_I$. If there is a $\mathcal{H}_I$-closed polyhedron P, we know that I is schedulable and give a winning scheduler's strategy $\sigma_{\overline{x}}$ for any point $\overline{x} \models S|_I$ as follows. Let $d > 0$ be a number such that for any $\overline{y} \models I_j$ where $j > i$ we have $\|\overline{x}, \overline{y}\| > d$, i.e. d is chosen so that all points of S contained in $B_d(\overline{y})$ satisfy $S|_{I'}$ for $I' \subseteq I$ (this follows from the properties of the sequence $I_1, I_2, \ldots$ and because S is a convex polytope). The strategy $\sigma_{\overline{x}}$ works as follows. If all points in the history satisfy $S|_I$, $\sigma_{\overline{x}}$ mimics $\sigma_{\mathcal{H}_I, \overline{x}, d}$. Otherwise, once a point $\overline{y} \not\models S|_I$ is reached, the strategy $\sigma_{\overline{x}}$ starts mimicking $\sigma_{\overline{y}}$. Note that the strategy $\sigma_{\overline{y}}$ is indeed defined by our choice of d and polytopes stored in Schedulable set. Although the strategy we obtain in this way may potentially be non-positional, it is a mere technicality to turn it into a positional one.

If $\mathcal{H}_I$ is not schedulable for any set and any point, then it is easy to see that for no point satisfying $S|_I$ there is a schedulable strategy. Indeed, for any strategy σ, as long as σ picks the modes from M', the environment can play a counter-strategy showing that $\mathcal{H}_I$ is not schedulable. When any mode from $m \in M \setminus M'$ is used by σ, we have that m is unusable and so the environment can pick a rate witnessing m's unusability: this will ensure reaching a point outside S. Hence, we can mark I as unschedulable. □

4. COMPLEXITY

In this section we analyze complexity of the schedulability problem for BMS. We begin by showing that in general it is co-NP-complete, however it can be solved in polynomial time if the system has only two variables.

4.1 General Case

Proposition 10. *The schedulability problem for BMS and convex polytope safety sets is in co-NP.*

Proof (Sketch). We show that when the answer to the problem of schedulability of a point $\overline{x}$ is No, there is a falsifier that consists of two components:

- a set $I \subseteq \{1, \ldots, rows(A)\}$ s.t. $\overline{x} \models S|_{I'}$ for $I' \supseteq I$, and
- a rate combination $(\vec{r}_m)_{m \in M}$ such that there is a set of modes $\mathsf{External} \subseteq M$ where every $\vec{r}_m$ for $m \in \mathsf{External}$

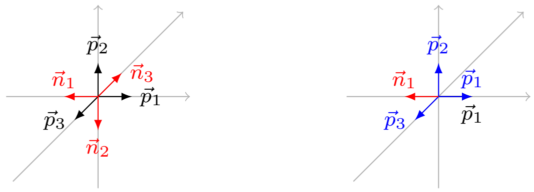


Figure 5: An example from proof of Proposition 11

is external for I; and the rates $\vec{r}_m$ for $m \notin \mathsf{External}$ are neither external, nor internal, and there is a vector $\vec{v}$ such that $\vec{v} \cdot \vec{r}_m > 0$ for all $m \notin \mathsf{External}$.

Let us first show that the existence of this falsifier implies that the answer to the problem is No. Indeed, as long as a strategy of a scheduler keeps using modes $m \notin \mathsf{External}$, the environment can pick the rates $\vec{r}_m$, and a point outside of S will be reached under any non-Zeno strategy, because S is bounded. If the strategy of a scheduler picks any mode $m \in \mathsf{External}$, the environment can win immediately by picking the external rate $\vec{r}_m$ and getting outside of S.

On the other hand, let us suppose that the answer to the problem is No, and let I' be such that $\overline{x} \models S|_{I'}$. Then consider any *minimal non-schedulable* $I \subseteq I'$. We put to External all modes which are unusable, and for every such mode, we pick a rate that witnesses it. Further, there is not any mode with only internal modes and the BMS $\mathcal{H}_I$ must be non-schedulable (otherwise I would be schedulable, or would not be minimal non-schedulable). By Proposition 7 there is an unsafe instance $H = (M', n, R) \in [\![Ext(\mathcal{H}_I)]\!]$. Since M' contains all the modes whose indices are not in External, we can pick the rate from this unsafe instance and we are finished. ☐

Proposition 11 (co-NP hardness). *The schedulability problem for MMS is co-NP hard.*

