Optimal linear separation of the safe and unsafe subspaces of sequential resource allocation systems as a set-covering problem: algorithmic procedures and geometric insights

Spyros Reveliotis and Ahmed Nazeem
School of Industrial & Systems Engineering
Georgia Institute of Technology
{spyros@isye.,anazeem@}gatech.edu

Abstract

A recent line of work has posed the design of the maximally permissive deadlock avoidance policy for a particular class of sequential resource allocation systems as a linear classification problem. It has also identified a connection between the classifier design problem addressed by it and the classical set-covering problem that has been studied in Operations Research and Computer Science. This paper seeks to explore and formalize further this connection, in an effort to (i) develop novel insights regarding the geometric and combinatorial structure of the classifier design problem mentioned above; (ii) set an analytical base for the development of additional customized and computationally (more) efficient algorithms for its solution; and (iii) identify necessary and sufficient conditions, and the corresponding computational tests, for the effective application and the extension of the representational results in the aforementioned work to broader classes of resource allocation systems.

Keywords: Discrete Event Systems, Deadlock Avoidance, Classification Theory, Set-Covering Problem, Polyhedral Theory

1 Introduction

In many contemporary applications, ranging from automated manufacturing systems, to intelligent transportation systems, to internet-based workflow management systems, and more recently to the parallel programs that are developed for the emerging multi-core computer architectures, the underlying control problems boil down to the effective and efficient management of the allocation of some limited system resources to a number of concurrently executing processes. These system resources are reusable, and the contesting processes acquire and release them in a staged manner, where each process stage is being associated with a multi-set of resources that is necessary for its effective support. The structure of the multi-sets that express the resource requests of the different stages, and also the sequential logic that defines the process transitions among their processing stages can be quite arbitrary, while in many cases, it is further required that a process must first secure the resource differential that is necessary for the support of its subsequent stage before it releases any currently held resources that are not needed for the next stage. This hold-while-wait effect, when combined with the arbitrary structure of the logic that defines the process resource allocation requests, can give rise to circular waiting patterns where a set of executing processes is waiting upon each other for the release of resources that are necessary for their further advancement. The resulting situation is characterized as a *deadlock* in the relevant literature, and the need for its effective resolution becomes especially prominent as the aforementioned applications migrate to operational modes that seek to combine the advantages and efficiencies of automation with the operational efficiencies and the flexibility that result from extensive concurrency and choice.

In fact, the problem of deadlock prevention / avoidance in the aforementioned operational contexts is a well established problem in the controls systems literature. From a methodological standpoint, this problem is typically addressed by modeling the resource allocation function that takes place in the aforementioned environments through an abstraction that is known as a *resource allocation system (RAS)* [24], and subsequently analyzing the RAS behavior with respect to (w.r.t.) deadlock by using concepts and techniques borrowed from qualitative Discrete Event Systems (DES) theory [4]. In particular, due to the extensive randomness that is typically present in the execution of the aforementioned processes, the relevant research community has come to the conclusion that robust solutions to the considered deadlock avoidance problem cannot be based on the control of the specific timing of the execution of the various resource allocation events but only on the control of their logical sequencing, in the sense that this sequencing is defined by the representational framework of formal languages [4, 13]. Ideally, the imposed control logic – known as the *deadlock avoidance policy (DAP)* – should allow all the resource allocation sequences that can be extended to a terminating sequence for all the enacted processes. Such a DAP is characterized as *maximally permissive* in the relevant literature, and essentially it constitutes a “filter” that extracts from the original feasible behavioral space of the underlying RAS the maximal admissible sub-space, where admissibility is defined on the basis of deadlock-free behavior. Once deadlock-free operation has been secured, a second level of control must be applied that will bias the system admissible behavior in a way that it aligns with certain, typically time-related, performance objectives. [1]

[1] From a conceptual standpoint, this second control level is closer to the more traditional *scheduling* problems that have been considered for the aforementioned application domains.

Given the above description of the role and function of the maximally permissive DAP, it is not hard to see that a sufficient statistic for the effective characterization of this policy is the number of process instances that execute each of the processing stages that are supported by the system, across all of the system process types; this information determines completely the current allocation of the system resources and all the possible ways for its future evolution, and in more technical terms, it constitutes the RAS *state*. The state space that results from this definition of the RAS state is further endowed with a transition structure that models the possible evolution of the system processes among their processing stages, while respecting the availability of the system resources and the aforementioned resource allocation protocol. In this way, the underlying RAS dynamics can be formalized into a state automaton [4]. This state automaton is also finite, since the number of processes that can execute concurrently any given processing stage is limited by the finite availability of the supporting resources.

Once a *finite state automaton (FSA)* - based representation of the RAS dynamics has been obtained, a formal characterization of the corresponding maximally permissive DAP can be obtained through basic, search-based enumerative techniques that assess the co-reachability w.r.t. the target empty state of any given RAS state in the underlying state transition diagram (STD); these techniques can be traced in any text on DES supervisory control theory, like [4]. In the relevant terminology of the RAS supervisory control theory, the subspace of co-accessible states, that is admitted by the maximally permissive DAP, is known as the RAS *safe* space, while its complement is characterized as the *unsafe* space. Furthermore, assuming the availability of a mechanism that can resolve the safety of any given RAS state, the maximally permissive DAP can be implemented through a single-step-lookahead scheme that simply tests the safety of the state that results from a tentative process transition, and admits this transition only if the resulting state is safe. However, a practical complication of such a control scheme arises from the fact that for most of the RAS structure that arises in the aforementioned application contexts, the corresponding decision problem of state safety is NP-complete [1, 12, 23]. Faced with this negative result, the relative research community had sought for a long time to circumvent the implied computational limitations either (i) by seeking to identify special RAS structure that enables the resolution of state safety in a polynomial manner w.r.t. any parsimonious representation of the underlying RAS, or (ii) by compromising with suboptimal solutions that substitute the safety property with surrogate properties that guarantee deadlock-free behavior and are polynomially evaluated on any given RAS state;[2] comprehensive expositions of these results can be found in [24, 26, 15].

Yet, more recently, and under the increasing pressures for process concurrency and operational flex-

[2] In more technical terms, these alternative properties are able to recognize a strongly connected component of the underlying RAS safe subspace that also contains the target empty state, but not necessarily the maximal one. Dijkstra's Banker's algorithm [9] and various more recent implementations of it in the context of the considered RAS [2, 14, 10] are some well known policies of this type.

ibility that was mentioned in the opening paragraph, there has been a new, more aggressive approach towards the implementation of the maximally permissive DAP in the aforementioned application contexts, that seeks the real-time deployment of this policy, in spite of the established NP-hardness. This new approach is motivated and enabled by the following remarks: From the above description of the RAS (optimal) deadlock avoidance problem, it is clear that the maximally permissive DAP acts as a *"classifier"* that dichotomizes the RAS behavioral space into its safe and unsafe subspaces. In particular, this state classification capability is at the core of the one-step-lookahead implementation of the maximally permissive DAP that was discussed in the previous paragraph. And while it is known that assessing the state safety is an NP-complete task, it might be possible to pre-compute the safety of the various states through an expensive "off-line" computation that uses the classical FSA-based representation of the RAS dynamics, and subsequently re-hash the obtained results in an alternative representation that will enable the "on-line" safety assessment of any given RAS state through a streamlined, more efficient computation. The construction of this last representation can be based on the interpretation of the maximally permissive DAP as a classifier, and draw upon concepts and results borrowed from classification theory [22]. Specific implementations of these ideas can be traced in the works of [20, 17, 19, 18, 8, 5].

A pertinent selection of the classifier structures that can effectively represent the RAS state safety concept, can be based on (i) the definition of the RAS state as a vector of nonnegative integers, and (ii) the finiteness of the safe and unsafe subspaces. More specifically, it is well known that any pair of finite vector sets from a Euclidean space can be effectively separated by a two-tier structure of linear inequalities known as a "(generalized) committee machine" [22], where the first set of inequalities is applied directly upon the classified vector and the second set is applied upon the set of indicator variables that express the satisfaction of the first layer of inequalities. Furthermore, the binary nature of the vectors that are processed by the second layer of inequalities, implies that one can substitute this second layer of inequalities of a committee machine by a Boolean function that recognizes the binary images of the accepted input vectors that are generated by the first layer of inequalities. In the sequel, we shall refer to these two classifier classes as *non-linear* classifiers. Finally, in [20] it is also shown that in the particular case where the underlying RAS state space is binary, effective separation of the safe and unsafe subspaces can be attained by a single layer of linear inequalities. Such a classifier will be characterized as *linear*.

In the sequel, we shall focus primarily on the particular class of linear classifiers. As explained in the previous paragraphs, for any given RAS instance, we are interested in the representation of the corresponding maximally permissive DAP by a *(structurally) minimal* element of the considered class of classifiers, i.e., by a classifier from the considered class that will characterize correctly the safety of the various RAS states while minimizing the involved "on-line" computational cost. For linear classifiers, this "on-line" cost minimization can be achieved by minimizing the number of the linear inequalities involved, and the relevant design problem can be posed as a mixed integer programming (MIP) formulation [20]. The tractability of this formulation is established upon a monotonicity property that is presented by RAS state safety in the context of the considered RAS classes, and it implies that a RAS state that is component-wise larger than an unsafe state, is also unsafe. In [20] it is shown that thanks to this monotonicity property, one can design a linear classifier that classifies correctly all the *maximal* safe states and the *minimal* unsafe states of the considered RAS, and the resulting classifier will classify correctly all the other RAS states as well, provided that all the classifier coefficients are restricted to be nonnegative. Furthermore, as long as the considered classification problem admits linear classification, there will exist a minimal linear classifier with nonnegative coefficients; hence, the aforementioned restriction impacts neither the feasibility nor the optimality of the considered optimization problem. In addition, in [20] it has also been shown that in the reduced classification problem, one can project away the state vector coordinates that are identically priced to zero in the set of minimal unsafe states, and the obtained classifier will still generalize correctly when applied on the original state space.[3] Finally, the results of [20] have also provided a heuristic for the synthesis of the sought classifier. This heuristic seeks to design the sought inequalities one at a time, and it can be applied when the MIP formulation that characterizes the optimal classifier remains an intractable proposition. From a more analytical standpoint, the logic underlying this heuristic has been motivated by and analyzed through a perceived similarity between the classifier design problem and the minimal set covering problem [25] that has been studied by the Operations Research (OR) and the Computer Science (CS) communities.

