2018 Annual American Control Conference (ACC)
June 27–29, 2018. Wisconsin Center, Milwaukee, USA

Throughput maximization of capacitated re-entrant lines through fluid relaxation

Michael Ibrahim and Spyros Reveliotis
School of Industrial & Systems Engineering
Georgia Institute of Technology
{michael.ibrahim@,spyros@isye.}gatech.edu

Abstract— **This paper extends the scheduling methodology for complex stochastic networks that is based on the solution of a "fluid" relaxation at each decision point of the original scheduling problem, to stochastic networks with blocking and deadlocking effects. For a clearer and more concrete treatment, the presented results are developed in the operational context of a re-entrant line with finite buffering capacity at each workstation; these re-entrant lines are characterized as "capacitated re-entrant lines (CRLs)". From a methodological standpoint, the paper results are enabled by a pre-established ability to control the underlying resource allocation for deadlock freedom, and by the further ability to express the corresponding deadlock avoidance policy as a set of linear inequalities on the system state. Also, the employed LP relaxation differs considerably from similar past developments, since it must account for the blocking effects that take place in the considered CRLs. A small example provided at the last part of the paper highlights all the aforementioned developments, and helps assessing their efficacy.**

Keywords – **Stochastic scheduling, stochastic networks with blocking and deadlocking effects, liveness-enforcing supervision, fluid relaxation**

I. Introduction

While the scheduling of complex stochastic networks is a very thriving area in the context of the controls and operations research (OR) literature [1], [2], [3], the particular problem of scheduling such stochastic networks with extensive blocking and deadlocking effects has received very limited attention. We believe that this reality is due to the fact that the effective scheduling of this particular class of stochastic networks requires the resolution of an additional type of problem that concerns the establishment of "liveness" for the underlying workflow, i.e., the ability of all the activated jobs to proceed to their completion securing successfully all the required resources for the execution of their various processing stages, and avoiding the formation of any deadlocks or livelocks. On the other hand, this problem of liveness-enforcement has received extensive attention within a certain part of the controls community during the past decades, and the currently available results provide a broad range of methods and policies able to support the necessary supervision for the aforementioned class of stochastic networks, in a way that is, both, operationally and computationally efficient [4]. Hence, motivated by the previous remarks, this work seeks to develop a methodology for effective and computationally efficient scheduling of stochastic networks with blocking and deadlocking effects, combining the existing liveness-enforcing supervision theory for this class of systems with some scheduling methodology that has already been developed by the OR community for some other classes of stochastic networks that do not present any blocking in their workflow dynamics.

More specifically, the scheduling methodology pursued in this work seeks to identify an optimized action at each decision point by formulating and solving a linear program (LP) that retains substantial information about the underlying system dynamics and will be referred to as the corresponding "LP relaxation". As already mentioned, the idea of determining a scheduling policy through the formulation and solution of pertinent LP relaxations has been pursued in the past in the context of some other classes of stochastic networks; the reader is referred to [5], [2], [6], [7], [8], [9], [10] for specific examples. However, to the best of our knowledge, this is the first work to extend the "LP relaxation"-based scheduling methodology to workflows with blocking and deadlocking effects. The extension is non-trivial since, as explained above, the new models and formulations must account for (i) the blocking effects that take place in these stochastic networks, and (ii) the stipulations of the employed liveness-enforcing supervisor (LES – also known as the applied deadlock avoidance policy, or DAP, for short). We address systematically all these issues in the later parts of the paper. But for better clarity and specificity, our results are presented in the particular operational context of re-entrant lines with finite buffering capacity at each workstation; we shall refer to these re-entrant lines as "capacitated re-entrant lines (CRLs)".[1] Furthermore, the last part of the paper provides a small example that demonstrates more vividly the salient points of the underlying methodology, and reveals its efficacy. On the other hand, the imposed space limitations do not allow an expansive treatment of the presented material; such a treatment can be found in [12], which also presents a set of numerical experiments that assess more thoroughly the efficacy and the computational efficiency of the presented method.[2]

In view of the above positioning of the paper content and its intended contributions, the rest of it is organized as follows: Section II introduces the considered CRL model and the corresponding throughput maximization problem. This section also reviews the main results from the control theory of liveness-enforcing supervision that are necessary for the complete definition of the addressed CRL scheduling problem, and expresses this scheduling problem as an average-reward continuous-time MDP. Subsequently, Section III presents the scheduling methodology that is pursued in this work, and Section IV provides the highlighting

[1] The uncapacitated re-entrant line (RL) has received extensive attention in the scheduling literature since it gives rise to challenging scheduling problems while retaining some of the operational simplicity and analytical tractability of a manufacturing flowline [11], [1].