Proof (Sketch). The proof for co-NP hardness uses a reduction from the classical NP-complete problem 3-SAT. For a SAT instance ϕ we construct a MMS $\mathcal{H}_\phi$ such that ϕ is satisfiable if and only if $\mathcal{H}_\phi$ is not schedulable for any starting state and bounded convex safety set. We only sketch the construction of $\mathcal{H}_\phi$ here and formally prove the correctness of the construction in Appendix B.2. Consider a SAT instance ϕ with k clauses and n variables denoted as $x_1, \ldots, x_n$. The corresponding MMS $\mathcal{H}_\phi = (M, n, \mathcal{R})$ is such that its set of modes $M = \{m_1, \ldots, m_k\}$ corresponds to the clauses in ϕ, and variables are such that variable i corresponds to variable x_i of ϕ. For each variable x_i we define vectors $\vec{p}_i$ and $\vec{n}_i$ such that $\vec{p}_i(i) = 1$, $\vec{n}_i(i) = -1$, and $\vec{p}_i(j) = \vec{n}_i(j) = 0$ if $i \neq j$. The rate-vector function $\mathcal{R}$ is defined such that for each mode m_j and for each SAT variable x_i we have that $\vec{p}_i \in \mathcal{R}(m_j)$ if x_i occurs positively in clause j, and $\vec{n}_i \in \mathcal{R}(m_j)$ if the variable x_i occurs negatively in clause j. The crucial property here is that there is no vector that can have a positive dot product with both $\vec{p}_i$ and $\vec{n}_i$, which allows us to map unsafe rate combinations to satisfying valuations and vice versa. Figure 5 shows an example of the reduction for two different formulas. On the left, we have a satisfiable formula $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$ which gives rise to a MMS with two modes: $\{\vec{p}_1, \vec{p}_2, \vec{p}_3\} \in m_1$ and $\{\vec{n}_1, \vec{n}_2, \vec{n}_3\} \in m_2$. The system has unsafe combination $\vec{p}_1, \vec{n}_2$. In Figure 5 (right) an unsatisfiable formula $(x_1 \vee x_1 \vee x_1) \wedge (\neg x_1 \vee \neg x_1 \vee \neg x_1) \wedge (x_1 \vee x_2 \vee x_3)$ is reduced to a MMS with three modes: $\{\vec{p}_1\} \in m_1$, $\{\vec{n}_1\} \in m_2$, and $\{\vec{p}_1, \vec{p}_2, \vec{p}_3\} \in m_3$. All combinations are safe. ☐

Algorithm 4: Decide if a two dimension BMS is safe.

Input: BMS $\mathcal{H}$ with two variables.
Output: Return Yes, if $\mathcal{H}$ is safe and No otherwise.
Set R to the set of extreme rate vectors of $\mathcal{H}$;
foreach $\vec{r}_\perp \in R$ **do**
  Set $\vec{u}$ to be a perpendicular vectors to $\vec{r}_\perp$;
  foreach $\vec{v} \in \{\vec{u}, -\vec{u}\}$ **do**
    if *for each $m \in M$ there is $\vec{r} \in m$ s.t. $\vec{v} \cdot \vec{r} > 0$ or there is $p > 0$ s.t. $\vec{r} = p\vec{r}_\perp$* **then return** No;
return Yes

The proof of the following easy corollary is postponed to Appendix B.3.

Corollary 12 (co-NP hardness result for BMS). *The schedulability problem for BMS is co-NP hard.*

4.2 BMS with two variables

For a special case of BMS which only have two variables, we show the following result.

Theorem 13. *Schedulability problems for BMS with convex polytope safety sets are in P for systems with 2 variables.*

The rest of the section is devoted to the proof of this theorem. The following lemma shows that to check whether a set of rate vectors $R = \{\vec{r}_1, ..., \vec{r}_k\}$ is unsafe it is sufficient to check properties of vectors $\vec{u}$ perpendicular to some vector of R. This observation yields a polynomial time algorithm.

Lemma 14. *Let R be a set of vectors. There is $\vec{v}$ such that $\vec{v} \cdot \vec{r} > 0$ for all $\vec{r} \in R$ if and only if there are $\vec{u}$ and $\vec{r}_\perp \in R$ satisfying $\vec{u} \cdot \vec{r}_\perp = 0$ and for all $\vec{r} \in R$ either $\vec{u} \cdot \vec{r} > 0$ or $\vec{r} = p \cdot \vec{r}_\perp$ for some $p > 0$.*

Proof (Sketch). To obtain $\vec{v}$ we keep changing $\vec{v}$ until it becomes perpendicular to some vector in R. On the other hand, $\vec{v}$ is obtained from $\vec{u}$ by making a sufficiently small change to $\vec{u}$. A formal proof is given in Appendix B.4. ☐

Consider an unsafe set of rate vectors $R = \{\vec{r}_1, \vec{r}_2, \vec{r}_3, \vec{r}_4\}$ shown in Figure 6 (left). All the rate vectors are on the right side of line $y = 0$ and vector $\vec{v}$ has strictly positive dot product with all of them. As it can be seen in the figure, all the rate vectors are on right-hand side of the line passing through $\vec{r}_1$ and there exists $\vec{u}$ perpendicular to $\vec{r}_1$ such that $\vec{v}' \cdot \vec{r}_i \geq 0$ for all $\vec{r}_i \in R$. Observe that adding a rate vector $\vec{r}_5 = -\vec{r}_1$ to R makes this set of rate vectors safe, and none

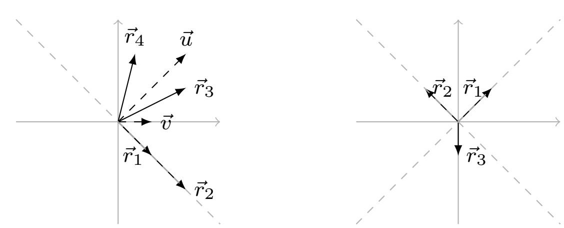


Figure 6: Examples for Lemma 14

of rate vectors would satisfy the conditions of Lemma 14. Figure 6 (right) shows a safe set of rate vectors. As one can see none of rate vectors has the others on one side of itself. The following corollary implies that we can use Lemma 14 to check the safety of a BMS.