The work presented in this paper seeks to explore further the problem of the representation of the maximally permissive DAP through a minimal set of linear inequalities, and its connection to the minimal set covering problem that was identified in [20]. In particular, the main results of this work give a complete formal characterization of this connection, by providing a systematic reduction of the (simplified) classifier design problem of [20] to the classical set covering problem. Hence, at a first level, the results provided in this work can be perceived as an alternative algorithmic procedure for the resolution of the optimal classifier design problem that is addressed in [20]. Even more importantly, however, the analysis that provides the aforementioned reduction also reveals additional important structure regarding the geometry of the classifier design problem addressed in the aforementioned works, and the particular elements in this geometry that determine the problem feasibility and optimality. When viewed from this standpoint, an important contribution of the presented work is a necessary and sufficient condition for

[3] In more practical terms, the removed state coordinates correspond to processing stages that will never get entangled in a deadlock.

the existence of a linear classifier representing the maximally permissive DAP for any given instance from the considered RAS class.[4] Furthermore, the geometric results and insights obtained by the analysis that is pursued in this work, and also the consideration of the optimal classifier design problem as a set covering problem defined in terms of certain state subsets, provide a starting point for the development of novel and more efficient customized algorithms for this problem. A first set of results along these lines can be found in [7, 6]; we briefly overview these results in Section 5.

In light of the above discussion, the rest of the paper is organized as follows: Section 2 provides a formal statement of the classifier design problem considered in this work, as abstracted from the results of [20], and it also summarizes some results from [20] that will be useful in the subsequent developments. Section 3 presents the main results of the paper, that will enable the problem reduction to a set covering problem and the further geometrical insights that were mentioned above. Section 4 provides a small but highlighting example of the main ideas and techniques presented in Section 3, while Section 5 elaborates on the more practical aspects of the derived results and their further implications. Finally, Section 6 concludes the paper by summarizing its results and outlining some directions for future work.

2 The considered classification problem

The reduced state classification problem that was introduced in the previous section can be formally described as follows: Consider two finite vector subsets of $(\mathbb{Z}_0^+)^\xi$, $\mathcal{S}$ and $\mathcal{U}$, and let $I \equiv \{1,\ldots,\xi\}$, i.e., I denotes the set of dimensions of the space where $\mathcal{S}$ and $\mathcal{U}$ live. Set $\mathcal{S}$ corresponds the *maximal safe* states of the introductory discussion, set $\mathcal{U}$ corresponds to the *minimal unsafe* states, and furthermore, it is assumed that any dimensions that are identically equal to zero in the minimal unsafe states have been removed from both of these sets. In addition, the sets $\mathcal{S}$ and $\mathcal{U}$ satisfy the following properties:

Property 1 *There is no dominance either in $\mathcal{S}$ or in $\mathcal{U}$ w.r.t. the relation "$\leq$", where the latter is defined through componentwise application. Furthermore, there is no pair $(\mathbf{s},\mathbf{u}) \in \mathcal{S} \times \mathcal{U}$ such that $\mathbf{u} \leq \mathbf{s}$.*

Property 2 *There is no $i \in I$ such that $\mathbf{s}_i = 0$, $\forall \mathbf{s} \in \mathcal{S}$, or $\mathbf{u}_i = 0$, $\forall \mathbf{u} \in \mathcal{U}$.*

Property 1 expresses the maximality of the elements of $\mathcal{S}$ and the minimality of the elements of $\mathcal{U}$ in the reduced classification problem described in Section 1, and also, the monotonicity property of state safety that was established in [20]. On the other hand, the first part of Property 2 results from the fact that, in a well-defined RAS, every processing stage can be safely executed by a process instance

[4] We remind the reader that the results of [20] establish the binary nature of the underlying RAS state space only as a sufficient condition for the existence of a linear classifier.

of the corresponding process type that runs in the considered RAS in isolation. The second part of Property 2 expresses the aforementioned fact that in the simplified classification problem, dimensions that are identically equal to zero in the set of minimal unsafe states have been projected away.

We also emphasize that, in the following developments, the components of the classified vectors are considered as non-negative integers, but not necessarily binary (which is the case in [20]). While the imposition of a binary constraint for the elements of $\mathcal{S}$ and $\mathcal{U}$ provides a guarantee for the existence of a linear classifier, as established in [20], it is not necessary for the development of the results pursued in this work. In fact, allowing the elements of $\mathcal{S}$ and $\mathcal{U}$ to take non-binary values will provide important insights about the potential non-existence of a linear classifier in this broader setting, and will facilitate a systematic investigation of the conditions that enable linear separation.

Following the developments of [20], we define the notion of a *linear classifier* for the aforementioned sets $\mathcal{S}$ and $\mathcal{U}$ as follows:

Definition 1 *A linear classifier $\mathcal{C}$ for the sets $\mathcal{S}$ and $\mathcal{U}$ introduced in the previous paragraphs is any set of linear inequalities $\{(\mathbf{a}_i, b_i),\ i = 1, \ldots, n\}$ such that (i) $b_i \geq 0,\ \forall i$; (ii) $\forall \mathbf{s} \in \mathcal{S}, \forall i \in \{1, \ldots, n\},\ \mathbf{a}_i^T \cdot \mathbf{s} \leq b_i$; and (iii) $\forall \mathbf{u} \in \mathcal{U}, \exists i \in \{1, \ldots, n\}$ with $\mathbf{a}_i^T \cdot \mathbf{u} > b_i$.*

The non-negativity requirement in item (i) of Definition 1 results from requirement that the constructed classifier must classify correctly the entire safe and unsafe subspaces of the underlying RAS. In particular, the state $\mathbf{s}_0 \equiv \mathbf{0}$, which corresponds to the RAS empty state, must be recognized as a safe state.[5] Let $\mathbb{C}(\mathcal{S}, \mathcal{U})$ denote the entire set of linear classifiers for two given sets $\mathcal{S}$ and $\mathcal{U}$ that meet the requirements of Definition 1. Also, for any classifier $\mathcal{C} \in \mathbb{C}(\mathcal{S}, \mathcal{U})$, let $card(\mathcal{C})$ denote the number of inequalities employed by $\mathcal{C}$. Then, we have the following notion of optimality defined over the elements of $\mathbb{C}(\mathcal{S}, \mathcal{U})$:

Definition 2 *A classifier $\mathcal{C}^* \in \mathbb{C}(\mathcal{S}, \mathcal{U})$ is an* optimal *linear classifier for the sets $\mathcal{S}$ and $\mathcal{U}$ iff $card(\mathcal{C}^*) = \inf\{card(\mathcal{C}) : \mathcal{C} \in \mathbb{C}(\mathcal{S}, \mathcal{U})\}$.*

Notice that, since $card(\mathcal{C}) \in \mathbb{Z}^+$, the above notion of the optimal classifier is well-defined, as long as $\mathbb{C}(\mathcal{S}, \mathcal{U}) \neq \emptyset$ (i.e., the sets $\mathcal{S}$ and $\mathcal{U}$ are indeed linearly separable). The next theorem provides an additional important result for the linear classification problem and its optimizing version that are defined respectively by Definitions 1 and 2; this result can be traced in the developments of [20].

[5] The reader should also notice the asymmetry in the role of the sets $\mathcal{S}$ and $\mathcal{U}$ in the classification logic of Definition 1. This asymmetry results by the further intention to eventually obtain a distributed implementation of the derived classifier, by means of the Petri net modeling framework [16]. This framework facilitates the imposition upon the plant dynamics of constraints expressed as conjunctions of linear inequalities, but not as disjunctions of such inequalities.

Theorem 1 *Consider the finite sets $\mathcal{S}$ and $\mathcal{U}$ of $(\mathbb{Z}_0^+)^\xi$ that possess Property 1, and further assume that $\mathbb{C}(\mathcal{S},\mathcal{U}) \neq \emptyset$. Then, there exists an optimal classifier $\mathcal{C}^*$ for $\mathcal{S}$ and $\mathcal{U}$ with $\mathbf{a}_i \geq \mathbf{0}$, $\forall i$.*

The practical significance of the result of Theorem 1 is that it allows us to limit our search for an optimal linear classifier in the subset of $\mathbb{C}(\mathcal{S},\mathcal{U})$ that involves only classifiers with non-negative coefficients.[6] Let us denote this subset by $\mathbb{C}^+(\mathcal{S},\mathcal{U})$. Then, we can epitomize the classification problem that is considered in this work as follows:

Definition 3 *Given the finite sets $\mathcal{S}$ and $\mathcal{U}$ of $(\mathbb{Z}_0^+)^\xi$ that possess Properties 1 and 2, find an optimal (i.e., min-card) classifier $\mathcal{C}^* \in \mathbb{C}^+(\mathcal{S},\mathcal{U})$, provided that $\mathcal{S}$ and $\mathcal{U}$ are linearly separable.*

We close this section with the introduction of some further concepts and remarks that will be useful in the subsequent developments.

