Logical Control of Complex Resource Allocation Systems

Spyros Reveliotis
School of Industrial & Systems Engineering
Georgia Institute of Technology
Atlanta, GA, 30032
USA

the essence of knowledge
Boston — Delft

Foundations and Trends® in Systems and Control

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

S. Reveliotis. *Logical Control of Complex Resource Allocation Systems*. Foundations and Trends® in Systems and Control, vol. 4, no. 1-2, pp. 1–223, 2017.

This Foundations and Trends® issue was typeset in LaTeX using a class file designed by Neal Parikh. Printed on acid-free paper.

ISBN: 978-1-68083-XXX-Y

Full text available at: http://dx.doi.org/10.1561/2600000010

Foundations and Trends® in Systems and Control

Volume 4, Issue 1-2, 2017

Editorial Scope

Topics

Foundations and Trends® in Systems and Control publishes survey and tutorial articles in the following topics:

- Control of:
  - Hybrid and discrete event systems
  - Nonlinear systems
  - Network systems
  - Stochastic systems
  - Multi-agent systems
  - Distributed parameter systems
  - Delay systems
- Systems
  - Energy storage
  - Grid integration
  - Conversion technologies
  - Underpinning materials developments
- Filtering, estimation, and identification
- Optimal control
- Systems theory
- Control applications

Information for Librarians

Foundations and Trends® in Systems and Control, 2017, Volume 4, 4 issues. ISSN paper version 2325-6818. ISSN online version 2325-6826. Also available as a combined paper and online subscription.

Foundations and Trends® in Systems and Control
Vol. 4, No. 1-2 (2017) 1–223

DOI: 10.1561/2600000010

Logical Control of Complex Resource Allocation Systems

Spyros Reveliotis
School of Industrial & Systems Engineering
Georgia Institute of Technology
Atlanta, GA, 30032
USA

Contents

Abstract

The problem addressed in this document concerns the coordinated allocation of a finite set of reusable resources to a set of concurrently running processes. These processes execute in a staged manner, and each stage requires a different subset of the system resources for its support. Furthermore, processes will hold upon the resources currently allocated to them until they will secure the necessary resources for their next processing stage. Such resource allocation dynamics currently arise in the context of many flexibly automated operations: from the workflow that takes place in various production shop floors and certain internet-supported platforms that seek to automate various service operations; to the traffic coordination in guidepath-based transport systems like industrial monorail and urban railway systems; to the resource allocation that takes place in the context of the contemporary multi-core computer architectures. From a theoretical standpoint, the resource allocation problems that are abstracted from the aforementioned applications, correspond to the problem of scheduling a stochastic network with blocking and deadlocking effects. This is an area of the modern scheduling theory with very limited results. To a large extent, this lack of results is due to the intricacies that arise from the blocking, and especially the deadlocking effects that take place in these networks, and prevents a tractable analysis of these problems through the classical modeling frameworks. Hence, the departing thesis of the work that is presented in this document, is the decomposition of the aforementioned scheduling problems to (i) a supervisory control problem that will seek to prevent the deadlock formation in the underlying resource allocation dynamics, and (ii) a scheduling problem that will be formulated on the admissible subspace to be defined by the adopted supervisory control policy. Each of these two subproblems can be further structured and addressed using some formal modeling frameworks borrowed, respectively, from the qualitative and the quantitative theory of Discrete Event Systems. At the same time, the above two subproblems possess considerable special structure that can be leveraged towards their effective and efficient solution. The presented material provides a comprehensive tutorial exposition of the current achievements of the corresponding research community with respect to the first of the two subproblems mentioned above. As it will be revealed by this exposition, the corresponding results are pretty rich in their theoretical developments and practically potent. At the same time, it is expected and hoped that the resulting awareness regarding the aforementioned results will also set the stage for undertaking a more orchestrated effort on the second of the two subproblems mentioned above.

S. Reveliotis. *Logical Control of Complex Resource Allocation Systems*. Foundations and Trends® in Systems and Control, vol. 4, no. 1-2, pp. 1–223, 2017.
DOI: 10.1561/2600000010.

1

Introduction

As indicated by its title, this document addresses problems pertaining to resource allocation. This is a fundamental concept in the design and operation of many different applications, and therefore, it has been the subject of study in many academic disciplines for a long time. In their basic positioning, resource allocation problems concern the arbitration of the utilization of a finite set of (frequently) reusable resources by a set of contesting processes in a way that promotes certain notions of operational efficiency. Usually, this efficiency is characterized and quantified by some time-based performance criteria, like (i) the maximization of the number of processes served per unit of time – also, known as the (average) throughput in the relevant terminology; (ii) the minimization of the average waiting time experienced by the contesting processes, and of the corresponding congestion that results from the incurred waits; and (iii) the ability to meet effectively pre-specified due dates for the various running processes. Collectively, the resulting problems define an area that is known as "scheduling theory" [116] and has been very conspicuous within the disciplines of Industrial Engineering (IE), Operations Research (OR) and Operations Management (OM). In fact, scheduling theory has also been a subject of study in the fields of theoretical Computer Science (CS) [109], Stochastic Control [8, 93], and even Artificial Intelligence (AI)

[97]. All this research activity has provided a rich body of results for various versions of the underlying resource allocation problems, but it has also established the computational challenges that are posed by a very large number of these problems; in particular, the majority of the formulated scheduling problems has been shown to belong to the class of NP-hard problems [46].

Some operational elements that contribute to the difficulty and the potential intractability of a scheduling problem are: (i) the staged / sequential execution of the involved processes; (ii) the presence of routing flexibility, i.e., the availability of alternative execution paths for certain processes; (iii) the need for coordination among the running processes either in the form of the synchronized execution of certain steps, or of precedence constraints among these steps; (iv) the requirement of an extensive set of resources for the support of any single processing stage; and eventually (v) the randomness that is typically present in the arrival times of the executed processes and in the service times of their processing stages. The OR community has tried to tackle many of the complexities that arise from the aforementioned operational features by abstracting the corresponding structures and dynamics through the concept of the "stochastic network" [31, 54, 93]. Furthermore, in the more recent years, by using some asymptotic analysis techniques that result in a more continuous representation of the involved dynamics,[1] the corresponding research community was able to establish the optimality or near-optimality of certain scheduling policies for various classes of stochastic networks [66, 67, 30, 94, 31, 92, 87, 93].