COROLLARY 15. *A BMS $\mathcal{H}$ with two variables is not safe if and only if there exists a rate vector $\vec{r}_\perp$ in one of the modes of system and vector $\vec{v}$ perpendicular to it, such that for all modes $m \in \mathcal{H}$: (i) there exists $\vec{r} \in m$ such that $\vec{v} \cdot \vec{r} > 0$; or (ii) $\vec{v} \cdot \vec{r} = 0$ and $\vec{r} = p \cdot \vec{r}_\perp$ for some $p > 0$.*

Algorithm 4 checks whether all the combinations are safe in polynomial time; it chooses a rate vector $\vec{r}_\perp$ at each step and tries to find an unsafe combination using the result of Corollary 15. Note that for any non-zero vector $\vec{r}_\perp$ in two dimensions there are only two vectors which we need to check. Although there are infinitely many vectors $\vec{v}$ which might satisfy conditions of Corollary 15, the conditions we are checking are preserved if we multiply $\vec{v}$ by a positive scalar.

5. DISCRETE SCHEDULABILITY

In this section we discuss the *discrete schedulability problem*, in which a scheduler can only make decisions at integer multiples of a specified *clock period* Δ and the environment has finitely many choices of rates. Formally, given a MMS $\mathcal{H}$, a closed convex polytope S as safety set, an initial state $x_0 \in S$, the discrete schedulability problem is to decide if there exists a winning strategy of the scheduler where the time delays are multiples of Δ.

THEOREM 16. *Discrete schedulability problem is EXPTIME-complete.*

PROOF. EXPTIME-membership of the problems is shown via discretization of the state space of $\mathcal{H}$. Since the set S is given as a bounded polytope, the size of the discretization can be shown to be at most exponential in the size of $\mathcal{H}$ and Δ, and since the safety games on a finite graph can be solved in P, EXPTIME membership follows. The hardness can be proved by a reduction from the countdown games [7]. For space constraints we give the proof in Appendix B.5. □

We turn the discrete schedulability problem to an optimization problem, by asking to find supremum of all Δ for which the answer to the discrete schedulability problem is yes. We prove the following, which also improves a result of [1] where only an approximation algorithm was given.

THEOREM 17. *Given a MMS $\mathcal{H}$, a closed convex polytope S and an initial state $\overline{x}_0$, there is an exponential time algorithm which outputs the maximal Δ for which the answer to the discrete schedulability problem is Yes. For a CMS the algorithm can be made to run in polynomial space.*

PROOF (SKETCH). We exploit the fact that as the clock period Δ increases, all the points of the discretization move continuously towards infinity, except for the initial point. This further implies that for Δ to be maximal, there must be a point of the discretization which lies on the boundary of S, since otherwise we could increase Δ by some small number, while preserving the existence of a safe scheduler. By using a lower bound on Δ from Section 3 (obtained as a by-product of the construction of a dynamic strategy), there are only exponentially many candidates for such points, which gives us exponentially many candidates for maximal Δ to consider, and we can check each one by Theorem 16. For the PSPACE bound we don't enumerate the points, but guess them nondeterministically in polynomial space, and utilize [1, Theorem 10] instead of Theorem 16. Full details of the proof are given in Appendix B.6. □

6. CONCLUSION

We investigated systems that comprise finitely many real-valued variables whose values evolve linearly based on a rate vector determined by strategies of the scheduler and the environment. We studied an important schedulability problem for these systems, with application to energy scheduling, that asks whether scheduler can make sure that the values of the variables never leave a given safety set. We showed that when the safety set is a closed convex polytope, existence of non-Zeno winning strategy for scheduler is decidable for any arbitrary starting state. We also showed how to construct such a winning strategy. On complexity side, we showed that the schedulability problem is co-NP complete in general, but for the special case where the system has only two variables, the problem can be decided in polynomial time. Directions for future research include investigation of schedulability problem with respect to more expressive higher-level control objectives including temporal-logic based specification and bounded-rate multi-mode systems with reward functions.

7. REFERENCES

[1] R. Alur, A. Trivedi, and D. Wojtczak. Optimal scheduling for constant-rate mulit-mode systems. In *HSCC*, 2012.

[2] F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747 – 1767, 1999.

[3] T. A. Henzinger. The theory of hybrid automata. In *LICS 96*, 1996.

[4] T. A. Henzinger, B. Horowitz, and R. Majumdar. Rectangular hybrid games. In *CONCUR 99*, pages 320–335. Springer, 1999.

[5] T. A. Henzinger and P. W. Kopke. Discrete-time control for rectangular hybrid automata. *TCS*, 221(1-2):369–392, 1999.