Definition 4 *Given a finite set X of $(\mathbb{Z}_0^+)^\xi$ and an inequality $(\mathbf{a},b)$, we shall say that $(\mathbf{a},b)$ is* valid w.r.t. *X iff $\mathbf{a}^T \cdot \mathbf{x} \leq b$, $\forall \mathbf{x} \in X$.*

According to the problem statement of Definition 3, in the rest of this work we are primarily interested in valid inequalities $(\mathbf{a},b)$ w.r.t. the set $\mathcal{S}$ that also satisfy $\mathbf{a} \geq \mathbf{0} \wedge b \geq 0$. In fact, unless stated otherwise, this non-negativity assumption will be implicitly assumed in the following. The reader should also notice that any such valid inequality $(\mathbf{a},b)$ w.r.t. S will have $b > 0$ since, otherwise, Property 2 together with the non-negativity of $\mathbf{a}$ and of the elements of $\mathcal{S}$ will imply that $\mathbf{a} = \mathbf{0}$, as well, and we shall end up with the trivial identity $\mathbf{0}^T \cdot \mathbf{s} = 0$. But if $b > 0$, we can divide both sides of the inequality $(\mathbf{a},b)$ by b, getting its normalized version $(\mathbf{a}',1)$, where $\forall i \in I,\ \mathbf{a}'[i] = \mathbf{a}[i]/b$. Such a normalization allows the representation of the inequalities in $\mathbb{C}^+(\mathcal{S},\mathcal{U})$ only through the vectors $\mathbf{a}'$ of $(\mathbb{R}_0^+)^\xi$ that collect the coefficients that appear in their left-hand-side, and it is the representation that will be adopted in the following.

Finally, given two linearly separable sets $\mathcal{S}$ and $\mathcal{U}$, a classifier $\mathcal{C} \in \mathbb{C}^+(\mathcal{S},\mathcal{U})$, and a (normalized) inequality $\mathbf{a}$ of $\mathcal{C}$, we shall denote by $\mathcal{U}^{\mathbf{a}}$ the set of points $\mathbf{u} \in \mathcal{U}$ that are separated by $\mathbf{a}$; i.e., $\mathbf{u} \in \mathcal{U}^{\mathbf{a}} \iff \mathbf{a}^T \cdot \mathbf{u} > 1$. The reader should also notice that for any pair of inequalities $\mathbf{a}_i$ and $\mathbf{a}_j$ belonging to an optimal classifier $\mathcal{C}^* \in \mathbb{C}^+(\mathcal{S},\mathcal{U})$, $\mathcal{U}^{\mathbf{a}_i} \setminus \mathcal{U}^{\mathbf{a}_j} \neq \emptyset$, since, otherwise, inequality $\mathbf{a}_i$ could be removed from $\mathcal{C}^*$ without affecting the classifier's capability to separate sets $\mathcal{S}$ and $\mathcal{U}$ (and therefore $\mathcal{C}^*$ cannot be optimal).

[6] As explained in the introductory section, this sign restriction is crucial for being able to reduce the original classifier design problem that characterizes the maximally permissive DAP to the (much) simpler classification problem that addresses explicitly only the classification of maximal safe and minimal unsafe states; this last possibility is instrumental for ensuring the computational tractability of the entire approach of [20].

3 Main results

In the first part of this section, we provide a series of results that will enable the reduction of the classifier design problem of Definition 3 to an equivalent set-covering problem, where the covering sets are defined by valid inequalities w.r.t. set $\mathcal{S}$. This reduction also provides an alternative algorithm for the solution of the classifier design problem of Definition 3. In the second part, we provide additional geometrical insights and computational procedures that (i) can streamline the classifier design algorithm developed in the first part, and also (ii) can help resolve the problem of the existence of a linear classifier for any pair of sets $\mathcal{S}$ and $\mathcal{U}$ of $(\mathbb{Z}_0^+)^\xi$ that also possess Properties 1 and 2.

3.1 Reducing the considered classifier-design problem to a set-covering problem

In this subsection, we assume that for the considered sets $\mathcal{S}$ and $\mathcal{U}$, $\mathbb{C}^+(\mathcal{S},\mathcal{U}) \neq \emptyset$; i.e., there exists a linear classifier $\mathcal{C}$ for the sets $\mathcal{S}$ and $\mathcal{U}$, and furthermore, all the inequalities of $\mathcal{C}$ have non-negative coefficients. The next result establishes that the existence of the classifier $\mathcal{C}$ mentioned above implies also the existence of another classifier $\mathcal{C}' \in \mathbb{C}^+(\mathcal{S},\mathcal{U})$ with some stronger topological properties than $\mathcal{C}$ w.r.t. the spatial distribution of the elements of $\mathcal{S}$.

Proposition 1 *Suppose that the considered sets $\mathcal{S}$ and $\mathcal{U}$ are linearly separable, and let $\mathcal{C} \in \mathbb{C}^+(\mathcal{S},\mathcal{U})$. Then, for every inequality $\mathbf{a}$ in $\mathcal{C}$ there is another inequality $\mathbf{a}'$ such that: (i) $\mathcal{U}^{\mathbf{a}'} \supseteq \mathcal{U}^{\mathbf{a}}$; (ii) $\mathbf{a}'$ is valid w.r.t. $\mathcal{S}$; and (iii) $\exists \mathbf{s} \in \mathcal{S}$ such that $\mathbf{a}' \cdot \mathbf{s} = 1$.*

Proof: Since $\mathbf{a}$ is an inequality of $\mathcal{C}$, it is a valid inequality w.r.t. $\mathcal{S}$; i.e.,

$$\forall \mathbf{s} \in \mathcal{S},\ \mathbf{a}^T \cdot \mathbf{s} \leq 1 \tag{1}$$

Define

$$\varepsilon \equiv \min\{\mathbf{s} \in \mathcal{S} : 1 - \mathbf{a}^T \cdot \mathbf{s}\} \tag{2}$$

Equation 1, combined with the non-negativity of $\mathbf{a}^T \cdot \mathbf{s}$, $\forall \mathbf{s} \in \mathcal{S}$, implies that $\varepsilon \in [0,1]$. Furthermore, Property 2 implies that $\max\{\mathbf{s} \in \mathcal{S} : \mathbf{a}^T \cdot \mathbf{s}\} > 0$, and therefore, $\varepsilon \neq 1$. But then, the (normalized) inequality defined by the vector

$$\mathbf{a}' = \frac{1}{1-\varepsilon}\mathbf{a} \tag{3}$$

is a valid inequality w.r.t. $\mathcal{S}$ that also satisfies the condition of part (iii) of the proposition. Furthermore, we have that

$$\forall \mathbf{u} \in \mathcal{U}^{\mathbf{a}},\ \mathbf{a}^T \cdot \mathbf{u} > 1 \geq 1 - \varepsilon \tag{4}$$

and therefore, part (i) of the proposition is also true. □

Proposition 1 implies that for any linearly separable pair of sets $\mathcal{S}$ and $\mathcal{U}$, we can restrict the search for an optimal classifier to those elements of $\mathbb{C}^+(\mathcal{S},\mathcal{U})$ consisting of linear inequalities that bind at (at least) one $\mathbf{s} \in \mathcal{S}$. Next, we provide a systematic characterization of the linear inequalities that are valid inequalities w.r.t. $\mathcal{S}$ and satisfy the aforementioned additional requirement.

We start by characterizing the elements of $\mathcal{S}$ that can bind at least one valid inequality w.r.t. this set; in the following, we shall denote the subset of $\mathcal{S}$ that collects all these elements by $\mathcal{S}'$.

Proposition 2 *An $\mathbf{s}' \in \mathcal{S}$ belongs to $\mathcal{S}'$ iff the optimal value of the following LP is non-negative.*

$$\max_{(\mathbf{a},h)} h \tag{5}$$

s.t.

$$\mathbf{a}^T \cdot \mathbf{s}' = 1 \tag{6}$$

$$\mathbf{a}^T \cdot \mathbf{s} + h \leq 1, \ \ \forall \mathbf{s} \in \mathcal{S} \setminus \{\mathbf{s}'\} \tag{7}$$

$$\mathbf{a} \geq \mathbf{0} \tag{8}$$

Furthermore, if S includes only binary vectors,[7] *then $\mathcal{S}' = \mathcal{S}$.*

Proof: To prove the first part of Proposition 2, we start by noticing that since (i) $\mathbf{a}^T \cdot \mathbf{s} \geq 0, \forall \mathbf{a}, \mathbf{s}$, and (ii) $\mathcal{S}$ is a finite set, the LP of Equations 5–8 will always have a finite optimal solution. Furthermore, if there is an inequality $\mathbf{a}$ that is valid w.r.t. $\mathcal{S}$ and binds at $\mathbf{s}'$, then, every corresponding inequality defined by Constraint 7 will have a non-negative slack, and therefore, the considered LP will have a non-negative optimal value. On the other hand, a negative optimal value for the considered LP implies that, for every selection of $\mathbf{a}$, there is at least one inequality among those defined by Constraint 7 that is violated. Therefore, there is no valid inequality w.r.t. $\mathcal{S}$ that binds at $\mathbf{s}'$, and $\mathbf{s}' \notin \mathcal{S}'$.