A particular trait of all the aforementioned analyses and results is the correspondence of the "resource" concept to the notion of a "server" that supports or participates in the execution of a subset of the operations that take place in the considered network. The protocol that governs the allocation of these servers to the various processes and their release is rather simple, and essentially it assumes that the servers can be re-distributed to the running processes in any way that satisfies the requirements of the applied scheduling policy. In many cases, such an assumption is quite natural, given the "active" nature that is assumed for these servers. This assumption also implies that the waiting processes do not interfere with those processes that (are assigned to) receive service. Under such an operational regime, the main concern for

[1] These representations are known as "fluid" or "diffusion" models [31, 54, 93].

a scheduling policy that seeks to maximize the throughput of the underlying system and control the expected delays for the running processes, is the effective utilization of the processing capacity of the various servers, especially those that experience the highest expected workloads and are known as the "bottlenecks" of the underlying stochastic network. In some more technical terms, this last requirement is characterized as the control of the "starvation" that is experienced by the network servers, especially those servers that constitute the bottlenecks of the network.

This work, however, deals with resource allocation problems where the allocated resources are not only the active servers that were described in the previous paragraph, but also the more passive elements that are necessary for the physical staging of the running processes and for the further support of their various processing steps. Some characteristic examples of these new resource types are as follows:

1. In the context of the operations that take place in modern computer-integrated production systems [51], a workpiece that goes through the different workstations of this system must always be staged in a well-defined area, that might be either a buffer slot, or the working table of a certain machine, or a certain position on a material handling device. Furthermore, in many cases, the processed workpieces must be mounted on fixtures that stabilize them in certain ways during their sojourn through the system and facilitate the execution of the operations that take place at the system workstations. The pertinent allocation of all these additional resources is a central function of the system controller and it is critical for the effective support of the extensive levels of automation, integration and autonomy that is expected for these systems.

2. In the context of the automated unit-load material handling systems, like the Automated Guided Vehicle (AGV) systems and the overhead monorail systems that are used in many contemporary production and distribution facilities [56], the system vehicles are forced to move on a specified guidepath network that is defined either by the inherent structure of this system (as in the case of the overhead monorail systems), or it is externally imposed in an effort to isolate the traffic of this particular system from the remaining activity of the facility (which

is typical in the case of the deployed AGV systems). Furthermore, the edges of this guidepath network define a set of "zones" that must be occupied by at most one vehicle at any time, a requirement that seeks to establish a certain level of separation among the traveling vehicles, and thus, the avoidance of collisions and other interference problems among them. The imposition of such a zoning scheme essentially turns the vehicle trip between any two endpoints of the underlying guidepath network into a sequential resource allocation process, where the occupation of every zone that is needed for this trip must be negotiated with a centralized controller that controls the entire traffic in this network [136, 163, 37, 146]. In fact, "zone"-based resource allocation schemes have been proposed recently even for the management of the traffic that is generated by a fleet of free-ranging agents that circulate in a confined area [132]. These agents can be, for instance, a set of robots moving in a confined 2-dim area, or a set of aircrafts or submarines moving in a 3-dim region, and the corresponding "zones" are respectively defined by a number of rectangles or parallelepipeds that tesselate the motion area. Finally, similar control schemes can be envisioned for automated subway and railway systems, and for other automated traffic systems that are contemplated for the future support of urban mass-transport needs [49].

3. In the operational context of multithreaded programming, the various concurrently executing threads typically share a number of resources that are provided by the underlying operating system in the form of registers and other storage locations, I/O devices, data files, etc. In many cases, these resources must be allocated exclusively to the requesting threads, and this allocation is coordinated through the association to each resource unit of a token that is known as a "semaphore" or "mut(ually)-ex(clusive) lock"; a process must acquire the corresponding semaphore before it can access the requested resource [32]. Also, in this operational regime, a process might need to acquire a set of resources for the execution of a single operation, and frequently these resources will not be allocated simultaneously as a "bundle" but are obtained sequentially, one semaphore at a time. Multithreaded programming has been a very popular programming paradigm since the early

days of modern computing, as it enabled time sharing in the operational context of the mainframe computers that were used at that era. More recently, the interest in this programing paradigm has been revived with the advent of the multi-core computer architectures that are prominent on all modern computer platforms [65]. But the semaphore-based resource allocation mechanism that is described in this paragraph can also be applied to the (internet-based) workflow management engines that have been proposed for the automation of various business processes; from the processing of insurance claims, to the backend operations supporting the transactions that take place in (e-)commerce and the banking sector [156, 110].

4. Another example comes from the more avant-garde world of quantum computing [106]. In the corresponding computational environments, the processed information is stored in the quantum states of a number of ionized atoms that are known as "qubits". These qubits are physically stored in certain locations, and they must be transported to some other locations where they will go through a controlled interaction for the execution of the various elementary operations that are supported by the corresponding processors. Furthermore, the transport of the qubits among the various locations is supported by a network of "ion traps", and it must take place in a way that isolates them from the surrounding environment and from each other. This last requirement gives rise to a zone-based traffic control scheme that has a very strong similarity to the traffic that takes place in the zone-based unit-load automated material handling systems that were discussed in item #2 of this list.

A novel element that is introduced by the resource allocation that takes place in the aforementioned examples is that of "blocking": A workpiece that has completed its processing in the current workstation, or an AGV that has completed the traversal of its current zone, might not be able to advance any further at the current time-point, due to the fact that the next requested resource(s) is not currently available. Also, in some other cases like that of multithreaded programming, a blocking effect might arise from the fact that a process has acquired a subset of the resources that are necessary for the execution of its next processing step, but it is still waiting for the allocation of the remaining resources. All these blocked processes hold upon their currently allocated resources, possibly preventing some other process to utilize these resources for its next step.

Frequently, this blocking is a transient phenomenon that is eventually resolved when a certain process instance completes its current processing and advances to its next processing step, releasing the required resources for the remaining blocked processes to advance as well. But it is also possible that, under a general structure for the process sequential logic and the corresponding resource allocation requests, the "hold while waiting" effect that was described in the previous paragraph, will give rise to circular waiting patterns among a subset of the running processes; these patterns are known as "(partial) deadlocks" or "deadly embraces". As both of these terms suggest, the occurrence of circular waiting among some of the running processes will result in (i) the inability of these processes to advance any further in their process plans without some external intervention / interrupting procedure that will resolve this deadlock, and (ii) the waste of the resource units that are involved in this deadlock. Hence, deadlock is an important problem in the operation of the aforementioned applications that must be promptly recognized and resolved for the effective management of these applications.