[6] M. Heymann, L. Feng, G. Meyer, and S. Resmerita. Analysis of zeno behaviors in a class of hybrid systems. *IEEE Trans. on Auto. Ctrl.*, 50:376–383, 2005.

[7] M. Jurdziński, J. Sproston, and F. Laroussinie. Model checking probabilistic timed automata with one or two clocks. *LMCS*, 4(3):12, 2008.

[8] J. Le Ny and G. J. Pappas. Sequential composition of robust controller specifications. In *ICRA*, 2012.

[9] J. Liu, N. Ozay, U. Topcu, and R. Murray. Switching protocol synthesis for temporal logic specifications. In *American Control Conference*, 2012.

[10] A. Maitra and W. Sudderth. Finitely additive stochastic games with borel measurable payoffs. *International Journal of Game Theory*, 27(2), 1998.

[11] T. X. Nghiem, M. Behl, R. Mangharam, and G. J. Pappas. Green scheduling of control systems for peak demand reduction. In *IEEE CDC*, December 2011.

[12] T. X. Nghiem, M. Behl, G. J. Pappas, and R. Mangharam. Green scheduling: Scheduling of

control systems for peak power reduction. *2nd International Green Computing Conference*, July 2011.
[13] L. T. X. Phan, I. Lee, and O. Sokolsky. Compositional analysis of multi-mode systems. In *ECRTS*, 2010.

APPENDIX

A. ABSENCE OF STATIC STRATEGIES

PROPOSITION 18. *For a given starting state in the interior of the safety set S, the scheduler has a static winning strategy in a* BMS $\mathcal{H} = (M, n, \mathcal{R})$ *iff there is* $M' \subseteq M$ *such that* $|\mathcal{R}(m)| = 1$ *for all* $m \in M'$, *and the* CMS $H = (M', n, R)$ *is safe, where* $R(m)$ *is the unique rate of* $\mathcal{R}(m)$.

PROOF. The "if" direction is trivial. To show the "only if" direction we show that if there is no CMS subsystem of $\mathcal{H}$ for which there is a safe and non-Zeno schedule, then there is no static winning schedule for schedulability objective. Let $\sigma = (m_1, t_1), (m_2, t_2), \ldots$ be a static scheduler.

Assume there is $m \in M$ with two different rates $\vec{r}_a$ and $\vec{r}_b$ such that $\sum_{i:m_i=m} t_i = \infty$. We then define two strategies for the environment, π_a and π_b which for a mode m always pick a rate $\vec{r}_a$ and $\vec{r}_b$, respectively. After the first k steps, the point reached under σ_b is equal to

$$\overline{x}_b = \overline{x}_a + (\vec{r}_b - \vec{r}_a) \cdot \sum_{i \leq k : m_i = m} t_i$$

Hence, the points $\overline{x}_a$ and $\overline{x}_b$ will be arbitrarily far apart for large enough k, since the safety set is bounded, one of the strategies π_a and π_b must ensure that a point outside is left eventually.

On the other hand, assume all modes m which have two different rates satisfy that $\sum_{i:m_i=m} t_i$ is finite. Let M' be all such modes, and let $d_1 := \|\vec{r}\| \cdot \sum_{i:m_i \in M'} t_i \leq \infty$ where $\vec{r}$ is the rate with the maximal Euclidean norm which occurs in $\mathcal{H}$. Intuitively, d is the upper bound on the change of the values of variables caused by using the modes of M'. Let d_2 be the diametre of S, and let p be the Euclidean distance of the initial point $\overline{x}_0$ from the boundary of S. We define a strategy

$$\sigma' = (m'_1, t'_1 \cdot \frac{p}{d_1 + d_2}), (m'_2, t'_2 \frac{p}{d_1 + d_2}), \ldots$$

where $(m'_1, t'_1), (m'_2, t_2), \ldots$ is the sequence $(m_1, t_1), (m_2, t_2) \ldots$ from which we omit all the tuples which have a mode from M' in the first component. The strategy σ' is safe for S and further shows that there is a safe CMS subsystem, which is a contradiction. $\square$

B. OMITTED PROOFS

B.1 Proof of Lemma 8

Let $\overline{x}$ and $\overline{y}$ be points satisfying $S|_I$. Assume $\overline{x}$ is safe with a strategy σ, and let σ_d be a strategy for a controller defined as follows: Let $\varrho = \langle \overline{y}_0, (m_1, t_1), \vec{r}_1, \overline{y}_1, \ldots \overline{y}_k \rangle$ where $\overline{y}_i = \overline{y} + \overline{y}'_i$ for some $\overline{y}'_i$, be a history, and let $(\vec{r}, t)$ be a decision of σ on $(\overline{x}_0, m_1, t'_1, \vec{r}_1, \overline{x}_1, \ldots \overline{x}_k)$, where $\overline{x}_i = \overline{x} + \overline{y}'_i \cdot d$ and $t'_i = t_i \cdot d$. The strategy σ_d chooses $(\vec{r}, t/d)$ in π. Intuitively, σ_d mimics the decision of σ, but it assumes the starting point is $\overline{y}$ rather than $\overline{x}$, and it scales the time intervals down by d, hence making sure that only points closer to $\overline{y}$ can be reached. For this reason it suffices to take large enough d to make sure that σ_d is safe. For example, we can put $d = (\sup_{\overline{x}' \models S|_I} \|\overline{x}, \overline{x}'\|)/(\inf_{\overline{y}' \models S|_{I'}, I \subseteq I'} \|\overline{y}, \overline{y}'\|)$.