To prove the second part of Proposition 2, consider a vector $\mathbf{s} \in \mathcal{S}$ and let $||\mathbf{s}||$ denote its support, i.e., $||\mathbf{s}|| \equiv \{i \in I_P : \mathbf{s}[i] = 1\}$. Also, let $K \equiv card(||\mathbf{s}||)$. Properties 1 and 2 imply that $K > 0$. Finally, define the vector $\mathbf{a} \in (\mathbb{R}_0^+)^\xi$ as follows:

$$\forall i \in I, \ \mathbf{a}[i] = \begin{cases} \frac{1}{K}, & \text{if } i \in ||\mathbf{s}|| \\ 0, & \text{o.w.} \end{cases} \tag{9}$$

It can be easily checked that if all the vectors of $\mathcal{S}$ are binary, the linear inequality defined by $\mathbf{a}$ is a valid inequality w.r.t. $\mathcal{S}$ and binds at $\mathbf{s}$. Hence, $\mathbf{s} \in \mathcal{S}'$. Since $\mathbf{s}$ was chosen arbitrarily, it follows that $\mathcal{S} \subseteq \mathcal{S}'$. Also, from the definition of $\mathcal{S}'$, $\mathcal{S}' \subseteq \mathcal{S}$. Hence, $\mathcal{S}' = \mathcal{S}$. □

[7] As is the case considered in [20]

Next, consider an $\mathbf{s} \in \mathcal{S}'$ and let $A(\mathbf{s})$ denote the set of valid inequalities w.r.t. $\mathcal{S}$ that pass through $\mathbf{s}$; i.e.,

$$A(\mathbf{s}) \equiv \{\mathbf{a} : \mathbf{a}^T \cdot \mathbf{s} = 1;\ \forall \mathbf{s}' \in \mathcal{S} \setminus \{\mathbf{s}\},\ \mathbf{a}^T \cdot \mathbf{s}' \leq 1;\ \mathbf{a} \geq \mathbf{0}\} \tag{10}$$

Property 2 implies that $A(\mathbf{s})$ is a polytope,[8] and its extreme points are deterimed by the binding of the two types of inequalities that appear in its defnition. Let $A'(\mathbf{s})$ denote the set of the extreme points of $A(\mathbf{s})$, and $\overline{A'}(\mathbf{s})$ denote the subset of $A'(\mathbf{s})$ containing its *maximal* elements. Then, we have the following result.

Proposition 3 *Consider a point $\mathbf{u} \in \mathcal{U}$ and suppose that there exists a valid inequality $\mathbf{a} \in A(\mathbf{s})$ for some $\mathbf{s} \in \mathcal{S}'$ that separates $\mathbf{u}$ from $\mathcal{S}$, i.e., $\mathbf{a}^T \cdot \mathbf{u} > 1$. Then, there exists $\mathbf{a}' \in \overline{A'}(\mathbf{s})$ that separates $\mathbf{u}$ from $\mathcal{S}$.*

Proof: First notice that, by the definition of $\overline{A'}(\mathbf{s})$, any $\mathbf{a}' \in \overline{A'}(\mathbf{s})$ is a valid inequality w.r.t. $\mathcal{S}$. Next we proceed to prove the result of Proposition 3 through contradiction. Hence, suppose that

$$\forall \mathbf{a}' \in \overline{A'}(\mathbf{s}),\ \ \mathbf{a}'^T \cdot \mathbf{u} \leq 1 \tag{11}$$

Equation 11, when combined with the non-negativity of the elements of $A(\mathbf{s})$ and of the vector $\mathbf{u}$, also implies that

$$\forall \mathbf{a}' \in A'(\mathbf{s}),\ \ \mathbf{a}'^T \cdot \mathbf{u} \leq 1 \tag{12}$$

According to Minkowski's representation theorem for polytopes [21], any $\mathbf{a} \in A(\mathbf{s})$ can be written as a convex combination of the elements of $A'(\mathbf{s})$. Hence,

$$\mathbf{a} = \sum_{\mathbf{a}' \in A'(\mathbf{s})} \lambda_{\mathbf{a}'} \mathbf{a}' \tag{13}$$

and furthermore,

$$\sum_{\mathbf{a}' \in A'(\mathbf{s})} \lambda_{\mathbf{a}'} = 1 \tag{14}$$

and

$$\forall \mathbf{a}' \in A'(\mathbf{s}),\ \ \lambda_{\mathbf{a}'} \geq 0 \tag{15}$$

Equations 12, 13 and 14 further imply that

$$\mathbf{a}^T \cdot \mathbf{u}\ =\ \sum_{\mathbf{a}' \in A'(\mathbf{s})} \lambda_{\mathbf{a}'} \mathbf{a}'^T \cdot \mathbf{u}\ \leq\ \sum_{\mathbf{a}' \in A'(\mathbf{s})} \lambda_{\mathbf{a}'} = 1 \tag{16}$$

But Equation 16 contradicts the fact that $\mathbf{a}$ separates $\mathbf{u}$ from S and establishes the result. □

[8] We remind the reader that a polytope is a bounded polyhedron.

Proposition 3 implies that, for any $\mathbf{s} \in \mathcal{S}'$, the set of unsafe states, $\mathcal{U}(\mathbf{s})$, that can be separated by valid inequalities in $A(\mathbf{s})$ coincides with the set of unsafe states that can be separated by valid inequalities in $\overline{A'}(\mathbf{s})$. But $\overline{A'}(\mathbf{s})$ is a finite set. Hence, in principle, $\mathcal{U}(\mathbf{s})$ can be obtained by enumerating the elements of $\overline{A'}(\mathbf{s})$, and identifying all the points $\mathbf{u} \in \mathcal{U}$ that are separated by each of them. More formally,

$$\forall \mathbf{s} \in \mathcal{S}', \ \ \mathcal{U}(\mathbf{s}) \equiv \bigcup_{\mathbf{a} \in \overline{A'}(\mathbf{s})} \mathcal{U}^{\mathbf{a}} \tag{17}$$

Once $\mathcal{U}(\mathbf{s})$ is available for every $\mathbf{s} \in \mathcal{S}'$, we can compute for each $\mathcal{U}(\mathbf{s})$ its maximal subsets that are separable from $\mathcal{S}$ by a single inequality in $A(\mathbf{s})$. This can be effectively done through a search process that starts with $\mathcal{U}(\mathbf{s})$ and traverses downwards the lattice of its subsets, terminating the traversal in any given direction every time that it comes across a subset that is separable from $\mathcal{S}$ by a single inequality.

Furthermore, as we saw in the proof of Proposition 3, Minkowski's theorem implies that every inequality $\mathbf{a} \in A(\mathbf{s})$ that is constructed in the aforementioned search can be expressed as a convex combination of the elements of $A'(\mathbf{s})$. The next proposition implies that we can restrict further the search of the aforementioned inequalities into the subset of $A(\mathbf{s})$ that is defined by the convex combinations of the elements of $\overline{A'}(\mathbf{s})$.

Proposition 4 *Consider an inequality* $\mathbf{a} \in A(\mathbf{s})$ *and the set* $\mathcal{U}^{\mathbf{a}}$ *that contains all the elements of* $\mathcal{U}$ *that are separated from* $\mathcal{S}$ *by it. Then, there exists an inequality* $\mathbf{a}'$ *that separates* $\mathcal{U}^{\mathbf{a}}$ *from* $\mathcal{S}$ *and belongs to the convex hull of* $\overline{A'}(\mathbf{s})$.

Proof: By the assumptions of Proposition 4,

$$\forall \mathbf{u} \in \mathcal{U}^{\mathbf{a}}, \ \ \mathbf{a}^T \cdot u > 1 \tag{18}$$

Also, from Minkowski's theorem, we can set

$$\mathbf{a} = \sum_{\mathbf{a}' \in A'(\mathbf{s})} \lambda_{\mathbf{a}'} \mathbf{a}' \tag{19}$$

with

$$\sum_{\mathbf{a}' \in A'(\mathbf{s})} \lambda_{\mathbf{a}'} = 1 \tag{20}$$

and

$$\forall \mathbf{a}' \in A'(\mathbf{s}), \ \ \lambda_{\mathbf{a}'} \geq 0 \tag{21}$$

Let $\hat{\mathbf{a}}'$ be a non-maximal element of $A'(\mathbf{s})$ appearing in the sum of Equation 19. Then, there exists $\tilde{\mathbf{a}}' \in \overline{A'}(\mathbf{s})$ such that $\tilde{\mathbf{a}}' \geq \hat{\mathbf{a}}'$. Furthermore, since $\tilde{\mathbf{a}}'$, $\hat{\mathbf{a}}'$ and $\mathbf{u}$ are non-negative vectors, the vector $\bar{\mathbf{a}}$ that is obtained from $\mathbf{a}$ through the substitution of $\hat{\mathbf{a}}'$ by $\tilde{\mathbf{a}}'$ in the right-hand-side of Equation 19, satisfies the inequalities of Equation 18 and remains an element of $A(\mathbf{s})$ (since it is still a convex combination of the elements of $A'(\mathbf{s})$). Hence, it constitutes an alternative separator of the sets $\mathcal{S}$ and $\mathcal{U}^{\mathbf{a}}$.

But then, the validity of Proposition 4 results through the application of the substitution described in the previous paragraph to every non-maximal element $\mathbf{a}'$ that appears in the representation of $\mathbf{a}$ provided by Equation 19. □

For $\mathbf{s} \in \mathcal{S}'$, let $M(\mathbf{s})$ collect the maximal subsets of $\mathcal{U}(\mathbf{s})$ identified through the aforementioned search process, i.e., each element of $M(\mathbf{s})$ is a maximal subset of $\mathcal{U}(\mathbf{s})$ separated from $\mathcal{S}$ through a single inequality in $A(\mathbf{s})$.[9] Also, let $M \equiv \bigcup_{\mathbf{s}\in\mathcal{S}'} M(\mathbf{s})$. Then, the following theorem is a straightforward implication of all the above developments.

Theorem 2 *Given two linearly separable sets $\mathcal{S}$ and $\mathcal{U}$ that correspond to the input state sets of the simplified classification problem of [20], the problem of constructing an optimal linear classifier for them, $\mathcal{C}^* \in \mathbb{C}^+(\mathcal{S}, \mathcal{U})$, reduces to the problem of constructing a minimal cover Ω for $\mathcal{U}$ from the elements of the set M that was defined in the previous paragraph.*

We notice that the reduction established in Theorem 2 suggests also an alternative algorithm for the construction of a min-card linear classifier for any pair of inearly separable sets $\mathcal{S}$ and $\mathcal{U}$. The main steps of this algorithm are outlined in Algorithm 1 and they are defined by (i) the computation that is necessary for the construction of the set M from the input data sets $\mathcal{S}$ and $\mathcal{U}$, (ii) the solution of the set covering problem defined by $\mathcal{U}$ and M, and (iii) the translation of the results of Step (ii) to a set of linear inequalities to be employed by the sought classifier.