Past industrial practice has tried to resolve the deadlocks that might arise in the aforementioned application contexts either (i) by adopting simple resource allocation patterns for the sequential logic of the corresponding processes that will not allow the formation of circular dependencies, or (ii) by allowing deadlock to occur and providing the necessary mechanisms for its detection and the recovery from it through the interruption of some of the deadlocked processes. As a case in point of the first approach, we mention the, so called, tandem AGV systems that are currently used in many industrial settings [11]. These systems decompose the underlying traffic into a number of unidirectional loops that are interfaced with a number of buffers. By having all vehicles in each loop moving in the same direction, deadlock is certainly avoided, but at the cost of longer and also slower trips, since traffic is eventually regulated by the slower vehicles. Furthermore, in the case of transports across different loops, there is a need for "double-handling" of the transported material, since it has to be transported by at least two different vehicles; this effect introduces an additional operational cost that could have been avoided by a more agile configuration of the AGV system. The second current approach to deadlock resolution, that relies on their detection and recovery, is most popular among the community of multithreaded programming, since the idea of process interruption and redefinition of its running stage is more easily implementable in that operational context [32]. In fact, such a scheme can be pretty efficient in the cases where deadlocks are rather rare events. But it can be quite disruptive when the coupling and the interaction among the program threads increases through extensive resource sharing.

Based on all the above remarks, it can be effectively argued that the resource allocation functions that were described in the aforelisted examples, and the corresponding industries, can benefit from the development of a control paradigm that will manage the corresponding resource allocation functions in a way that ensures their deadlock-freedom, and at the same time it can support the extent of the automation and the autonomy, as well as the operational concurrency, flexibility and efficiency that are currently sought for these applications. This document provides a methodological base and a set of key results that are currently available for the aforementioned problem.

The departing point for the developments to be presented in this work is the realization that deadlock formation is an effect that results from the *sequencing* of the various resource allocation events that take place in the underlying system, and not by the exact timing of these events. This realization implies that the investigation of the deadlock-related problems that are described in the previous paragraphs, and the effective resolution of these problems, will necessitate a different set of methods and tools than the methods and tools used by the scheduling theory for the analysis and control of the time-based (or *"timed"*) dynamics of the considered resource allocation systems. To acknowledge and highlight this differentiation in the methodological approaches, in the sequel we shall refer to the study of the event sequences that are generated by the considered resource allocation functions and systems, as the *"untimed"* dynamics of these systems (also known as *"logical"* or *"qualitative"* dynamics).

In the context of systems and control theory, the analysis and the control of the "event" sequences that are generated by various natural and engineered event-driven systems, has been the subject of qualitative Discrete Event Systems (DES) theory [17, 162, 149]. Hence, the developments that are presented in this work have sought to leverage and extend the theoretical developments of qualitative DES theory in order to provide rigorous and computationally tractable solutions to the deadlock resolution problem that was described in the previous paragraphs. Using the formal abstraction of the *"(sequential) resource allocation system (RAS)"* [139], and further formal representations borrowed from qualitative DES theory, these developments have provided:

1. a succinct analytical characterization of the considered class of problems;
2. the formulation of a notion of "optimal control" for the corresponding dynamics;
3. the characterization of the computational complexity of the sought optimal solutions (it turns out that for the majority of the considered resource allocation problems, the computation and deployment of the corresponding optimal solution is NP-hard [131], which is another manifestation of the "curse of dimensionality" that haunts most sequential decision-making problems [7]);
4. effective and efficient algorithms that are able to provide optimal and near-optimal deadlock resolution for any practical instantiation from the considered RAS classes, in spite of the negative result of item #3 above;
5. and eventually, a methodological base that can be further leveraged towards the effective scheduling of the considered RAS.

From a more conceptual standpoint, and in line with the basic predications of the DES Supervisory Control (SC) theory [17, 162], the developments that are described in the previous paragraph constitute "preventive control" for the underlying RAS classes; i.e., the corresponding SC policies seek to restrain the dynamics generated by the underlying resource allocation function in order to keep the resultant operation (partial-)deadlock-free. Stated in a different manner, the theory that is presented in this document seeks to confine the original *feasible* behavior of the underlying RAS into a subspace that constitutes the *admissible* behavior, where the latter is characterized by the absence of partial-deadlock. Furthermore, the optimal control problem that was mentioned in the above list intends to specify this admissible behavior in a *maximally permissive* manner. With the admissible RAS behavior well-defined, one can subsequently formulate and address the corresponding scheduling problem over this more restricted behavioral space. Hence, the developments that are presented in this work can eventually facilitate the scheduling of stochastic networks with blocking and deadlocking effects, an area that has received very limited attention in the current literature. Figure 1.1 provides the basic architecture of a real-time, event-driven controller for these networks, that results from the proposed decomposition of the corresponding control problem into a logical control problem and its scheduling counterpart.

In view of the above positioning of the content and the intended contribution of this document, the rest of its chapters are organized as follows: Chapter 2 provides the modeling abstraction of the sequential RAS, and a formal representation of the qualitative RAS dynamics in the modeling framework of the Finite State Automata (FSA) [59, 17]. This chapter also characterizes the corresponding SC problem and the associated notion of "maximal permissiveness", and it reviews a series of results that establish the NP-hardness of the sought maximally permissive SC policy. Chapter 3 overviews a set of results characterizing a number of RAS classes for which the optimal SC policy is of polynomial complexity with respect to (w.r.t.) the size of the underlying RAS. Chapter 4 presents a series of recently developed results that have managed to deploy the maximally permissive DAP for very large RAS instances by isolating the expensive part of the corresponding computation in an off-line stage of the overall deployment process. Instrumental for these developments is the realization that the sought SC policy essentially functions as a classifier that dichotomizes the underlying state space on the basis of the state admissibility. Chapter 5 presents the major results on the considered problem of RAS deadlock resolution that have been derived through the Petri net (PN) modeling framework [98], the second major modeling framework offered by qualitative DES theory. Petri nets can offer more compactness and a higher specificity in the representation of the underlying RAS dynamics, and a richer set of analytical tools for the characterization of the emergent behavior. In particular, they can reveal more succinctly the connection between the underlying RAS structure and the emergent behavioral properties,

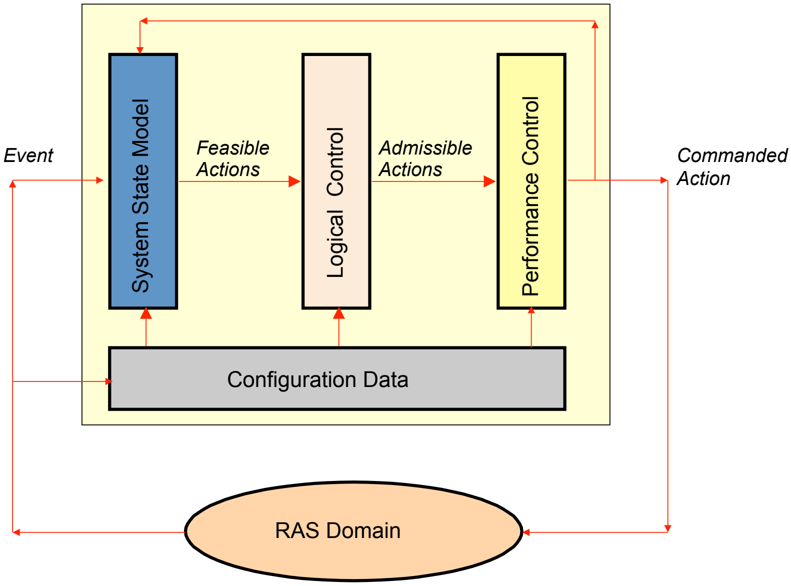

Figure 1.1: An event-driven control scheme for the real-time management of the considered RAS [139]. The controller responds to the various events taking place in the controlled RAS by updating a state model that defines the feasible behavior generated by this system. This behavior is "filtered" through the logical controller in order to obtain the admissible behavior, i.e., the behavior that is consistent with certain specifications imposed on the RAS operation, including the requirement for deadlock-freedom. Finally, the admissible behavior is processed through the performance-oriented controller in order to select the particular action(s) among the admissible behavior that eventually will be commanded upon the RAS.