[2] This manuscript is accessible through the personal website of the second author.

978-1-5386-5428-6/$31.00 ©2018 AACC

example. Finally, Section V concludes the paper and points out some directions for future work.

II. The capacitated re-entrant line and the corresponding throughput maximization problem

The considered CRL model: The capacitated re-entrant line to be considered in this work consists of L single-server workstations, $W_1, W_2, \ldots, W_L$, each possessing finite buffering capacity B_i, $i = 1, \ldots, L$. The line supports a single process plan with M processing stages, $J_1, J_2, \ldots, J_M$. Each processing stage J_j is carried out at one of the line workstations which will be denoted by $W(J_j)$. We further assume that $L < M$, an assumption that manifests the re-entrant nature of the line.

Some additional assumptions regarding the structure and the operation of the considered CRL, that will facilitate the exposition of the developments that are presented in this work, are as follows: The buffering capacity B_i of each workstation W_i is provided as B_i buffer slots, with each slot being able to hold a single part. Furthermore, parts visiting the workstation are processed within their allocated buffer slot, with the workstation server visiting this buffer slot. Hence, any part visiting this workstation will remain in its allocated slot during its entire sojourn at this station, and at any time point during this sojourn, the part will either be waiting for processing, be in processing, or will have completed processing and it will be waiting for transfer to the next required workstation. Furthermore, completed parts can move to the next required workstation only when there is available capacity at this workstation.[3]

Processing times for each processing stage J_j are assumed to be exponentially distributed with mean processing time τ_j. And we shall also set $\mu_j \equiv 1/\tau_j,\ \forall j$.[4] Furthermore, part loading and transfer times between the line workstations are assumed to be negligible with respect to (w.r.t.) processing times.[5]

Finally, since in the following developments our primary objective is the throughput maximization of the line, we also assume the existence of an "infinite backlog" of parts waiting for processing in front of the line; hence, the line never starves for work.

Abstracting the CRL "untimed" dynamics through a finite state automaton: Following the relevant theory presented in [4], we model the basic structure of the workflow dynamics of the introduced CRL model, and the corresponding resource allocation function, by means of a finite state automaton (FSA) Φ [14]. A pertinent definition of a notion of *state* for this automaton can be based on the number of the parts waiting for processing, being processed or having completed processing of the different processing stages, J_j, of the supported process plan. More specifically:

Definition 2.1: The state $\mathbf{s}$ of the CRL model considered in this work is a $3M$-dimensional vector with component $3j + k,\ j = 0, \ldots, M-1,\ k = 1, 2, 3$, denoting respectively the number of parts that are waiting for processing, executing, or having completed processing stage J_j.

In the following, we shall denote the resulting state space by S. The set of *events*, E, that advance state $\mathbf{s}$, consists of (i) the event e^l that loads a new part on the line; (ii) the events $e^a_j,\ j = 1, \ldots, M-1$, that advance a part from workstation $W(J_j)$ to the next requested workstation, $W(J_{j+1})$, allocating to this part a free buffer slot of the new workstation; (iii) the events $e^p_j,\ j = 1, \ldots, M$, that initiate the processing a part at workstation $W(J_j)$ by allocating to it the corresponding server; (iv) the events $e^d_j,\ j = 1, \ldots, M$, that de-allocate the server upon completion of the part processing; and (v) the event e^u that unloads a completed part from the line. Events of type (i), (ii), (iii) and (v) can be executed in zero time w.r.t. their activation, and their execution is *controllable* by the line supervisor. On the other hand, events of type (iv) occur spontaneously upon the completion of the processing of the corresponding part, and therefore, they will be treated as *uncontrollable* events. Furthermore, there is a nonzero lag between a type (iii) event and the execution of the corresponding type (iv) event that represents the necessary processing time. Finally, since, in the considered CRL model, part loading and unloading require zero time, and there is an infinite backlog of jobs waiting for processing, it is possible to simplify the state concept introduced in Definition 2.1 by dropping the first and the last component of the state vector $\mathbf{s}$; i.e., parts seeking to execute their first processing stage can be loaded into the line only when the corresponding server is available, and parts having completed processing of the last processing stage can be unloaded immediately. In the following, we shall adopt this simplified state model, with the necessary adjustments in the corresponding notation.