The next section provides some additional results that will enable (i) the resolution of the linear separability of the considered pair of sets $\mathcal{S}$ and $\mathcal{U}$, and (ii) the further streamlining of Algorithm 1.

3.2 Determining the existence of a linear classifier

We begin this section by defining the set

$$\overline{A'}(\mathcal{S}) \equiv \bigcup_{\mathbf{s}\in\mathcal{S}'} \overline{A'}(\mathbf{s}) \tag{22}$$

The significance of this set is revealed by the following proposition.

Proposition 5 *A vector $\mathbf{u} \in \mathcal{U}$ is linearly separable from $\mathcal{S}$ iff there exists at least one inequality in the set $\overline{A'}(\mathcal{S})$, that is defined by Equation 22, such that $\mathbf{a}^T \cdot \mathbf{u} > 1$.*

[9] Or, as established by Proposition 4, through a single inequality in the subspace of $A(\mathbf{s})$ that constitutes the convex hull of $\overline{A'}(\mathbf{s})$.

Algorithm 1: Solving the classifier-design problem of Definition 3 through its reduction to a set-covering problem

```
Input: (i) Subsets $\mathcal{S}$ and $\mathcal{U}$ of $(\mathbb{Z}_0^+)^\xi$ that correspond to the input state sets of the simplified classification problem presented in [20].
Output: A min-card classifier $\mathcal{C} \in \mathbb{C}^+(\mathcal{S}, \mathcal{U})$.
Compute the set $\mathcal{S}' \subseteq \mathcal{S}$ by applying the LP of Proposition 2 to every $\mathbf{s} \in \mathcal{S}$.    /* If $\mathcal{S}$ includes binary vectors only, then the second part of Prop. 2 implies that we can immediately set $\mathcal{S}' = \mathcal{S}$. */
for all $\mathbf{s} \in \mathcal{S}'$ do
    Compute the set $\overline{A'}(\mathbf{s})$.
    Compute the set $\mathcal{U}(\mathbf{s})$ containing all the $\mathbf{u} \in \mathcal{U}$ that are separated from $\mathcal{S}$ by some inequality in $\overline{A'}(\mathbf{s})$.
    Compute the set $M(\mathbf{s})$ containing the maximal subsets of $\mathcal{U}(\mathbf{s})$ that are separable from $\mathcal{S}$ through a single inequality in the convex hull of $\overline{A'}(\mathbf{s})$.
end for
$M := \bigcup_{\mathbf{s} \in \mathcal{S}'} M(\mathbf{s})$.
Compute a min-card cover $\Omega \subseteq M$ for $\mathcal{U}$ from the elements of $M$.
$\mathcal{C} := \{\mathbf{a} :\ \mathbf{a}$ is the separating inequality for one of the elements of $\Omega$ that was constructed in Step 5$\}$.
Return $\mathcal{C}$.
```

Proof: First, assume that there exists an $\mathbf{a} \in \overline{A'}(\mathcal{S})$ such that $\mathbf{a}^T \cdot \mathbf{u} > 1$. Since, by construction, all the elements of $\overline{A'}(\mathcal{S})$ are valid inequalities w.r.t. $\mathcal{S}$, $\mathbf{u}$ is linearly separable from $\mathcal{S}$.

Next, assume that $\mathbf{u}$ is linearly separable from $\mathcal{S}$. Then, from the remark provided at the end of Section 2, it can be inferred that an optimal classifier will involve one inequality only. Theorem 1 of Section 2 also implies that this inequality can have only non-negative coefficients. Subsequently, Proposition 1 implies that the considered inequality can be chosen so that it binds at some $\mathbf{s} \in \mathcal{S}'$, and finally, Proposition 3 further ensures the existence of an inequality with the aforementioned properties which is also a maximal extreme point for the corresponding $A(\mathbf{s})$, and therefore, it belongs in $\overline{A'}(\mathcal{S})$. □

Next we introduce the polyhedron $\Upsilon(\mathcal{S})$ that is defined through $\overline{A'}(\mathcal{S})$ as follows:

$$\Upsilon(\mathcal{S}) \equiv \{\mathbf{x} \in (\mathbb{R}_0^+)^\xi : \mathbf{a}^T \cdot \mathbf{x} \leq 1,\ \forall \mathbf{a} \in \overline{A'}(\mathcal{S})\} \tag{23}$$

The following result is an immediate corollary of the above definition of $\Upsilon(\mathcal{S})$ and of the result of Proposition 5.

Corollary 1 *The sets $\mathcal{S}$ and $\mathcal{U}$ that correspond to the input state sets of the simplified classification problem of [20] are linearly separable iff* $\Upsilon(\mathcal{S}) \cap \mathcal{U} = \emptyset$.

Proposition 5 and Corollary 1 resolve the issue of the linear separability of any pair of sets $\mathcal{S}$ and $\mathcal{U}$ that correspond to the input state sets of the simplified classification problem of [20], and through the generalizing capabilities of the constructed classifier that were discussed in the introductory section, they also resolve the issue of the linear separability of the entire sets of safe and unsafe RAS states, which are the primary sets of interest in the implementation of the maximally permissive DAP. The next series of results establish some further properties of $\Upsilon(\mathcal{S})$ that will lead to a tighter characterization of this polyhedron and will also reveal a duality between $\Upsilon(\mathcal{S})$ and the set of valid inequalities w.r.t. $\mathcal{S}$ with non-negative coefficients. Finally, the presented results can help streamline some of the computations involved in Algorithm 1.

Proposition 6 *The polyhedron $\Upsilon(\mathcal{S})$ that is defined by Equation 23 is a* full-dimensional polytope *lying in* $(\mathbb{R}_0^+)^\xi$.

Proof: First we show that $\Upsilon(\mathcal{S})$ is a polytope. By its definition, $\Upsilon(\mathcal{S})$ lies in $(\mathbb{R}_0^+)^\xi$, and therefore, each of its elements is bounded from below by $\mathbf{0}$. Also, it should be clear that the origin itself belongs in $\Upsilon(\mathcal{S})$. To show that $\Upsilon(\mathcal{S})$ is also bounded from above, we use contradiction.

Hence, suppose that $\Upsilon(\mathcal{S})$ possesses a ray $\mathbf{r} \geq \mathbf{0}$ such that $\forall \lambda \in \mathbb{R}_0^+$, $\lambda \mathbf{r} \in \Upsilon(\mathcal{S})$; i.e., there exists some direction in $(\mathbb{R}_0^+)^\xi$ in which $\Upsilon(\mathcal{S})$ expands to infinity. Also, let $||\mathbf{r}||$ denote the support of $\mathbf{r}$, i.e., the dimensions $i \in I$ for which $\mathbf{r}[i] > 0$.

Let $\theta \equiv \max_{\mathbf{s} \in \mathcal{S}} \max_{i \in I} \mathbf{s}[i]$ and pick a λ' such that $\exists i \in ||\mathbf{r}||, \;\; \lambda' \mathbf{r}[i] > \theta$. The point $\lambda' \mathbf{r}$ is linearly separable from $\mathcal{S}$ through the inequality $\mathbf{x}[i] \leq \theta$. But then, according to Corollary 1, $\lambda' \mathbf{r}$ cannot belong in $\Upsilon(\mathcal{S})$. The resulting contradiction establishes the boundedness of $\Upsilon(\mathcal{S})$ (and the fact that it is a polytope).

To show that $\Upsilon(\mathcal{S})$ is full-dimensional, it suffices to show that it includes all the unit vectors $\mathbf{e}_i$, $i \in I$. Since, as pointed out above, $\Upsilon(\mathcal{S})$ also includes the origin, the set $\{\mathbf{e}_i, \; i \in I\} \cup \{\mathbf{0}\}$ provides $\xi + 1$ affinely independent points of $\Upsilon(\mathcal{S})$, and the result will be established. Hence, pick an $i \in I$ and notice that the corresponding $\mathbf{e}_i$ is the smallest non-zero integral vector, w.r.t. the relationship "$\leq$" that was introduced in Property 1, that has a non-zero value for its i-th component. By Property 2, there exists some $\mathbf{s} \in \mathcal{S}$ such that $\mathbf{e}_i \leq \mathbf{s}$. But then, for any inequality $\mathbf{a}$ with non-negative coefficients that is a valid inequality w.r.t. $\mathcal{S}$, we shall also have $\mathbf{a}^T \cdot \mathbf{e}_i \leq 1$. Hence, $\mathbf{e}_i$ is not linearly separable from $\mathcal{S}$, and therefore, by Corollary 1, $\mathbf{e}_i \in \Upsilon(\mathcal{S})$. $\square$

The following result is taken from [21] (cf. Propositions 5.8 and 5.9 in pg. 103 of that text), and it will help us characterize the dual relationship between $\Upsilon(\mathcal{S})$ and the set of valid inequalities w.r.t. $\mathcal{S}$

with non-negative coefficients, that was mentioned above.