a line of analysis that is known as "structural analysis" in the relevant PN theory. At the same time, the material of this chapter will reveal a complementarity between the analytical power and capabilities offered by the FSA and the PN modeling frameworks. Chapter 6 complements the fundamental results presented in the previous chapters through a series of refinements and extensions that further enrich the presented theory and augment its applicability. Finally, Chapter 7 concludes the presentation and highlights directions for further extensions and future work. Furthermore, two appendices provide the basic background on the FSA and the PN modeling frameworks that is necessary for the communication of the corresponding results that are presented in the main part of the text.

A note on the adopted notation: In the technical part to be presented in the rest of this document, variables or parameters that are scalar quantities will be denoted by small letters. Vectors will be denoted by boldface small letters, and they will be considered as column vectors. Furthermore, the zero vector will be denoted by $\mathbf{0}$, the vector with all its elements equal to one will be denoted by $\mathbf{1}$, and the unit vector with the nonzero element in its i-th component will be denoted by $\mathbf{1}_i$. Matrices will be denoted by capital letters. Sets and the tuples that define the structured objects that are addressed in this document, will be denoted by capital letters, and they might also be scripted. Transposition of vectors and matrices will be denoted by superscripting these entities by "T". $\mathbb{R}$ will denote the set of real numbers, and $\mathbb{Z}$ will denote the set of integers. Furthermore, $\mathbb{R}^+$ and $\mathbb{R}_0^+$ will respectively denote the sets of strictly positive and the nonnegative reals; and similar notation will be used for the integers. At certain occasions, we shall also set $\mathbb{Z}_0^+ \equiv \mathbb{N}$, in order to emphasize the standard interpretation of this particular set as the set of "natural numbers". We also set $\mathbb{B} \equiv \{0,1\}$, and we shall use this notation in order to characterize the domain of the binary variables that are used in the text. Finally, the application of the notation "$\leq$" on a pair of vectors will imply the component-wise interpretation of this relationship, and the application of the notation "$<$" on a pair of vectors strengthens the "$\leq$" relationship among these vectors by implying that the strict inequality holds for at least one coordinate. The operator $|\cdot|$ when applied on a set returns its cardinality; when applied on a vector returns its l_1 norm; and when applied on one of the structured objects that are defined in the text, returns the "size" of this object (as defined in the text).

1.1 Notes and Sources

The problem of deadlock formation and its effective resolution was first studied in the late 60's and early 70's, in the context of the multi-threaded computation that was emerging at that time. A series of seminal works presented, for instance, in [34, 55, 53, 58] sought to understand the structural elements of the underlying resource allocation function that lead to deadlock formation, and to provide some structural characterizations of these formations that were eventually used primarily for deadlock detection and recovery. The mathematical tools used in these studies were rather *ad hoc* graph-theoretic structures that facilitated the tracing of the existing dependencies among the running processes and the system resources, that were established by the current resource allocation and the posed requests. A particular line of work that sought a more proactive real-time control approach against deadlock formation, and at the same time tried to provide a solution of polynomial computational complexity w.r.t. the size of the underlying system, is the Banker's algorithm that was proposed by Dijkstra in the late 60's [34]. Currently, this algorithm is standard material in any textbook that deals with computer operating systems and concurrent processes; we shall return to this algorithm in Chapter 6 where we discuss and extend this algorithm to render it applicable to the more complex and more dynamic resource allocation functions that are considered in this work. Another theme from that time that is standard textbook material in the literature on computer operating systems and concurrent processes, is the "Dining Philosophers" problem [32], a stylized case study that demonstrates the formation of deadlock due to the sequential acquisition by a set of concurrently executing processes of the resources that are necessary for the execution of a single processing step.

A second seminal set of results on the RAS deadlock resolution problem appeared in the late 70's, in the wake of the major advances in computational complexity theory that occurred at that time. In particular, the works of [3, 50] established the NP-hardness of the optimal deadlock resolution problem that was outlined in the earlier parts of this chapter, and they also sought to specify a boundary between the corresponding hard and easy cases. More recently, these complexity results and the corresponding boundary have been sharpened in [74, 131].

The last distinct wave in the developments on the deadlock resolution problem considered in this work, which also constitutes the major base for the presented material, originated in the late 80's / early 90's. This wave was motivated by the quest for extensive flexibility, automation, integration and autonomy of the operations that take place in the context of various major contemporary applications, including production and distribution, fleets of mobile agents, urban subway and railway systems, and internet-based workflow management systems. These enhanced requirements were, themselves, inspired and facilitated by the dramatic advancement of the computational capabilities of those times. On the theoretical side, all these practical trends were complemented and supported by the emergence of DES SC theory [121] that provided a rich and rigorous analytical base for the formal modeling of the considered resource allocation functions and the investigation of their dynamics. The corresponding literature is too broad to be enumerated exhaustively in this discussion, but we shall visit most of its key developments in the subsequent parts of this work, where we shall also provide the corresponding references. Some pioneering seminal works from this latest era are presented in [158, 5, 164, 35, 143, 38], while various parts of this literature are accumulated and classified in the texts and the survey papers of [139, 169, 82, 81, 15, 123].

Concluding this introductory chapter, we should also notice that, besides the aforementioned developments, the notion of "blocking" has been studied, to a certain extent, by queueing theory. Most of the corresponding results can be traced in the monographs [115, 114, 108]. However, in line with the broader spirit of queueing theory, all these results are of descriptive rather than prescriptive nature; i.e., they try to characterize the impact of any arising blocking and deadlocking effects on the performance of the underlying system, without making any effort to control these effects.

A first formulation of the "companion" control problem to the SC problems that are addressed in this work, regarding the real-time scheduling of the logically controlled RAS depicted in Figure 1.1, can be found in [139]. Also, a first set of results for this problem are presented in [20, 21, 22]. Furthermore, a more recent study of this problem, with a stronger and a more extensive set of results, especially from a practical computational standpoint, can be found in [75, 76, 77]. However, it is generally true that the real-time scheduling of the logically controlled RAS has received only limited attention in the current literature and it is pretty open to further investigation.