The initial and also the only marked state of the abstracted FSA Φ is state $\mathbf{s}_0 = \mathbf{0}$, i.e., the state where the line is empty of any parts. However, as remarked in the introductory section, the ability of the considered CRL to reach its marked state $\mathbf{s}_0$ can be compromised by the formation of *deadlock*. In the considered operational context, deadlock results when a set of workstations have their buffer slots fully allocated to parts requesting a workstation in this same set for their advancement to their next processing stage. CRL states containing such deadlocks must be actively identified and blocked during the line operation. This task necessitates the deployment of a pertinent *deadlock avoidance policy (DAP)*, and the effective development of such a DAP for any given CRL configuration is supported by the relevant supervisory control theory presented in [4]. Next, we overview some basic developments in [4] that are particularly relevant to the needs of this work.

Establishing deadlock freedom for the considered CRL model: Since, in the considered CRL context, deadlock is due only to the allocation of the workstation buffering capacity (and not to the allocation of the processing capacity of the line servers), the corresponding problem of deadlock avoidance can be focused on this particular allocation. This can be achieved by considering the further abstraction of the

[3] This statement will be further qualified in the later parts of this section in order to account for the need for deadlock avoidance and the additional restrictions that are enforced by the adopted DAP.

[4] While the assumption of exponential processing times is meant to simplify the exposition of the theory that is developed in this paper, more generally distributed processing times can be handled by approximating them by phase-type distributions to any desired degree of accuracy; please, c.f. to [13] for an introduction to phase-type distributions, and to [14] for a brief introduction on the modeling of non-Markovian dynamics by phase-type distributions.

[5] Non-zero loading and transfer times can be easily included in the considered model through the addition of further stages in the underlying process plan.

FSA Φ, that was introduced in the previous subsection, to the FSA $\hat{\Phi}$ with a (vector) state $\hat{\mathbf{s}}$ that considers collectively all the parts located at workstation $W(J_j),\ j = 1, \ldots, M$, for the execution of the corresponding processing stage J_j (i.e., state $\hat{\mathbf{s}}$ will not discriminate whether these parts are waiting for processing, are in processing, or have completed processing and are waiting for transfer to the next required workstation). Also, we shall let $\hat{S}$ denote the entire state space of FSA $\hat{\Phi}$. Similarly, the event set $\hat{E}$ of $\hat{\Phi}$ will consist only of the type (i), type (ii) and type (v) events of the original FSA Φ. Finally, the initial and also unique marked state of $\hat{\Phi}$ will be state $\hat{\mathbf{s}}_0 = \mathbf{0}$, representing the empty state of the underlying CRL.

We shall further denote by $\hat{S}_r$ the set of *reachable* states of $\hat{\Phi}$, i.e., the states $\hat{\mathbf{s}} \in \hat{S}$ that are accessible from state $\hat{\mathbf{s}}_0$ through some feasible event sequence $\sigma \in \hat{E}^*$.[6] On the other hand, state set $\hat{S}_s$ will denote the set of *co-reachable* – or *"safe"* – states of $\hat{\Phi}$, i.e., the states $\hat{\mathbf{s}} \in \hat{S}$ from which state $\hat{\mathbf{s}}_0$ is accessible through some feasible event sequence $\sigma' \in \hat{E}^*$. We shall also set $\hat{S}_{\bar{r}} \equiv \hat{S} \setminus \hat{S}_r$ and $\hat{S}_{\bar{s}} \equiv \hat{S} \setminus \hat{S}_s$, and we shall refer to these two sets, respectively, as the sets of the *unreachable* and the *unsafe* states. Finally, we shall also use the notation $\hat{S}_{xy} \equiv \hat{S}_x \cap \hat{S}_y$, for $x \in \{r, \bar{r}\}$ and $y \in \{s, \bar{s}\}$.