Proposition 7 *Let $P = \{\mathbf{x} \in (\mathbb{R}_0^+)^\xi : \; B\mathbf{x} \leq \mathbf{1}\}$, where B is a non-negative matrix with no zero columns. Also, denote by Ψ the $\kappa \times \xi$ matrix whose rows are the extreme points of P. Finally, define the polytope $P^C \equiv \{\pi \in (\mathbb{R}_0^+)^\xi : \; \pi^T \cdot \mathbf{x} \leq 1, \; \forall \mathbf{x} \in P\}$; polytope P^C is typically known as the* antiblocker *of P. Then, the following statements hold true:*

i. $P^C = \{\pi \in (\mathbb{R}_0^+)^\xi : \; \Psi\pi \leq \mathbf{1}\}$.

ii. $(P^C)^C = P$.

iii. The facet-defining inequalities of P^C are the inequalities $\bar{\mathbf{x}}_i^T \cdot \pi \leq 1, \; i = 1, \ldots, \bar{\kappa}$, where $\{\bar{\mathbf{x}}_i\}_{i=1}^{\bar{\kappa}}$ are the extreme points of P that are maximal *in P.*

Next, consider the set $\mathcal{V}(\mathcal{S}) \equiv \{\mathbf{a} \in (\mathbb{R}_0^+)^\xi : \; \mathbf{s}^T \cdot \mathbf{a} \leq 1, \; \forall \mathbf{s} \in \mathcal{S}\}$. $\mathcal{V}(\mathcal{S})$ essentially collects all the valid inequalities w.r.t. $\mathcal{S}$ with non-negative coefficients. Also, Property 2 implies that there is an inequality with a strictly positive coefficient for every dimension $i \in I$, and therefore, $\mathcal{V}(\mathcal{S})$ is a polytope.

Proposition 8 *Polytope $\Upsilon(\mathcal{S})$ is the* antiblocker *of polytope $\mathcal{V}(\mathcal{S})$. Furthermore, the facets of $\Upsilon(\mathcal{S})$ correspond to the* maximal *extreme points of $\mathcal{V}(\mathcal{S})$.*

Proof: It should be clear to the reader that, by its construction, the set $\overline{A'}(\mathcal{S})$ of Equation 22 collects all the maximal extreme points of $\mathcal{V}(\mathcal{S})$ that bind at some element(s) of $\mathcal{S}$. But then, the first part of Proposition 8 follows from the definition of $\Upsilon(\mathcal{S})$ in Equation 23, and parts (i) and (iii) of Proposition 7. The second part of Proposition 8 results from its first part and part (iii) of Proposition 7. □

In plain terms, Proposition 8 implies that $\Upsilon(\mathcal{S})$ collects all the points $\mathbf{x} \in (\mathbb{R}_0^+)^\xi$ that satisfy all the possible valid inequalities w.r.t. $\mathcal{S}$ with non-negative coefficients. It is exactly for this reason that Corollary 1 is true. The second part of Proposition 8 implies that one can compute the facets of $\Upsilon(\mathcal{S})$ by computing the maximal extreme points of $\mathcal{V}(\mathcal{S})$. This computation can also take advantage of the set $\mathcal{S}'$ of Proposition 2, since the constraints of $\mathcal{V}(\mathcal{S})$ corresponding to $\mathbf{s} \in \mathcal{S} \setminus \mathcal{S}'$ cannot be binding for any $\mathbf{a} \in (\mathbb{R}_0^+)^\xi$, and therefore, they cannot be part of a set of constraints defining an extreme point of $\mathcal{V}(\mathcal{S})$. Once the facets of $\Upsilon(\mathcal{S})$ have been obtained, then, it is also possible to check the separability of $\mathcal{U}$ and $\mathcal{S}$ according to the result of Corollary 1. The computation involved in this last test can also be leveraged for the construction of the sets $\overline{A'}(\mathbf{s})$ and $\mathcal{U}(\mathbf{s})$, $\mathbf{s} \in \mathcal{S}'$, that appear in Algorithm 1. The details for the completion and re-organization of the computation of Algorithm 1 along the lines discussed above are depicted in Algorithm 2. In the next section we demonstrate the application of Algorithm 2 through a small but highlighting example.

Algorithm 2: Checking the feasibility of the classifier-design problem of Definition 3 through the construction of polytope $\Upsilon(\mathcal{S})$, and solving the separable cases through their reduction to a set-covering problem

```
Input: (i) Subsets $\mathcal{S}$ and $\mathcal{U}$ of $(\mathbb{Z}_0^+)^\xi$ that correspond to the input state sets of the simplified classification problem presented in [20].
Output: A min-card linear classifier $\mathcal{C} \in \mathbb{C}^+(\mathcal{S},\mathcal{U})$ that separates $\mathcal{S}$ and $\mathcal{U}$, or a signal indicating the non-separability of these two sets.
 1: Compute the set $\mathcal{S}' \subseteq \mathcal{S}$ by applying the LP of Proposition 2 to every $\mathbf{s} \in \mathcal{S}$.   /* If $\mathcal{S}$ includes binary vectors only, then the second part of Prop. 2 implies that we can immediately set $\mathcal{S}' = \mathcal{S}$. */
 2: Compute the set $\mathcal{F}$ of the (non-trivial) facets of polytope $\Upsilon(\mathcal{S})$ through the enumeration of the maximal extreme points of polytope $\mathcal{V}(\mathcal{S})$, while avoiding the consideration of inequalities that correspond to the elements of $\mathcal{S} \setminus \mathcal{S}'$.
 3: for all $\mathbf{u} \in \mathcal{U}$ do
 4:   $SEPARABLE := FALSE$.
 5:   for all $\mathbf{a} \in \mathcal{F}$ do
 6:     if $\mathbf{a}^T \cdot \mathbf{u} > 1$ then
 7:       $SEPARABLE := TRUE$.
 8:       Enter $\mathbf{u}$ in $\mathcal{U}^{\mathbf{a}}$.
 9:     end if
10:   end for
11:   if $SEPARABLE = FALSE$ then
12:     Exit indicating non-separability.
13:   end if
14: end for
15: for all $\mathbf{s} \in \mathcal{S}'$ do
16:   Extract the set $\overline{A'}(\mathbf{s}) \subseteq A'(\mathbf{s})$ that collects all the facets in $\mathcal{F}$ that are bound by $\mathbf{s}$.
17:   $\mathcal{U}(\mathbf{s}) := \bigcup_{\mathbf{a} \in \overline{A'}(\mathbf{s})} \mathcal{U}^{\mathbf{a}}$.
18:   Compute the set $M(\mathbf{s})$ containing the maximal subsets of $\mathcal{U}(\mathbf{s})$ that are separable from $\mathcal{S}$ through a single inequality in the convex hull of $\overline{A'}(\mathbf{s})$.
19: end for
20: $M := \bigcup_{\mathbf{s} \in \mathcal{S}'} M(\mathbf{s})$.
21: Compute a min-card cover $\Omega \subseteq M$ for $\mathcal{U}$ from the elements of $M$.
22: $C := \{\mathbf{a} :\ \mathbf{a}$ is the separating inequality for one of the elements of $\Omega$ that was constructed in Step 18$\}$.
23: Return $\mathcal{C}$.
```

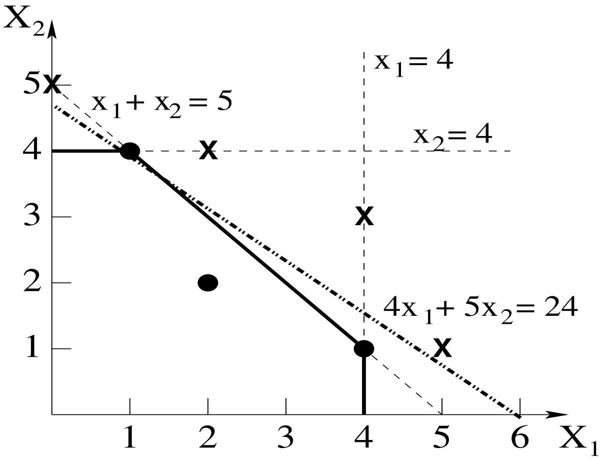

Figure 1: The geometry of the classification problem discussed in Section 4.

4 Example

In this section we present an example that demonstrates the ideas and the techniques that were discussed in Section 3, and provides some further details for the involved computations. The example is rather small and artificial,[10] and for expository purposes it is kept in the 2-dim space, but it highlights all the main concepts and themes of the previous discussion. More specifically, in this example we are called to test the linear separability of the following two sets, $\mathcal{S} = \{(1,4),(2,2),(4,1)\}$ and $\mathcal{U} = \{0,5),(2,4),(4,3),(5,1)\}$, and construct a min-card linear classifier, in the case that they are found to be linearly separable. It can be easily checked that $\mathcal{S}$ and $\mathcal{U}$ satisfy the conditions stated in Properties 1 and 2, and therefore, we can proceed with the application of the results of Section 3.

Hence, using the LP of Proposition 2,[11] we can see that for the considered example, $\mathcal{S}' = \{(1,4),(4,1)\}$. The set of the valid inequalities w.r.t. $\mathcal{S}$ can be expressed as follows:

$$\mathcal{V}(\mathcal{S}) = \{(a_1,a_2) \in \mathbb{R}^2 : \ a_1 + 4a_2 \leq 1; \ 2a_1 + 2a_2 \leq 1; \ 4a_1 + a_2 \leq 1; \ a_1 \geq 0; \ a_2 \geq 0\} \tag{24}$$

The computation of the maximal extreme points of $\mathcal{V}(\mathcal{S})$, while taking into consideration the content of set $\mathcal{S}'$, is tabulated in Table 1. In the depicted computation, the constraints that define $\mathcal{V}(\mathcal{S})$ are numbered in the order that they appear in the right-hand-side of Equation 24. From Table 1, we can see that the set $\mathcal{F}$, that containts the facets of the polytope $\Upsilon(\mathcal{S})$ for the considered example, is

$$\mathcal{F} = \{(1/5,1/5),(0,1/4),(1/4,0)\} \tag{25}$$

[10] By artificial we mean that the sets $\mathcal{S}$ and $\mathcal{U}$ considered in this example have not been produced through a particular RAS instance.

[11] Of course, in this simple case, this issue can be resolved quite easily by inspection!