References

[1] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modeling with Generalized Stochastic Petri Nets*. John Wiley & Sons, 1994.

[2] K. Akesson, M. Fabian, H. Flordal, and R. Malik. SUPREMICA-an integrated environment for verification, synthesis and simulation of discrete event systems. In *Proceedings of the 8th International Workshop on Discrete Event Systems*, pages 384–385. IEEE, 2006.

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

[4] E. Badouel and P. Darondeau. Theory of regions. In W. Reisig and G. Rozenberg, editors, *LNCS 1491 – Advances in Petri Nets: Basic Models*, pages 529–586. Springer-Verlag, 1998.

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

[6] K. Barkaoui and I. Ben Abdallah. A deadlock prevention method for a class of FMS. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 4119–4124. IEEE, 1995.

[7] R. Bellman. *Applied Dynamic Programming*. Princeton University, Princeton, N. J., 1957.

[8] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. 1*. Athena Scientific, Belmont, MA, 1995.

[9] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. 2 (4th ed.)*. Athena Scientific, Belmont, MA, 2012.

[10] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.

[11] Y. A. Bozer and M. M. Srinivasan. Tandem configurations for AGV systems offer simplicity and flexibility. *Industrial Engineering*, 21:23–27, 1989.

[12] P. Brass. *Advanced Data Structures*. Cambridge University Press, New York, NY, 2008.

[13] R. E. Bryant. Symbolic Boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24(3):293–318, 1992.

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

[15] J. Campos, C. Seatzu, and X. Xie. *Formal Methods in Manufacturing*. CRC Press, London, UK, 2014.

[16] E. E. Cano, C. A. Rovetto, and J. M. Colom. An algorithm to compute minimal siphons in S^4PR nets. *Discrete Event Dynamic Systems: Theory and Applications*, 22:403–428, 2012.

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

[18] 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.

[19] S. F. Chew and M. A. Lawley. Robust supervisory control for production systems with multiple resource failures. *IEEE Transactions on Automation Science and Engineering*, 3:309–323, 2006.

[20] J. Y. Choi. *Performance Modelling, Analysis and Control of Capacitated Re-entrant Lines*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2004.

[21] J. Y. Choi and S. A. Reveliotis. A generalized stochastic Petri net model for performance analysis and control of capacitated re-entrant lines. *IEEE Transactions on Robotics and Automation*, 19:474–480, 2003.

[22] J. Y. Choi and S. A. Reveliotis. Relative value function approximation for the capacitated re-entrant line scheduling problem. *IEEE Transactions on Automation Science and Engineering*, 2:285–299, 2005.

[23] F. Chu and X. Xie. Deadlock analysis of Petri nets using siphons and mathematical programming. *IEEE Transactions on R&A*, 13:793–804, 1997.

[24] V. Chvátal. *Linear Programming*. W. H. Freeman & Co., New York, NY, 1983.

[25] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158. ACM, 1971.

[26] R. Cordone, L. Ferrarini, and L. Piroddi. Enumeration algorithms for minimal siphons in Petri nets based on place constraints. *IEEE Transactions on SMC: Part A*, 35:844–854, 2005.

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

[28] R. Cordone, A. Nazeem, L. Piroddi, and S. A. Reveliotis. Designing optimal deadlock avoidance policies for sequential resource allocation systems through classification theory: existence results and customized algorithms. *IEEE Transactions on Automatic Control*, 58:2772–2787, 2013.

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

[30] J. G. Dai. On positive Harris recurrence of multiclass queueing networks: A unified approach via fluid limit models. *Annals of Applied Probability*, 5:49–77, 1995.

[31] J. G. Dai. Stablity of fluid and stochastic processing networks. Technical Report Miscellanea, No. 9, Dept. of Mathematical Sciences, Unversity of Aarhus, Denmark, 1998.

[32] H. M. Deitel. *Operating Systems*. Addison Wesley, Reading, MA, 1990.

[33] L. E. Dickson. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *American Journal of Mathematics*, 35(4):413–422, 1913.

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

[35] J. Ezpeleta, J. M. Colom, and J. Martinez. A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Transactions on R&A*, 11:173–184, 1995.

[36] 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 Transactions on R&A*, 18:621–625, 2002.

[37] M. P. Fanti. Event-based controller to avoid deadlock and collisions in zone-controlled AGVS. *International Journal of Production Research*, 40:1453–1478, 2002.

[38] M. P. Fanti, B. Maione, S. Mascolo, and B. Turchiano. Event-based feedback control for deadlock avoidance in flexible production systems. *IEEE Transactions on Robotics and Automation*, 13:347–363, 1997.

[39] M. P. Fanti, B. Maione, and B. Turchiano. Event control for deadlock avoidance in production systems with multiple capacity resources. *Studies in Informatics and Control*, 7:343–364, 1998.

[40] M. P. Fanti, B. Maione, and B. Turchiano. Deadlock avoidance policies in production systems via a digraph approach. In J. Campos, C. Seatzu, and X. Xie, editors, *Formal Methods in Manufacturing*, pages 229–258. CRC Press / Taylor & Francis, 2014.

[41] M. P. Fanti, B. Maione, and T. Turchiano. Comparing digraph and Petri net approaches to deadlock avoidance in FMS modeling and performance analysis. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 30:783–798, 2000.

[42] Z. Fei. *Symbolic Supervisory Control of Resource Allocation Systems*. PhD thesis, Chalmers University of Technology, Gothenburg, Sweden, 2014.

[43] Z. Fei, K. Akesson, and S. A. Reveliotis. Symbolic computation and representation of deadlock avoidance policies for complex resource allocation systems with application to multithreaded software. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, IEEE, 2014.

[44] Z. Fei, S. A. Reveliotis, S. Miremadi, and K. Akesson. A BDD-based approach for designing maximally permissive deadlock avoidance policies for complex resource allocation systems. *IEEE Transactions on Automation Science and Engineering*, 12:990–1006, 2015.

[45] F. Garcia-Valles, F. Tricas, J. Ezpeleta, and J. M. Colom. Optimal control of discrete event systems. In *Proceedings of the 4th Workshop on Discrete Event Systems*, pages 88–93. IEE, 1998.

[46] 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.

[47] A. Ghaffari, N. Rezg, and X. Xie. Design of a live and maximally permissive Petri net controller using the theory of regions. *IEEE Transactions on Robotics & Automation*, 19:137–141, 2003.

[48] A. Giua, F. DiCesare, and M. Silva. Generalized mutual exclusion constraints on nets with uncontrollable transitions. In *Proceedings of the 1992 IEEE International Conference on Systems, Man and Cybernetics*, pages 974–979. IEEE, 1992.