Similar arguments can be made for the second part of the lemma, i.e. any strategy safe for a point satisfying $S|_{I'}$ can be scaled to a strategy safe for a point satisfying $S|_{I'}$.

B.2 Proof of Proposition 11 (correctness of construction)

We show that the construction proposed in the proof of Proposition 11 in the main body is correct. We show that there is a satisfying assignment for φ iff there exists an unsafe instance of $\mathcal{H}_\phi$.

- Now let us suppose that there is an unsafe combination $\{\vec{r}_i | \vec{r}_i \in m_i,\ 1 \leq i \leq |M|\}$. Then for every rate $\vec{r}_i$ which contains 1 at i-th position assign true to the variable x_i, and for every rate $\vec{r}_i$ which contains -1 at i-th position assign false to the variable x_i. Note that no variable would be assigned both true and false since if two vectors $\vec{r}'$ and $\vec{r}''$ are chosen which go to the opposite direction, then every $\vec{v}$ which satisfies $\vec{v} \cdot r' > 0$ also satisfies $\vec{v} \cdot r'' < 0$, and vice versa, which means that the combination is not unsafe. Further, observe that the assignment is satisfying, because for every clause c_j we have that if $\vec{r}_j$ contains 1 at i-th position, then c_j contains the literal x_i which is satisfied, and if $\vec{r}_j$ contains -1 at i-th position, then c_j contains the literal $\neg x_i$ which is satisfied. Hence there is at least one true literal in each clause and thus the formula ϕ is satisfiable.
- To prove the other direction, assume that there is a satisfying assignment to ϕ, then choose one true literal from each clause and consider the corresponding rate vector for each mode. Note that there would be no two vectors along one axis with different directions since $\neg x_i$ and x_i can not be true at the same time. Therefore we have k vectors along $1 \leq d \leq n$ axises where each two vectors are either same or perpendicular. This set of rate vectors will be unsafe since there exists a $\vec{v}$ with strictly positive dot product with all of them: We build vector $\vec{v}$ such that each i-th entry of vector $\vec{v}$ is 1 (resp. -1), if there are some vectors whose i-th entry is 1 (resp. -1), and zero otherwise. The product of v with any vector from the combination is equal to 1, and hence greater than zero. □

B.3 Proof of Corollary 12

To prove this corollary we show that if there is an unsafe instance of BMS $\mathcal{H}$ then there is an unsafe instance of corresponding extreme-rate MMS $Ext(\mathcal{H})$. With this observation, the corollary then follows from Proposition 11. Assume m is a mode in the bounded-rate multi-mode system $\mathcal{H}$ with extreme rate vectors $\{r_1^*, ..., r_k^*\}$. First we show that if there is a rate vector $r \in m$ and a rate vector v such that their dot product is positive, i.e. $v.r > 0$, then there exists at least one extreme rate vector r_i^* which makes angle less than 90 with v, i.e. $v.r_i^* > 0$. We can write $r = \sum \lambda_i r_i^*$ where $\sum \lambda_i = 1$. Assume vector v has positive dot product with r, $v.r > 0$. Assume for the purpose of contradiction that $\forall i\ v.r_i^* \leq 0$, which is a contradiction because then we have $v.\lambda_i r_i^* \leq 0 \to \sum v.\lambda_i r_i^* \leq 0 \to v.\sum \lambda_i r_i^* = v.r \leq 0$. Thus if there is an unsafe instance of BMS, for each mode we can choose a extreme rate such that the corresponding extreme-rate instance is unsafe.

B.4 Proof of Lemma 14

If $|R| \leq 1$, then the claim is immediate. Assume R contains at least two rates.

Let us start with $\Rightarrow$. Intuitively, we keep changing $\vec{v}$ until it becomes perpendicular to some vector in R, and then we show that the vector obtained in this way satisfies the desired properties. Formally, pick a vector $\vec{w}$ such that $\vec{w} \cdot \vec{r}_0 = 0$. Find a maximal $\alpha \in [0, 1)$ such that for the vector $\vec{v}_\alpha := \alpha \cdot \vec{v} + (1-\alpha) \cdot \vec{w}$ there is a vector in R perpendicular to $\vec{v}_\alpha$. Such α must exists, since at least for $\vec{v}_0 = \vec{w}$ we have $\vec{r}_0$ perpendicular. We claim $\vec{v}_\alpha$ is our vector $\vec{u}$, and we put $\vec{r}_\perp$ any vector of R perpendicular to it. First, observe that there is no $\vec{r} \in R$ such that $\vec{u} \cdot \vec{r} < 0$. If this was the case, then $\alpha \cdot \vec{v} \cdot \vec{r} + (1-\alpha) \cdot \vec{w} \cdot \vec{r} < 0$ and since $\vec{v} \cdot \vec{r}$ is positive, we could have picked $\alpha' > \alpha$ for which $\vec{v}_{\alpha'} \cdot \vec{r} = \alpha' \cdot \vec{v} \cdot \vec{r} + (1-\alpha') \cdot \vec{w} \cdot \vec{r} = 0$ (for the same $\vec{r}$ as before), contradicting the maximality of α. Now for any vector $\vec{r} \in R$ such that $\vec{u} \cdot \vec{r} = 0$, if $\vec{r} \neq p\vec{r}_\perp$ for any $p > 0$, then $\vec{r} = p\vec{r}_\perp$ for some $p < 0$. But since $\vec{r}_\perp \cdot \vec{v} > 0$, we get $\vec{r} \cdot \vec{v} = p \cdot (\vec{r}_\perp \cdot \vec{v}) < 0$, which is a contradiction with properties of $\vec{v}$.