Binding Constraints	(a_1, a_2)	Feasible	Maximal
(1), (3)	(1/5, 1/5)	YES	YES
(1), (4)	(0, 1/4)	YES	YES
(1), (5)	(1, 0)	NO	N/A
(3), (4)	(0, 1)	NO	N/A
(3), (5)	(1/4, 0)	YES	YES
(4), (5)	(0, 0)	YES	NO

Table 1: Computing the maximal extreme points for the set $\mathcal{V}(\mathcal{S})$ of Equation 24.

$\mathcal{U}^{(1/5,1/5)} = \{(2,4),(4,3),(5,1)\}$
$\mathcal{U}^{(0,1/4)} = \{(0,5)\}$
$\mathcal{U}^{(1/4,0)} = \{(5,1)\}$
$\overline{A'}(1,4) = \{(1/5,1/5),(0,1/4)\}$
$\overline{A'}(4,1) = \{(1/5,1/5),(1/4,0)\}$
$\mathcal{U}(1,4) = \{(2,4),(4,3),(5,1),(0,5)\}$
$\mathcal{U}(4,1) = \{(2,4),(4,3),(5,1)\}$

Table 2: A listing of the sets $\mathcal{U}^{\mathbf{a}}$, $\overline{A'}(\mathbf{s})$, and $\mathcal{U}(\mathbf{s})$, that are employed by Algorithm 2, as they materialize in the example of Section 4.

The corresponding inequalities are:

$$\frac{1}{5}x_1 + \frac{1}{5}x_2 \leq 1 \Longleftrightarrow x_1 + x_2 \leq 5 \tag{26}$$

$$\frac{1}{4}x_2 \leq 1 \Longleftrightarrow x_2 \leq 4 \tag{27}$$

$$\frac{1}{4}x_1 \leq 1 \Longleftrightarrow x_1 \leq 4 \tag{28}$$

Figure 1 depicts in solid lines the polytope $\Upsilon(\mathcal{S})$ that is defined by these inequalities, and also the distribution of the sets $\mathcal{S}$ and $\mathcal{U}$ in the corresponding plane. From this figure, it can be easily checked that the sets $\mathcal{U}^{\mathbf{a}}$ for $\mathbf{a} \in \mathcal{F}$, and also the sets $\overline{A'}(\mathbf{s})$ for $\mathbf{s} \in \mathcal{S}'$, are structured as depicted in Table 2. Also, the availability of these two types of sets enables the computation of the sets $\mathcal{U}(\mathbf{s})$ for $\mathbf{s} \in \mathcal{S}'$, according to Step 17 of Algorithm 2; these sets are also listed in Table 2.

At this point, Algorithm 2 would need to construct the sets $M(\mathbf{s})$, $\mathbf{s} \in \mathcal{S}'$, containing the maximal subsets of the corresponding $\mathcal{U}(\mathbf{s})$ that are linearly separable from $\mathcal{S}$ by a single straight line passing through $\mathbf{s}$. Since, in the considered case, $\mathcal{U}(1,4) = \mathcal{U}$, it is pertinent to start by examining the possibility of separating the entire set $\mathcal{U}$ by a single line that passes through point $(1,4)$. This can be done by testing whether the following LP admits a strictly positive optimal solution.

$$\max \varepsilon \tag{29}$$

s.t.

$$2a_1 + 4a_2 - \varepsilon \geq 1 \tag{30}$$

$$4a_1 + 3a_2 - \varepsilon \geq 1 \tag{31}$$

$$5a_1 + a_2 - \varepsilon \geq 1 \tag{32}$$

$$5a_2 - \varepsilon \geq 1 \tag{33}$$

$$a_1 - \frac{1}{5}\lambda_1 = 0 \tag{34}$$

$$a_2 - \frac{1}{5}\lambda_1 - \frac{1}{4}\lambda_2 = 0 \tag{35}$$

$$\lambda_1 + \lambda_2 = 1 \tag{36}$$

$$\lambda_1, \lambda_2 \geq 0 \tag{37}$$

The objective of this LP together with the first four constraints of it express the request for a separatrice of $\mathcal{U}$ from $\mathcal{S}$, while the remaining constraints express the fact that this separatrice is sought in the convex hull of the set $\overline{A'}(1,4)$.[12] For the considered case, it is easy to check, through Figure 1, that a linear separatrice for sets $\mathcal{S}$ and $\mathcal{U}$ can be obtained by pivoting the line $x_1 + x_2 = 5$ at point $(1,4)$ in the counter-clockwise sense. Indeed, the solution of the LP of Equations 29–37 resulted in an optimal value of $\varepsilon^* = 1/24$.[13] The particular line that corresponds to this optimal value is described by the vector $(a_1^*, a_2^*) = (1/6, 5/24)$, and it is depicted in Figure 1 by a bold dashed line.

[12] Also, it should be noticed, for completeness, that Constraints 34 and 35 enable the substitution of variables a_1 and a_2 by the corresponding linear combinations of λ_1 and λ_2, across the remaining constraints. We have opted to retain variables a_1 and a_2 in the presented formulation in order to provide a clearer demonstration of the logic that underlies this formulation.

[13] In natural terms, ε^* is the distance between the constructed separatrice and the point $\mathbf{u} \in \mathcal{U}$ that is closest to this line, and it can be perceived as a "measure" of the separability of $\mathcal{S}$ and $\mathcal{U}$ by a straight line that passes through the chosen point $\mathbf{s} \in \mathcal{S}'$.

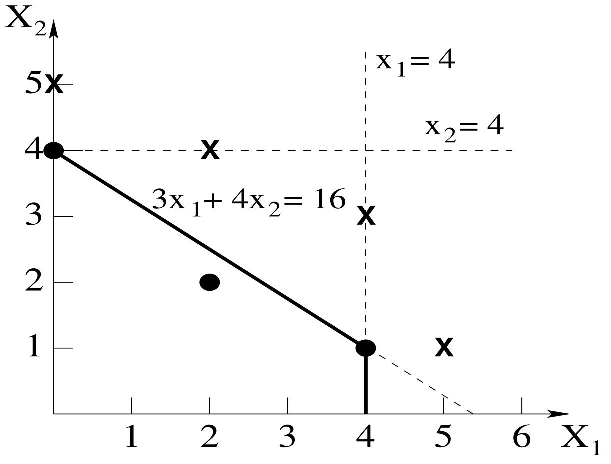

Figure 2: A variation of the Example of Figure 1 that demonstrates the need for focusing on the maximal extreme points of $\mathcal{V}(\mathcal{S})$ during the characterization of the polytope $\Upsilon(\mathcal{S})$.

Binding Constraints	(a_1, a_2)	Feasible	Maximal
(1), (3)	(3/16, 1/4)	YES	YES
(1), (4)	(0, 1/4)	YES	NO
(1), (5)	–	NO	N/A

Table 3: Computing the maximal extreme points for the set $A(0,4)$ of Equation 38.

In the last part of this section, we discuss a variation of the above example that will explain, from a more conceptual standpoint, the need for focusing only on (i) the maximal extreme points of $\mathcal{V}(\mathcal{S})$ during the computation of the facets of polytope $\Upsilon(\mathcal{S})$, and, consequently, on (ii) the sets $\overline{A'}(\mathbf{s})$, $\mathbf{s} \in \mathcal{S}'$, in the computations of Algorithms 1 and 2. More specifically, the set $\mathcal{S}$ for this new problem instance is obtained from the corresponding set in the original example by substituting the point $(1,4)$ with $(0,4)$. The new situation is depicted in Figure 2. It is easily checked (e.g., through Figure 2) that, in this new case, point $(2,2)$ remains outside the set $\mathcal{S}'$, i.e., $\mathcal{S}' = \{(0,4),(4,1)\}$.

Next, we focus upon the set $A(0,4)$, that contains the valid inequalities wr.t. $\mathcal{S}$ that have non-negative coefficients only and bind at $(0,4)$. According to Equation 10, $A(0,4)$ is described by the following constraints:

$$A(0,4) = \{(a_1,a_2) \in \mathbb{R}^2 : \; 4a_2 = 1; \; 2a_1 + 2a_2 \leq 1; \; 4a_1 + a_2 \leq 1; \; a_1 \geq 0; \; a_2 \geq 0\} \qquad (38)$$

The computation of the extreme points of $A(0,4)$ is depicted in Table 3. In the depicted computation, the constraints defining $A(0,4)$ have been numbered according to the order that they appear in the right-hand-side of Equation 38, and furthermore, we have also accounted for the fact that point $(2,2)$ cannot bind any valid inequality that is defined by the elements of $A(0,4)$ (and, therefore, it cannot be part of a set of inequalities that define an extreme point of $A(0,4)$). According to Table 3, the set of extreme points of $A(0,4)$ is $A'(0,4) = \{(3/16, 1/4), (0, 1/4)\}$, which further implies that the only maximal extreme point of $A(0,4)$ is $(3/16, 1/4)$; i.e., $\overline{A'}(0,4) = \{(3/16, 1/4)\}$.