[49] A. Giua, M. P. Fanti, and C. Seatzu. Monitor design for colored Petri nets: an application to deadlock prevention in railway networks. *Control Engineering Practice*, 10:1231–1247, 2006.

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

[51] M. P. Groover. *Fundamentals of Modern Manufacturing: Materials, Processes and Systems*. Prentice Hall, Englewood Cliffs, NJ, 1996.

[52] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris. *Queueing Theory (4th ed.)*. Wiley, Hoboken, NJ, 2008.

[53] A. N. Habermann. Prevention of system deadlocks. *Communications of the ACM*, 12:373–377, 1969.

[54] J. M. Harrison. Brownian models of open processing networks: canonical representation of workload. *Annals of Applied Probability*, 10:75–103, 2000.

[55] J. W. Havender. Avoiding deadlock in multi-tasking systems. *IBM Systems Journal*, 2:74–84, 1968.

[56] S. S. Heragu. *Facilities Design (3rd ed.)*. CRC Press, 2008.

[57] G Hoffmann and H. Wong-Toi. Symbolic synthesis of supervisory controllers. In *American Control Conference*, pages 2789–2793, Chicago, IL, USA, 1992.

[58] R. D. Holt. Some deadlock properties of computer systems. *ACM Computing Surveys*, 4:179–196, 1972.

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

[60] F. S. Hsieh. Robutness of deadlock avoidance algorithms for sequential processes. *Automatica*, 39:1695–1706, 2003.

[61] H. Hu, M. Zhou, and Z. Li. Liveness and ratio-enforcing supervision of automated manufacturing systems using Petri nets. *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, 42:392–403, 2012.

[62] Y. Huang, M. Jeng, X. Xie, and S. Chung. Deadlock prevention policy based on Petri nets and siphons. *International Journal of Production Research*, 39:283–305, 2001.

[63] M. V. Iordache and P. J. Antsaklis. *Supervisory Control of Concurrent Systems: A Petri net structural approach*. Birkhäuser, Boston, MA, 2006.

[64] M. Jeng, X. Xie, and M. Y. Peng. Process nets with resources for manufacturing modeling and their analysis. *IEEE Transactions on Robotics & Automation*, 18:875–889, 2002.

[65] T. Kelly, Y. Wang, S. Lafortune, and S. Mahlke. Eliminating concurrency bugs with control engineering. *IEEE Computer*, 42(12):52–60, December 2009.

[66] P. R. Kumar. Scheduling manufacturing systems of re-entrant lines. In D. D. Yao, editor, *Stochastic Modeling and Analysis of Manufacturing Systems*, pages 325–360. Springer-Verlag, 1994.

[67] P. R. Kumar and S. P. Meyn. Duality and linear programs for stability and performance analysis of queueing networks and scheduling policies. *IEEE Transactions on Automatic Control*, 41:4–17, 1996.

[68] H. J. Kushner and G. G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, New York, NY, 2003.

[69] M. A. Lawley, S. A. Reveliotis, and P. Ferreira. Design guidelines for deadlock handling strategies in flexible manufacturing systems. *International Journal of Flexible Manufacturing Systems*, 9:5–29, 1997.

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

[71] M. A. Lawley, S. A. Reveliotis, and P. Ferreira. A correct and scalable deadlock avoidance policy for flexible manufacturing systems. *IEEE Transactions on Robotics & Automation*, 14:796–809, 1998.

[72] M. A. Lawley and W. Sulistyono. Robust supervisory control policies for manufacturing systems with unreliable resources. *IEEE Transactions on R&A*, 18:346–359, 2002.

[73] M. A. Lawley. Deadlock avoidance for production systems with flexible routing. *IEEE Transactions on Robotics & Automation*, 15:497–509, 1999.

[74] M. A. Lawley and S. A. Reveliotis. Deadlock avoidance for sequential resource allocation systems: hard and easy cases. *International Journal of FMS*, 13:385–404, 2001.

[75] R. Li. *Performance Optimization of Complex Resource Allocation Systems*. PhD thesis, ISyE, Georgia Tech, Atlanta, GA, 2016.

[76] R. Li and S. A. Reveliotis. Performance optimization for a class of generalized stochastic Petri nets. *Discrete Event Dynamic Systems: Theory and Applications*, 25:387–417, 2015.

[77] R. Li and S. A. Reveliotis. Designing parsimonious scheduling policies for complex resource allocation systems through concurrency theory. *Discrete Event Dynamic Systems: Theory and Applications*, 26:511–537, 2016.

[78] Y. Li and W. M. Wonham. Deadlock issues in supervisory control of discrete event systems. In *Proceedings of the Conference on Information Sciences and Systems*, pages 57–63, 1988.

[79] Z. Li and M. Zhou. Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems. *IEEE Transactions on SMC – Part A*, 34:38–51, 2004.

[80] Z. Li and M. Zhou. Control of elementary and dependent siphons in Petri nets and their application. *IEEE Transactions on SMC – Part A*, 38:133–148, 2008.

[81] Z. Li and M. Zhou. *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*. Springer Science & Business Media, 2009.

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

[83] H. Liao. *Modeling, Analysis and Control of a Class of Resource Allocation Systems Arising in Concurrent Software*. PhD thesis, University of Michigan, Ann Arbor, MI, 2012.

[84] H. Liao, S. Lafortune, S. A. Reveliotis, Wang. Y., and S. Mahlke. Optimal liveness-enforcing control for a class of Petri nets arising in multithreaded software. *IEEE Transactions on Automatic Control*, 58:1123–1138, 2013.

[85] H. Liao, Y. Wang, H. K. Cho, J. Stanley, T. Kelly, S. Lafortune, S. Mahlke, and S. A. Reveliotis. Concurrency bugs in multithreaded software: Modeling and analysis using Petri nets. *Discrete Event Systems: Theory and Applications*, 23:157–195, 2013.

[86] H. Liao, Y. Wang, J. Stanley, S. Lafortune, S. A. Reveliotis, T. Kelly, and S. Mahlke. Eliminating concurrency bugs in multithreaded software: a new approach based on discrete-event control. *IEEE Transactions on Control, Systems Technology*, 21:2067–2082, 2013.

[87] W. Lin and J. Dai. Maximum pressure policies in stochastic processing networks. *Operations Research*, 53:197–218, 2005.

[88] J. P. Lopez-Grao and J. M. Colom. A Petri net perspective on the resource allocation problem in software engineering. *Transactions on Petri Nets and Other Models of Concurrency V*, pages 181–200, 2012.