In the other direction, if there are no $\vec{r} \in R$ such that $\vec{u} \cdot \vec{r} > 0$, we can just put $\vec{v}$ to be an arbitrary element of R. Otherwise, we show that we can obtain $\vec{v}$ if we make a small enough change to $\vec{u}$. Fix some $\vec{r}_\perp$ where $\vec{u} \cdot \vec{r}_\perp = 0$. Let $\tau := \min_{\vec{r} \in R : \vec{u} \cdot \vec{r} > 0} \vec{u} \cdot \vec{r}$ be the minimal *positive* dot product of $\vec{u}$ with vectors of R, and let $\kappa := \min_{\vec{r} \in R} \vec{r}_\perp \cdot \vec{r}$ be the minimal (possibly negative) dot product of $\vec{r}_\perp$ with vectors of R. Set $\vec{v} = \vec{u} + \frac{\tau}{2 \cdot (|\kappa|+1)} \vec{r}_\perp$. For every $\vec{r} \in R$, we have $\vec{v} \cdot \vec{r} = \vec{u} \cdot \vec{r} + \frac{\tau}{2 \cdot (|\kappa|+1)} \vec{r}_\perp \cdot \vec{r}$ which is positive, because: (i) if the left summand is 0, then the right summand is positive because $\vec{r}_\perp \cdot \vec{r} > 0$, and (ii) if the left summand is positive, then it is at least τ and the right summand is at least $\frac{\tau}{2 \cdot (|\kappa|+1)} \kappa \geq -\frac{\tau}{2}$, and so the sum is positive.

B.5 Proof of Theorem 16 (the hardness part)

A countdown game is a tuple $\mathcal{G} = (N, T, n_1, B_1)$ where

- $N = \{n_1, n_2, \dots, n_d\}$ is a finite set of nodes;
- $T \subseteq N \times \mathbb{N}_{>0} \times N$ is a set of transition; and
- $(n_1, B_1) \in N \times \mathbb{N}_{>0}$ is the initial configuration.

From any configuration $(n, B) \in N \times \mathbb{N}_{>0}$, first player 1 chooses a number $k \in \mathbb{N}_{>0}$, such that $k \leq B$ and there exists some $(n, k, n') \in T$, and then player 2 chooses a transition $(n, k, n'') \in T$ labeled with that number. Note that there can be more than one such transition. The new configuration then transitions to $(n'', B-k)$. Player 1 wins a play of the game when a configuration $(n, 0)$ is reached, and loses (i.e., player 2 wins) when a configuration (n, B) is reached in which player 1 is stuck, i.e., for all transitions $(n, k, n') \in T$, we have $k > B$.

For a countdown game (N, T, n_1, B_1) we define a BMS $\mathcal{H}$, a safety set S and an initial state $\overline{x}$ such that there is a safe scheduler in Σ_Δ for $\Delta = 1$ iff player 1 has a winning strategy in the countdown game. W.l.o.g we assume that when $(n, k, n') \in T$, then $n \neq n'$, and also we assume that the initial state is (n_1, B_1) and there is no node n and k such that $(n, k, n_1) \in T$.

The BMS $\mathcal{H}$ has $d+1$ variables. The intuition is that the value of the first variable corresponds to the value of the counter, while $(i+1)$th variable is equal to 1 if the game is in node n_i, and 0 otherwise.

For all $n, k \in N \times \mathbb{N}_{>0}$ such that there is $(n, k, n') \in T$ for some n', we add a mode (n, k) to $\mathcal{H}$. For all $(n_i, k, n_j) \in T$, we add the rate r to the mode (n_i, k) such that the first component of r is $-k$, the $(i+1)$th component is -1 and $(j+1)$th component is 1. All other components of r are zero. We further add modes m_i for $3 \leq i \leq d+1$ which contain the unique rate with B_1 in the first component, 1 in the second component, and -1 in i-th component. All other components of this rate are zero.