Figure 2 also depicts the straight lines that correspond to the elements of $A'(0,4)$. It is easy to see from this drawing that the maximal extreme point $(3/16, 1/4)$, that corresponds to a valid inequality $\mathbf{a}_1$ binding at the points $(0,4)$ and $(4,1)$ of $\mathcal{S}'$, defines indeed a facet of $\Upsilon(\mathcal{S})$. On the other hand, the non-maximal extreme point $(0, 1/4)$ corresponds to a valid inequality $\mathbf{a}_2$ that is defined by its binding at $(0,4)$ and the non-negativity restriction for coefficient a_1; the binding of this last constraint implies that the line defined by $\mathbf{a}_2$ is parallel to axis X_1. Since, however, point $(0,4)$ is located on the X_2 axis, the facet of $\Upsilon(\mathcal{S})$ that would have been defined by $\mathbf{a}_2$ reduces to the zero-dimensional set containing only the point $(0,4)$. More generally, in problem instances formulated in $(\mathbb{R}_0^+)^\xi$, a non-maximal extreme point of a set $A(\mathbf{s})$, $\mathbf{s} \in \mathcal{S}'$, will correspond to a valid inequality that is defined by (i) its binding at some points $\mathbf{s} \in \mathcal{S}'$ that are lying in some hyperplane $x_i = 0,\ i \in I$, and (ii) a number of non-negativity constraints $a_i \geq 0,\ i \in I$, in a way that the facet of $\Upsilon(\mathcal{S})$ that would be induced by this inequality degenerates to a polytope of dimensionality strictly smaller than $\xi - 1$.

5 Discussion

In this section, we consider the more practical implications and value of the results that were developed in the earlier parts of this manuscript. In particular, first we take a closer look at the computational complexity of Algorithm 2 and the potential limitations that this complexity might imply for the practical applicability of this algorithm. On the other hand, as stated in the earlier parts of this manuscript, the main value of the presented results stems from the analytical insights that they provide regarding the geometry of the linear classification problem that is addressed in [20]. These insights have enabled the development of the condition of Section 3.2 regarding the existence of a linear classifier for the considered classification problem, and they also establish a theoretical basis for the development of more streamlined algorithms for the computation of a structurally minimal classifier, for, both, the linear and the non-linear problem versions that were discussed in Section 1; this potential is considered in the second part of this section.

Regarding the computational complexity of Algorithm 2, we remark the following: It is well known that the minimal set covering problem is NP-hard [11]. Hence, the solution of the transformed problem remains a challenging task. Even more importantly, the reduction process itself, that is materialized through Steps 1–20 in Algorithm 2, is a computationally intensive task. In particular, Step 2, that constructs the non-trivial facets of polytope $\Upsilon(\mathcal{S})$, and Step 18, that identifies the maximal subsets of $\mathcal{U}$ that are separable from $\mathcal{S}$ by a single inequality passing through one of the maximal elements of the latter, are quite demanding tasks, as they involve enumerations over spaces that grow exponentially w.r.t. their defining elements. These remarks imply that, from a purely computational standpoint, Algorithm 2 might not be a very competitive proposition; yet, its primary value lies in the constructive insights that it provides regarding the underlying problem structure.

Along these more qualitative lines, as mentioned in the introductory section, the work of [20] has already exploited the analogies that are established by Algorithm 2 in order to provide a very efficient heuristic for the problem of constructing a minimal linear classifier for the maximally permissive DAP, assuming that such a classifier is available. This heuristic seeks to develop a linear separator by developing the linear inequalities in an iterative scheme, one at a time, and with the objective of each iteration being to construct an inequality that separates the maximal possible subset of the currently non-separated unsafe states. Using a line of argumentation that parallels the development of a similar heuristic for the set covering problem, in [20] it is shown that the suboptimality of the resulting classifier can be bounded by a factor of $\ln|\mathcal{U}|$.

More recently, the identified relationship between the classifier design problem of [20] and the classical set covering problem has been employed towards the development of a customized Branch & Bound (B&B) algorithm [21] for the former. This algorithm appeared originally in [8], and it essentially orchestrates an efficient search among the possible partitions of the set $\mathcal{U}$ into a number of subsets that are linearly separable from set $\mathcal{S}$ by a single linear inequality, for a minimal such partition. Furthermore, the works of [7, 6] have sought to complement the developments of [8] with the existence results and the analytical insights that were developed in this work, in order to extend the B&B scheme of [8] so that it applies to the broader non-linear classification problem that is addressed in [17, 19]. The basic logic that facilitates this extension can be stated as follows: As long as the classification problem addressed admits a linear classifier, then, the construction of a minimal classifier can be resolved through the B&B scheme of [8]. In the opposite case, the results of Section 3 in this paper suggest that one can represent the safe subspace that is defined by $\mathcal{S}$ as a *"union"* of polytopes $\Upsilon_i(\mathcal{S}),\ i = 1,\ldots,q$, where each polytope $\Upsilon_i(\mathcal{S})$ encompasses a subset $\mathcal{S}_i$ of $\mathcal{S}$ and possesses similar topological properties w.r.t. set $\mathcal{U}$ with those that are possessed by the set $\Upsilon(\mathcal{S})$ in the linear case. Hence, in this more complex case, the search for a minimal classifier boils down to a double search for (i) a pertinent partition of the set $\mathcal{S}$ to a set of subsets S_i that are linearly separable from $\mathcal{U}$, and for (ii) minimal linear classifiers for each of these subsets. The B&B method of [7, 6] can be perceived as an efficient process that carries out these two searches in parallel and in an incremental manner, while capitalizing upon the geometrical and combinatorial perspectives and insights that were revealed by the analysis that is pursued in this work.

6 Conclusion

This paper has further explored and formalized the connection, that was originally identified in [20], between the linear classification problem that underlies the implementation of the maximally permissive DAP for certain RAS classes that arise in many contemporary applications and the classical set-covering problem that has been studied in OR and CS. Furthermore, it has provided additional insights about the problem geometry that determine its feasibility and its combinatorial attributes. Finally, through the discussion of the last section, it has also highlighted how these new insights provide a starting point for the development of further, more efficient customized algorithms for the considered problem, as well as for its variant that might necessitate a non-linear structure for the sought classifier. The complete realization of this possibility is part of our ongoing investigations.

References

[1] T. Araki, Y. Sugiyama, and T. Kasami. Complexity of the deadlock avoidance problem. In *2nd IBM Symp. on Mathematical Foundations of Computer Science*, pages 229–257. IBM, 1977.

[2] Z. A. Banaszak and B. H. Krogh. Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows. *IEEE Trans. on Robotics and Automation*, 6:724–734, 1990.

[3] R.E. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, 100(8):677–691, 1986.

[4] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems (2nd ed.)*. Springer, NY,NY, 2008.

[5] Y. F. Chen and Z. W. Li. Design of a maximally permissive liveness-enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems. *Automatica*, 47:1028–1034, 2011.

[6] R. Cordone, A. Nazeem, L. Piroddi, and S. Reveliotis. Designing optimal deadlock avoidance policies for sequential resource allocation systems through classification theory: existence results and customized algorithms. Technical Report (submitted for publication), Univ. of Milan, 2012.

[7] R. Cordone, A. Nazeem, L. Piroddi, and S. Reveliotis. Maximally permissive deadlock avoidance for sequential resource allocation systems using disjunctions of linear classifiers. In *Proceedings of CDC 2012*. IEEE, 2012.

[8] R. Cordone and L. Piroddi. Monitor optimzation in Petri net control. In *Proceedings of the 7th IEEE Conf. on Automation Science and Engineering*, pages 413–418. IEEE, 2011.

[9] E. W. Dijkstra. Cooperating sequential processes. Technical report, Technological University, Eindhoven, Netherlands, 1965.

[10] J. Ezpeleta, F. Tricas, F. Garcia-Valles, and J. M. Colom. A Banker's solution for deadlock avoidance in FMS with flexible routing and multi-resource states. *IEEE Trans. on R&A*, 18:621–625, 2002.

[11] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., New York, NY, 1979.

[12] E. M. Gold. Deadlock prediction: Easy and difficult cases. *SIAM Journal of Computing*, 7:320–336, 1978.

[13] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, MA, 1979.

[14] M. Lawley, S. Reveliotis, and P. Ferreira. The application and evaluation of Banker's algorithm for deadlock-free buffer space allocation in flexible manufacturing systems. *Intl. Jrnl. of Flexible Manufacturing Systems*, 10:73–100, 1998.

[15] Z. Li, M. Zhou, and N. Wu. A survey and comparison of Petri net-based deadlock prevention policies for flexible manufacturing systems. *IEEE Trans. Systems, Man and Cybernetics – Part C: Applications and Reviews*, 38:173–188, 2008.

[16] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77:541–580, 1989.

[17] A. Nazeem and S. Reveliotis. Designing maximally permissive deadlock avoidance policies for sequential resource allocation systems through classification theory. In *Proceedings of the 7th IEEE Conf. on Automation Science and Engineering*, pages 405–412. IEEE, 2011.

[18] A. Nazeem and S. Reveliotis. A practical approach for maximally permissive liveness-enforcing supervision of complex resource allocation systems. *IEEE Trans. on Automation Science and Engineering*, 8:766–779, 2011.

[19] A. Nazeem and S. Reveliotis. Designing maximally permissive deadlock avoidance policies for sequential resource allocation systems through classification theory: the non-linear case. *IEEE Trans. on Automatic Control*, 57:1670–1684, 2012.

[20] A. Nazeem, S. Reveliotis, Y. Wang, and S. Lafortune. Designing maximally permissive deadlock avoidance policies for sequential resource allocation systems through classification theory: the linear case. *IEEE Trans. on Automatic Control*, 56:1818–1833, 2011.

[21] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, NY, NY, 1988.

[22] N. J. Nilsson. *The Mathematical Foundations of Learning Machines*. Morgan Kaufmann, San Mateo, CA, 1990.

[23] S. Reveliotis and E. Roszkowska. On the complexity of maximally permissive deadlock avoidance in multi-vehicle traffic systems. *IEEE Trans. on Automatic Control*, 55:1646–1651, 2010.

[24] S. A. Reveliotis. *Real-time Management of Resource Allocation Systems: A Discrete Event Systems Approach*. Springer, NY, NY, 2005.

[25] V. Vazirani. *Approximation Algorithms*. Springer, NY,NY, 2003.

[26] M. Zhou and M. P. Fanti (editors). *Deadlock Resolution in Computer-Integrated Systems*. Marcel Dekker, Inc., Singapore, 2004.