In the *state transition diagram (STD)* $\hat{\mathcal{G}}$ representing the dynamics of FSA $\hat{\Phi}$, the subgraph that is induced by the state set $\hat{S}_{rs}$ is the maximal strongly connected component of $\hat{\mathcal{G}}$ containing the empty state $\hat{\mathbf{s}}_0$. Hence, in order to avoid deadlock while imposing the minimal possible restriction on the CRL behavior that is represented by FSA $\hat{\Phi}$, we need to identify and block transitions from subspace $\hat{S}_{rs}$ to subspace $\hat{S}_{r\bar{s}}$. The resulting DAP will be characterized as *maximally permissive*, it is uniquely defined, and it will be denoted by Δ^*.

The work of [4] provides a complete methodology for the effective deployment of the optimal DAP Δ^* for any instantiation of the CRL model that is considered in this work, as well as efficient approximations of this DAP that are guaranteed to take the form of a small set of linear inequalities imposed upon the FSA state $\hat{\mathbf{s}}$. In the next section, we exploit this linear representation of the applied DAP Δ in order to develop a scheduling method for the considered CRLs through a pertinent "fluid relaxation" of the underlying dynamics.

The problem of throughput maximization of the considered CRL model: The application of the aforementioned DAPs, Δ, on any given CRL configuration will restrict the line operation from the original subspace $\hat{S}_r$ to the policy-admissible subspace $\hat{S}_a$. Furthermore, for a correctly defined DAP Δ, the state set $\hat{S}_a$ is a subset of $\hat{S}_{rs}$, the set of reachable and safe states, and it induces a strongly connected component, $\hat{\mathcal{G}}_a$, of the STD $\hat{\mathcal{G}}$, that contains the empty state $\hat{\mathbf{s}}_0$. Policy Δ can be "lifted" to the original FSA Φ that models more completely the operation of the underlying CRL, through a state admission rule that will admit a state $\mathbf{s} \in S$ if and only if (*iff*) the corresponding state $\hat{\mathbf{s}}$ belongs in $\hat{S}_a$; the resulting admissible subspace of S will be denoted by S_a, and the subgraph $\mathcal{G}_a$ induced by the state set S_a in the STD $\mathcal{G}$ of the FSA Φ has similar connectivity properties to the connectivity properties of the subgraph $\hat{\mathcal{G}}_a$ w.r.t. the STD $\hat{\mathcal{G}}$. Finally, the notions of "(state) reachability" and "co-reachability / safety" are naturally extended to the CRL dynamics that are described by the FSA Φ.

For the pursued formulation of the CRL throughput maximization problem, it is pertinent to differentiate the states $\mathbf{s} \in S_a$ into states where the only enabled events are some uncontrollable events e_j^d, and states that enable controllable events as well.[7] States belonging in the first subclass of S_a essentially define an exponential race among the enabled events e_j^d, and therefore, they will possess a non-zero sojourn time. Hence, following standard terminology in the corresponding literature [15], we characterize these states as *tangible*, and the corresponding state set will be denoted by S_a^T. On the other hand, for those states that enable controllable events, it is expected that the line supervisor will select one of these events for immediate execution. This assumption, combined with the aforestated assumptions about the CRL operation in the earlier parts of this section, imply that the states $\mathbf{s}$ belonging in this subclass of S_a will have zero sojourn times. Therefore, these states are characterized as *vanishing*, and the corresponding state set will be denoted by S_a^V. The enhanced permissiveness of the applied DAP Δ further implies that, for a large number of states $\mathbf{s} \in S_a^V$, there will be more than one enabled controllable event. Also, many of these events will be conflicting, in that the execution of one of these events in state $\mathbf{s}$ will lead to a state $\mathbf{s}'$ where (some of) the remaining events will not be enabled anymore. These conflicts must be managed by the line controller, and the corresponding choices define the applied *scheduling policy*.

The resulting problem of determining a pertinent scheduling policy for the throughput maximization of any given CRL configuration that is controlled by an applied DAP Δ, can be formulated as an average-reward continuous-time MDP (CT-MDP) [16]. The decision epochs of this CT-MDP are the time points of the occurrence of an event e_j^d at any of the admissible tangible states $\mathbf{s} \in S_a^T$. Let $\mathbf{s}'$ denote the resulting state. At state $\mathbf{s}'$, the line supervisor must select a sequence of controllable events, σ, that will lead the line to a new admissible tangible state $\mathbf{s}'' \in S_a^T$. In the following, we shall refer to the set of admissible tangible states $\mathbf{s}''$ that are reachable from state $\mathbf{s}'$ as the *"tangible reach"* of state $\mathbf{s}'$, and we shall denote this set by $\mathcal{TR}(\mathbf{s}')$. Then, it is clear from the above discussion that, in the corresponding CT-MDP terminology, $\mathcal{TR}(\mathbf{s}')$ denotes the set of *"actions"*, a, that are feasible and admissible at the considered state $\mathbf{s}'$. Furthermore, for future reference, we shall characterize the vanishing states $\mathbf{s}'''$ that are reachable from state $\mathbf{s}'$ through the controllable event sequences σ leading to some state $\mathbf{s}'' \in \mathcal{TR}(\mathbf{s})$, as the *"vanishing reach"* of state $\mathbf{s}'$; this set of states will be denoted by $\mathcal{VR}(\mathbf{s}')$. [8]