[89] J. P. Lopez-Grao, J. M. Colom, and F. Tricas. Structural deadlock prevention policies for flexible manufacturing systems: A Petri net outlook. In J. Campos, C. Seatzu, and X. Xie, editors, *Formal Methods in Manufacturing*, pages 197–228. CRC Press / Taylor & Francis, 2014.

[90] C. Ma and W. M. Wonham. Nonblocking supervisory control of state tree structures. *IEEE Transactions on Automatic Control*, 51(5):782–793, May 2006.

[91] Z. Y. Ma, Z. W. Li, and A. Giua. Design of optimal Petri net controllers for disjunctive generalized mutual exclusion constraints. *IEEE Transactions on Automatic Control*, 60:1774–1785, 2015.

[92] C. Maglaras. Discrete-review policies for scheduling stochastic networks: trajectory tracking and fluid-scale asymptotic optimality. *Annals of Applied Probability*, 10:897–929, 2000.

[93] S. P. Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, Cambridge, UK, 2008.

[94] S. P. Meyn. Stability and optimization of multi-class queueing networks and their fluid models. *Lectures in Applied Mathematics*, 33:175–199, 1997.

[95] S. Miremadi, B. Lennartson, and K. Akesson. A BDD-based approach for modeling plant and supervisor by extended finite automata. *IEEE Transactions on Control Systems Technology*, 20(6):1421–1435, 2012.

[96] J. O. Moody and P. J. Antsaklis. *Supervisory Control of Discrete Event Systems Using Petri Nets*. Kluwer Academic Pub., Boston, MA, 1998.

[97] T. Morton and D. W. Pentico. *Heuristic Scheduling Systems: With Applications to Production Systems and Project Management*. Wiley, New York, NY, 1993.

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

[99] A. Nazeem. *Designing parsimonious representations of the maximally permissive deadlock avoidance policy for complex resource allocation systems through classification theory*. PhD thesis, Georgia Tech, Atlanta, GA, 2012.

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

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

[102] A. Nazeem and S. A. Reveliotis. Efficient enumeration of minimal unsafe states in complex resource allocation systems. *IEEE Transactions on Automation Science & Engineering*, 11:111–124, 2014.

[103] A. Nazeem and S. A. Reveliotis. Maximally permissive deadlock avoidance for resource allocation systems with R/W-locks. *Discrete Event Dynamic Systems: Theory and Applications*, 25:31–63, 2015.

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

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

[106] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, 2010.

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

[108] R. O. Onvural and I. F. Akyildiz. *Queueing Networks with Finite Capacity*. North Holland, 1993.

[109] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algortihms and Complexity*. Dover, Mineola, New York, NY, 1998.

[110] J. Park. A deadlock and livelock free protocol for decentralized internet resource coallocation. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 34:123–131, 2004.

[111] J. Park and S. A. Reveliotis. Algebraic synthesis of efficient deadlock avoidance policies for sequential resource allocation systems. *IEEE Transactions on R&A*, 16:190–195, 2000.

[112] J. Park and S. A. Reveliotis. Liveness-enforcing supervision for resource allocation systems with uncontrollable events and forbidden states. *IEEE Transactions on R&A*, 18:234–240, 2002.

[113] J. Park and S. A. Reveliotis. Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings. *IEEE Transactions on Automatic Control*, 46:1572–1583, 2001.

[114] H. G. Perros. *Queueing Networks with Blocking: Exact and Approximate Solutions*. Oxford University Press, New York, NY, 1994.

[115] H. G. Perros and T. Altiok (Eds.). *Queueing Networks with Blocking*. North Holland, 1989.

[116] M. Pinedo. *Scheduling*. Prentice Hall, Upper Saddle River, NJ, 2002.

[117] L. Piroddi, R. Cordone, and I. Fumagalli. Selective siphon control for deadlock prevention in Petri nets. *IEEE Transactions on SMC: Part A*, 38:1337–1348, 2008.

[118] L. Piroddi, R. Cordone, and I. Fumagalli. Combined siphon and marking generation for deadlock prevention in Petri nets. *IEEE Transactions on SMC: Part A*, 39:650–661, 2009.

[119] W. B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley, New York, NY, 2007.

[120] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

[121] P. J. G. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77:81–98, 1989.

[122] S. A. Reveliotis. Algebraic deadlock avoidance policies for sequential resource allocation systems. In M. Lahmar, editor, *Facility Logistics: Approaches and Solutions to Next Generation Challenges*, pages 235–289. Auerbach Publications, 2007.

[123] S. A. Reveliotis. Coordinating autonomy: sequential resource allocation systems for automation. *IEEE Robotics & Automation Magazine*, June, 2015.

[124] S. A. Reveliotis. On the complexity of some state-counting problems for bounded Petri nets. *IEEE Transactions on Automatic Control*, 60:205–210, 2015.

[125] S. A. Reveliotis. Real-Time Management of Complex Resource Allocation Systems: Necessity, Achievements and Further Challenges. *Annual Reviews in Control*, 41:147–158, 2016.

[126] S. A. Reveliotis and Z. Fei. Invariant-based supervisory control of switched Discrete Event Systems. *IEEE Transactions on Automatic Control*, 62(2):921–927, 2017.

[127] S. A. Reveliotis and Z. Fei. Invariant-based supervisory control of switched Discrete Event Systems. In *The IFAC Conference on Analysis and Design of Hybrid Systems*. IFAC, 2015.

[128] S. A. Reveliotis and Z. Fei. Robust deadlock avoidance for sequential resource allocation systems with resource outages. In *IEEE International Conference on Automation Science and Engineering*. IEEE, 2016.

[129] S. A. Reveliotis and Z. Fei. Robust deadlock avoidance for sequential resource allocation systems with resource outages. Technical Report (submitted for publication), School of Industrial & Systems Engineering, Georgia Tech, 2016.

[130] S. A. Reveliotis and A. Nazeem. Optimal linear separation of the safe and unsafe subspaces of sequential RAS as a set-covering problem: algorithmic procedures and geometric insights. *SIAM Journal on Control and Optimization*, 51:1707–1726, 2013.

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

[132] S. A. Reveliotis and E. Roszkowska. Conflict resolution in free-ranging multi-vehicle systems: A resource allocation paradigm. *IEEE Transactions on Robotics*, 27:283–296, 2011.

[133] S. A. Reveliotis, E. Roszkowska, and J. Y. Choi. Generalized algebraic deadlock avoidance policies for sequential resource allocation systems. *IEEE Transactions on Automatic Control*, 52:2345–2350, 2007.

[134] S. A. Reveliotis, E. Roszkowska, and J. Y. Choi. Correctness verification of generalized algebraic deadlock avoidance policies through mathematical programming. *IEEE Transactions on Automation Science and Engineering*, 7:240–248, 2010.

[135] S. A. Reveliotis. Accommodating FMS operational contingencies through routing flexibility. *IEEE Transactions on R&A*, 15:3–19, 1999.