The safety set S is defined so that the only points with integer values are exactly $(i_1, \ldots, i_{d+1})$, where $0 \le i_1 \le B_1$, and exactly one of $i_2, \ldots, i_{d+1}$ is 1, while the others are 0. Such safety set can be defined using equations

$$\begin{array}{rll} x_1 & \le B_1 & \\ \sum_{i=2}^{d+1} x_i & \le 1 & \\ x_i & \ge 0 & \quad \text{for } 1 \le i \le d+1 \end{array}$$

Now we claim that the system is schedulable from the point $(B_1, 1, 0, 0, \ldots, 0)$ iff player 1 has a winning strategy in the countdown game. The intuition is that the winning strategy for player 1 in the countdown game directly gives a strategy for the scheduler in $\mathcal{H}$ such that a point is reached which has zero in the first component, and zeros everywhere else except for some i-th component. Then the scheduler uses the mode m_i, which leads back to the initial state and then he can repeat the same strategy. On the other hand, if player 2 has a winning strategy in the countdown game, this strategy can be used to get to a state from which the scheduler has no chance but to leave the safety set (which corresponds to not having any choices in the countdown game).

B.6 Proof of Theorem 17

In this section we show how to solve the following problem: given a MMS $\mathcal{H} = (M, n, \mathcal{R})$, a convex polytope S and an initial state $\overline{x} \in S$, find the maximal number Δ_{max} such that there is a winning strategy for the scheduler which only takes decisions at times $i \cdot \Delta_{max}$ where $i \in \mathbb{N}$. Formally, let Σ_Δ denote the set of strategies for the scheduler which schedule in multiples of Δ. Then we wish to find a supremum, over all Δ, such that there is a safe scheduler in Σ_Δ.

Let R be the set of all possible rate vectors of $\mathcal{H}$. Note that since $\mathcal{H}$ is a MMS, the set R is finite.

Let $discr(\Delta)$ be the points reachable from $\overline{x}$ when using a scheduler from Σ_Δ. All such points are equal to $\overline{x} + \sum_{\vec{r} \in R} i_{\vec{r}} \cdot \Delta \cdot \vec{r}$ for some $i_{\vec{r}} \in \mathbb{N}$. This implies that the set $discr(\Delta) \cap S$ is finite.

The intuition of our algorithm is the following. Every strategy from Σ_Δ can be seen as a function which rather than observing and choosing time delays observes and chooses the number of time periods (multiples of Δ) elapsed. Using this abstracted view of strategies, every strategy in Σ_Δ corresponds to a strategy in Σ'_Δ which differs only in the length of the time period. It can be shown that there is a correspondence of points reachable under these two strategies. Seeing the points of $discr(\Delta)$ as a "grid", the points of $discr(\Delta')$ are obtained by stretching (if $\Delta' > \Delta$) or squeezing (if $\Delta' < \Delta$) this grid. It follows that for a Δ to be maximal, there must be a point in $discr(\Delta)$ which lies on the boundary of S, since otherwise the grid $discr(\Delta)$ can be stretched to some $discr(\Delta')$ where $\Delta' > \Delta$, preserving the existence of a safe scheduler. Exploiting this property together with the fact that we already know a lower bound on Δ_{max}, we get only finitely many candidates for maximal Δ, and we can check in each of them whether a safe scheduler exists using Theorem 16. Our algorithm is presented as Algorithm 5.

Let us now prove the correctness of the algorithm. Clearly the algorithm terminates in exponential time since the "foreach" loop is executed only exponentially many times at most, and each of the respective lines can be executed in exponential time. Hence, we only need to show that the result returned by the algorithm is correct.

Algorithm 5: algorithm computing Δ_{max}

Input: schedulable $\mathcal{H}$, safety set S given as $Ax \le b$, point $\overline{x} \in S$
Output: Δ_{max}

```
Let Γ be the lower bound on Δ_max;
Compute discr(Γ) ∩ S;
Δ_max := Γ;
foreach ȳ = x̄ + Σ_{r⃗∈R} i_r⃗ Δ · r⃗ ∈ discr(Γ) ∩ S do
    maximise Δ subject to A · (x̄ + Σ_{r⃗∈R} i_r⃗ Δ · r⃗) ≤ b if
    Σ_Δ contains a safe scheduler and Δ > Δ_max then
        Δ_max := Δ
    return Δ_max
```

We first introduce some technical notation to capture the intuition of correspondence between points of different discretisations. Define a bijection $g_{\Delta,\Delta'}$ between $discr(\Delta)$ and $discr(\Delta')$ that to a point $\overline{x} + \Delta \cdot \sum_{\vec{r} \in R} i_{\vec{r}} \cdot \vec{r}$ where $i_{\vec{r}} \in \mathbb{N}$ assigns the point $\overline{x} + \Delta' \cdot \sum_{r \in R} i_{\vec{r}} \cdot \vec{r}$. Intuitively, this function pairs the corresponding points on the "grids" given by $discr(\Delta)$ and $discr(\Delta')$. Note that $g_{\Delta,\Delta'}$ is well defined and does not depend on the choice of $i_{\vec{r}} \in \mathbb{N}$ which represent the point and can be non-unique. Indeed, if

$$\overline{x} + \Delta \cdot \sum_{\vec{r} \in R} i_{\vec{r}} \cdot \vec{r} = \overline{x} + \Delta \cdot \sum_{\vec{r} \in R} i'_{\vec{r}} \cdot \vec{r}$$

for some $i_{\vec{r}}, i'_{\vec{r}} \in \mathbb{N}$, then $\sum_{\vec{r} \in R} i_{\vec{r}} \cdot \vec{r} = \sum_{\vec{r} \in R} i'_{\vec{r}} \cdot \vec{r}$ and hence also $\overline{x} + \Delta' \cdot \sum_{\vec{r} \in R} i_{\vec{r}} \cdot \vec{r} = \overline{x} + \Delta' \cdot \sum_{\vec{r} \in R} i'_{\vec{r}} \cdot \vec{r}$.