The transitional dynamics for the CT-MDP model that result from the execution of an action $a \in \mathcal{TR}(\mathbf{s}')$ at some decision state $\mathbf{s}'$, are determined by the exponential race that takes place in the tangible state $\mathbf{s}''$ that corresponds

[6] We remind the reader that in the relevant automata theory, E^* denotes the *Kleene closure* of the event set $\hat{E}$; i.e., E^* contains all the finite-length sequences σ of the elements of E, including the empty sequence ϵ.

[7] The application of the DAP Δ ensures that every state $\mathbf{s} \in S_a$ will possess at least one enabled event that is also admissible by the applied DAP Δ.

[8] In general, $\mathcal{VR}(\mathbf{s}')$ can be empty for some states $\mathbf{s}'$.

to action a. Furthermore, the expected immediate reward, $r(\mathbf{s}', a)$, from executing action a at state $\mathbf{s}'$ is equal to the probability that the next decision epoch will be defined by the occurrence of event e^d_M, that corresponds to the completion by a part of its last processing stage and its unloading from the line. Hence, letting $\mathcal{E}(\mathbf{s}'')$ denote the set of the events e^d_j enabled in $\mathbf{s}''$, the expected immediate reward $r(\mathbf{s}', a)$ will be equal to $\mu_M / \sum_{e^d_j \in \mathcal{E}(\mathbf{s}'')} \mu_j$ if $e^d_M \in \mathcal{E}(\mathbf{s}'')$, and zero otherwise. Finally, as already stated at the beginning of this paragraph, with this definition of the expected immediate rewards, the problem of maximizing the throughput of the considered CRLs is reduced to the problem of maximizing the (long-term) average reward of the corresponding CT-MDP.[9]

Some further simplification of this CT-MDP formulation, and some methodology for its solution through uniformization [14], are presented in Appendix A of [19]. But, in most practical cases, the solution of this CT-MDP model will be intractable due to the very large size of the involved state spaces. Hence, in the next section we present a heuristic methodology that can lead to a good approximation of the corresponding optimal scheduling policies; the quality of this approximation is empirically demonstrated through the example of Section IV, and it is further assessed in [12].

III. The proposed scheduling method

The CRL scheduling methodology considered in this section is motivated by ideas similar to those pursued in [20], [21], [8] for the scheduling of uncapacitated re-entrant lines and more general multi-class queueing networks. In particular, similar to those earlier approaches, we consider the material flowing through the underlying CRL as a continuous quantity, and at every decision epoch, we formulate and solve a linear program (LP) that maximizes the volume of the material processed by this CRL over a sufficiently long time horizon T, when the line is started at the state $\mathbf{s}$ that corresponds to the current decision epoch. Subsequently, we use the obtained LP solution in order to define an action-selection criterion for the current decision epoch. But while the above basic scheme of our method parallels the corresponding schemes of [20], [21], [8], its implementational details in the current work differ considerably from those earlier implementations, since (i) we are addressing a different scheduling objective, and more importantly, (ii) we need to effectively address the blocking and deadlocking phenomena that arise in the operational context of the considered CRL.

A primary differentiator of our fluid model that is motivated by the need to capture in it more effectively the impact of the blocking that is caused by the finite buffers and the imposed DAP, is that it is formulated in discrete and not in continuous time. In particular, assuming that the processing times τ_j for the different processing stages $J_j,\ j = 1, \ldots, M$, are rationally valued, we set the discretizing time interval Δt equal to the greatest common divisor (GCD) of τ_j. In this way, the mean processing time, τ_j, of any processing stage J_j, corresponds to an integral multiple of Δt, which will be denoted by $\hat{\tau}_j$. In the following discussion, we also scale time by further assuming that $\Delta t = 1.00$, and thus, $\hat{\tau}_j$ also denotes the mean processing time of processing stage J_j in this new time scale. The rest of this section discusses the details of the employed LP relaxation and of the action selection scheme that is induced by the LP solution.