[136] S. A. Reveliotis. Conflict resolution in AGV systems. *IIE Transactions*, 32(7):647–659, 2000.

[137] S. A. Reveliotis. On the siphon-based characterization of liveness in sequential resource allocation systems. In *Applications and Theory of Perti Nets 2003*, pages 241–255, 2003.

[138] S. A. Reveliotis. Structural analysis of assembly/disassembly resource allocation systems. In *Proceedings of the International Conference on Robotics and Automation 2003*. IEEE, 2003.

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

[140] S. A. Reveliotis. A necessary and sufficient condition for the liveness and reversibility of process-resource nets with acyclic, quasi-live, serialisable and reversible process subnets. *IEEE Transactions on Automation Science and Engineering*, 3:462–468, 2006.

[141] S. A. Reveliotis. Implicit siphon control and its role in the liveness enforcing supervision of sequential resource allocation systems. *IEEE Transactions on SMC: Part A*, 37:319–328, 2007.

[142] S. A. Reveliotis and J. Y. Choi. Designing reversibility-enforcing supervisors of polynomial complexity for bounded Petri nets through the theory of regions. In *Proceedings of International Conference on Application and Theory of Petri Nets*, pages 322–341, 2006.

[143] S. A. Reveliotis and P. M. Ferreira. Deadlock avoidance policies for automated manufacturing cells. *IEEE Transactions on Robotics & Automation*, 12:845–857, 1996.

[144] S. A. Reveliotis, M. A. Lawley, and P. M. Ferreira. Polynomial complexity deadlock avoidance policies for sequential resource allocation systems. *IEEE Transactions on Automatic Control*, 42:1344–1357, 1997.

[145] E. Roszkowska. Supervisory control for deadlock avoidance in compound processes. *IEEE Transactions on SMC – Part A*, 34:52–64, 2004.

[146] E. Roszkowska and S. A. Reveliotis. On the liveness of guidepath-based, zoned-controlled, dynamically routed, closed traffic systems. *IEEE Transactions on Automatic Control*, 53:1689–1695, 2008.

[147] E. Roszkowska and S. A. Reveliotis. A distributed protocol for motion coordination in free-ranging vehicular systems. *Automatica*, 49:1639–1653, 2013.

[148] E. Roszkowska and R Wojcik. Problems of process flow feasibility in flexible assembly systems. In *CIM in Process and Manufacturing Industry*, pages 115–120. Pergamon Press, 1993.

[149] C. Seatzu, M. Silva, and J. H. van Schuppen. *Control of Discrete Event Systems: Automata and Petri Net Perspectives*. Springer, London, UK, 2013.

[150] M. Silva, E. Teruel, and J. M. Colom. Linear algebraic and linear programming techniques for the analysis of place/transition net systems. In W. Reisig and G. Rozenberg, editors, *Lecture Notes in Computer Science, Vol. 1491*, pages 309–373. Springer-Verlag, 1998.

[151] R. Song and R. J. Leduc. Symbolic Synthesis and Verification of Hierarchical Interface-based Supervisory Control. In *8th Discrete Event Systems, WODES '06*, pages 419–426, Ann Arbor, MI, July 2006.

[152] J. Stanley, H. Liao, and S. Lafortune. SAT-based control of concurrent software for deadlock avoidance. *IEEE Transactions on Automatic Control*, 60:3269–3274, 2015.

[153] F. Tricas, F. Garcia-Valles, J. M. Colom, and J. Ezpeleta. A structural approach to the problem of deadlock prevention in processes with resources. In *Proceedings of the 4th Workshop on Discrete Event Systems*, pages 273–278. IEE, 1998.

[154] F. Tricas, F. Garcia-Valles, J. M. Colom, and J. Ezpeleta. A Petri net structure-based deadlock prevention solution for sequential resource allocation systems. In *Proceedings of the ICRA 2005*, pages 271–277. IEEE, 2005.

[155] A. Vahidi, M. Fabian, and B. Lennartson. Efficient supervisory synthesis of large systems. *Control Engineering Practice*, 14(10):1157–1167, 2006.

[156] W. Van der Aalst and K. Van Hee. *Workflow Management: Models, Methods and Systems*. The MIT Press, Cambridge, MA, 2002.

[157] V. Vazirani. *Approximation Algorithms*. Springer, New York, NY, 2003.

[158] N. Viswanadham, Y. Narahari, and T. L. Johnson. Deadlock avoidance in flexible manufacturing systems using Petri net models. *IEEE Transactions on Robotics and Automation*, 6:713–722, 1990.

[159] S. G. Wang, C. Y. Wang, M. Z. Zhou, and Z. W. Li. A method to compute strict minimal siphons in a class of Petri nets based on loop resource subsets. *IEEE Transactions on Systems, Man & Cybernetics – Part A*, 42:226–237, 2012.

[160] Y. Wang, T. Kelly, M. Kudlur, S. Lafortune, and S. Mahlke. Gadara: Dynamic deadlock avoidance for multithreaded programs. In *Symposium on Operating Systems Design and Implementation*, 2008.

[161] Y. Wang, S. Lafortune, T. Kelly, M. Kudlur, and S. Mahlke. The theory of deadlock avoidance via discrete control. In *Symposium on Principles of Programming Languages*, 2009.

[162] W. M. Wonham. Supervisory control of discrete event systems. Technical Report ECE 1636F / 1637S 2013-14, Electrical & Computer Engineering, University of Toronto, 2006.

[163] N. Wu and M. Zhou. Resource-oriented Petri nets in deadlock avoidance of AGV systems. In *Proceedings of the International Conference on Robotics and Automation*, pages 64–69. IEEE, 2001.

[164] R. A. Wysk, N. S. Yang, and S. Joshi. Detection of deadlocks in flexible manufacturing cells. *IEEE Transactions on Robotics and Automation*, 7:853–859, 1991.

[165] R. A. Wysk, N. S. Yang, and S. Joshi. Resolution of deadlocks in flexible manufacturing systems: Avoidance and recovery approaches. *Journal of Manufacturing Systems*, 13:128–138, 1994.

[166] K. Y. Xing, B. S. Hu, and H. X. Chen. Deadlock avoidance policy for Petri net modeling of flexible manufacturing systems with shared resources. *IEEE Transactions on Automatic Control*, 41:289–295, 1996.

[167] K. Yamalidou, J. Moody, M. D. Lemmon, and P. J. Antsaklis. Feedback control of Petri nets based on place invariants. *Automatica*, 32:15–28, 1996.

[168] H. Yue, K. Y. Xing, and Z. Hu. Robust supervisory control policy for avoiding deadlock in automated manufacturing systems with unreliable resources. *International Journal of Production Research*, 52:1573–1591, 2014.

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