The following lemma essentially says that when we enlarge the length of a time period, the set of points on the corresponding grid that are within S can only get smaller.

Lemma 19. *Let $\Delta \ge \Delta'$. Then $g_{\Delta',\Delta}(discr(\Delta') \cap S) \supseteq discr(\Delta) \cap S$.*

Proof. Follows because S is closed, convex and contains $\overline{x}$. □

The following lemma intuitively says when we can increase the time period while preserving the existence of a safe scheduler.

Lemma 20. *Let $\Delta \ge \Delta'$ be such that*

$$discr(\Delta) \cap S = g_{\Delta',\Delta}(discr(\Delta') \cap S)$$

and assume that there is a safe scheduler in $\Sigma_{\Delta'}$. Then there is a safe scheduler in Σ_Δ.

Proof. Using $g_{\Delta',\Delta}$, we can define a function $h_{\Delta',\Delta}$ from $\Sigma_{\Delta'}$ to Σ_Δ to capture our intuition of strategies that differ only on the length of a time period as follows. Given $\sigma \in \Sigma_{\Delta'}$ and a history

$$\langle x_0, (m_1, i_1 \cdot \Delta'), r_1, \ldots x_k \rangle$$

we put

$$\begin{aligned} h_{\Delta',\Delta}(\sigma)(\langle g_{\Delta',\Delta}(x_0), (m_1, i_1 \cdot \Delta), r_1, \ldots g_{\Delta',\Delta}(x_k) \rangle) \\ = \sigma(\langle x_0, (m_1, i_1 \cdot \Delta'), r_1, \ldots x_k \rangle) \end{aligned}$$

Now it is easy to prove by induction that if the set of points that are reachable under $\sigma \in \Sigma_{\Delta'}$ is X, then the set of points reachable by $h_{\Delta',\Delta}(\sigma) \in \Sigma_\Delta$ is equal to $g_{\Delta',\Delta}(X)$. □

Now we are ready to proceed with the proof of the correctness of the algorithm. Let Δ_{max} be the actual solution, and let $\bar{\Delta}_{max}$ be the returned number. We know that $\bar{\Delta}_{max} \leq \Delta_{max}$, since the algorithm ensures that there is a safe scheduler in $\Sigma_{\bar{\Delta}_{max}}$. To prove $\bar{\Delta}_{max} \geq \Delta_{max}$, it suffices to show that there is a safe scheduler in $\Sigma_{\Delta_{max}}$, and that Δ_{max} is found for some $\overline{y}$ at line 4 of Algorithm 5.

To show that there is a safe scheduler in $\Sigma_{\Delta_{max}}$, let

$$X := \bigcap_{\Delta < \Delta_{max}} g_{\Delta,\Delta_{max}}(discr(\Delta) \cap S).$$

We have $X = discr(\Delta_{max}) \cap S$. The inclusion $\supseteq$ follows by Lemma 19; the inclusion $\subseteq$ follows by the fact that S is closed and the fact that as Δ gets arbitrary close to Δ_{max}, the points $\overline{y} \in discr(\Delta)$ get arbitrary close to $g_{\Delta,\Delta_{max}}(\overline{y})$. By Lemma 19 and because $discr(\Delta) \cap S$ is finite for all $\Gamma \leq \Delta < \Delta_{max}$, there is $\Delta < \Delta_{max}$ such that $g_{\Delta,\Delta_{max}}(discr(\Delta) \cap S) = X$, and by definition of Δ_{max} there is a safe scheduler in Σ_{Δ}. Finally by Lemma 20 there must be a safe scheduler $\sigma \in \Sigma_{\Delta_{max}}$.

Now suppose that Δ_{max} is not a solution to any of the linear programs executed on line 4. For each $\overline{y} \in discr(\Gamma) \cap S$, let $\Delta_{\overline{y}}$ be the solution to the linear program for $\overline{y}$. Let P be the set of all $\overline{y} \in discr(\Gamma) \cap S$ satisfying $g_{\Gamma,\Delta_{max}}(\overline{y}) \in discr(\Delta_{max}) \cap S$. Define $\Delta = \min_{\overline{y} \in P} \Delta_{\overline{y}}$. We have $\Delta > \Delta_{max}$, since if $\Delta = \Delta_{max}$ then Δ_{max} would be the solution to the linear program for the point $\overline{y}$ which realises the minimum, and if $\Delta < \Delta_{max}$ then $g_{\Gamma,\Delta_{max}}(\overline{y}) \notin discr(\Delta_{max}) \cap S$. In addition, $g_{\Delta_{max},\Delta}(discr(\Delta_{max}) \cap S) = discr(\Delta) \cap S$ which by Lemma 20 implies that there is a safe scheduler in Σ_{Δ}, contradicting the maximality of Δ_{max}.