The employed LP relaxation: We proceed with the presentation of the employed LP relaxation, by first introducing some supporting notation. Then, we proceed with the introduction of the decision variables, the constraints and the objective function.

Supporting notation:

- $\mathcal{J}_l,\ l = 1, ..., L$: The set of all processing stages executed on workstation WS_l; i.e., $\mathcal{J}_l = \{j : W(J_j) = l, j = 1, \ldots, M\}$.
- T: The total time horizon over which we are maximizing the line throughput; as explained in the opening part of this section, T is expressed in terms of the discretizing time interval Δt.
- $\mathbf{s}^{init}$: The CRL vanishing state that corresponds to the current decision epoch.
- $\mathbf{v}$: A $2M$-dim vector with its components $\mathbf{v}_{1+2j},\ j = 0, \ldots, M-1$, representing the volume of "fluid" waiting for the execution of processing stage J_{j+1} at the corresponding workstation $W(J_{j+1})$, and the components $\mathbf{v}_{2+2j},\ j = 0, \ldots, M-1$, representing the volume of "fluid" that has completed the execution of processing stage J_{j+1} but it is still located at the corresponding workstation $W(J_{j+1})$. We shall refer to the components of vector $\mathbf{v}$ as the corresponding "fluid buffers".
- $\mathbf{v}^{init}$: The initial value for the "buffer fluid" vector $\mathbf{v}$ as defined by the state vector $\mathbf{s}^{init}$. Due to the presumed exponential nature for the distribution of the various processing times, components $\mathbf{v}_{1+2j},\ j = 0, \ldots, M-1$, will aggregate all the parts that either wait for the initiation of the execution of the corresponding processing stage J_{j+1} or have already initiated the execution of this processing stage.
- f: A fictitious "fluid feeder" at the beginning of the line representing an "infinite backlog".
- d: A fictitious "fluid buffer" at the end of the CRL, of unlimited capacity, that collects all the "fluid" that is output by this line over the considered time horizon T.

Decision Variables:

- $x_{j,t}, j = 1, \ldots, 2M, t = 1, \ldots, T$: The "fluid" volume in "fluid buffer" $\mathbf{v}_j$ at the end of period t.
- $u_{j,t}, j = 1, \ldots, 2M+1, t = 1, \ldots, T$: The amount of the "fluid" that is added, during period t, to the "fluid" buffer $\mathbf{v}_j$ or, in the case of $j = 2M+1$, to the "output fluid buffer" d. More specifically:
  - $u_{1,t}$ represents the amount of "fluid" that is added to the "fluid buffer" $\mathbf{v}_1$ at period t. This "fluid" is drawn from the external "fluid feeder" f, during the same period, and its addition to the "fluid

[9] The specification of the set of actions at each decision state $\mathbf{s}'$ of this CT-MDP through the corresponding tangible reach $\mathcal{TR}(\mathbf{s}')$ implies a *non-idling* scheduling policy for the underlying CRL; i.e., under such a policy, no server that could be engaged in the processing of some available part will remain idle. Due to the blocking experienced in the operation of the considered CRLs, such a non-idling scheme might be suboptimal [17], [18]. We have opted to confine the presented developments within the class of the non-idling scheduling policies, in an effort to attain some simplicity for the presentation of the main concepts and ideas involved. But it is possible to extend the presented methodology to *deliberately idling* schemes, by introducing further actions at the states $\mathbf{s}'$ that correspond to decision epochs; these actions will correspond to controllable-event sequences σ leading to some state $\mathbf{s}''$ in the vanishing reach $\mathcal{VR}(\mathbf{s}')$ of the considered state $\mathbf{s}'$, that contains some enabled events e^d_j.

buffer" $\mathbf{v}_1$ is equivalent to the action of loading new material to the CRL.
- $u_{2+2i,t}, i = 0, ..., M-1$, represent the amount of "fluid" that is added to the corresponding "fluid buffer" $\mathbf{v}_{2+2i}$ at period t. This "fluid" corresponds to material completing the processing of processing stage J_{i+1}, and it was drawn from "fluid buffer" $\mathbf{v}_{1+2i}$ at period $t - \hat{\tau}_{i+1} + 1$.
- $u_{1+2i,t},\ i = 1, \ldots, M-1$, represent the amount of "fluid" that is added to the corresponding "fluid buffer" $\mathbf{v}_{1+2i}$ at period t. This "fluid" corresponds to material transferred to this "fluid buffer" from "fluid buffer" $\mathbf{v}_{2i-1}$ during this period.
- $u_{2M+1,t}$ represents the amount of "fluid" that is transferred from the "fluid buffer" $\mathbf{v}_{2M}$ to the "output fluid buffer" d during period t.

Constraints:

1) The first set of constraints expresses the limited processing capacity at each workstation; namely, the server at each workstation cannot process more than a unit amount of work during a single time unit.

$$\sum_{j \in S_l} \sum_{q=t}^{\min\{t+\hat{\tau}_j-1, T\}} u_{2j,q} \ \leq\ 1,\ l = 1, ..., L,\ t = 1, ..., T$$

2) The second set of constraints expresses the material flow conservation; these constraints break down into the following two parts:

   a) Material flow conservation constraints for period $t = 1$:
   $x_{1+2i,1} = \mathbf{v}_{1+2i}^{init} + u_{1+2i,1} - u_{2+2i,\hat{\tau}_{i+1}} \mathbf{1}_{\{\hat{\tau}_{i+1} \leq T\}},$
   $i = 0 \ldots, M-1$
   $x_{2i,1} = \mathbf{v}_{2i}^{init} + u_{2i,1} - u_{1+2i,1},\ i = 1 \ldots, M$

   b) Material flow conservation constraints for periods $t = 2, \ldots, T$:
   $x_{1+2i,t} = x_{1+2i,t-1} + u_{1+2i,t} - u_{2+2i,t+\hat{\tau}_{i+1}-1} \mathbf{1}_{\{t+\hat{\tau}_{i+1}-1 \leq T\}},\ i = 0 \ldots, M-1$
   $x_{2i,t} = x_{2i,t-1} + u_{2i,t} - u_{1+2i,t},\ i = 1 \ldots, M$

3) This set of constraints expresses the fact that a server cannot work on an empty buffer, while also acknowledging the availability of the "infinite backlog" that provides the input material for processing stage J_1; similar to the second set of constraints, we express these constraints separately for period 1 and for the remaining periods:

   a) For period $t = 1$:
   $\mathbf{v}_{1+2i}^{init} + \mathbf{v}_{2i}^{init} - u_{2+2i,\hat{\tau}_{i+1}} \mathbf{1}_{\{\hat{\tau}_{i+1} \leq T\}} \geq\ 0,$
   $i = 1 \ldots, M-1$

   b) For periods $t = 2, \ldots, T$:
   $x_{1+2i,t-1} - u_{2+2i,t+\hat{\tau}_{i+1}-1} \mathbf{1}_{\{t+\hat{\tau}_{i+1}-1 \leq T\}} \geq\ 0,$
   $i = 1 \ldots, M-1$

4) These constraints express the finite buffering capacity of the line workstations.
   $\sum_{j \in S_l} x_{2j-1,t} + x_{2j,t} + \sum_{q=t+1}^{\min\{t+\hat{\tau}_j-1, T\}} u_{2j,q} \ \leq\ B_l,$
   $l = 1, ..., L, \quad t = 1, ..., T$

5) These constraints account for the imposed deadlock avoidance policy Δ. For the purposes of the presented LP relaxation, we assume that the applied DAP Δ can be expressed as a set of K inequalities having the form

$$A \cdot \hat{\mathbf{s}} \ \leq\ \mathbf{b} \tag{1}$$

   where $\hat{\mathbf{s}}$ is the condensed state of the considered CRL, A is a $K \times M$ matrix, and $\mathbf{b}$ is a K-dim positive vector. The constraints of Eq. 1 can be introduced in the considered LP relaxation by substituting each component $\hat{\mathbf{s}}_j,\ j = 1, \ldots, M$, of the state vector $\hat{\mathbf{s}}$ by the quantity

$$x_{2j-1,t} + x_{2j,t} + \sum_{q=t+1}^{\min\{t+\hat{\tau}_j-1, T\}} u_{2j,q}$$

6) We also want to prevent activity that will not contribute to the total output volume by the end of the time horizon T. For this, we enforce the condition that the total outflow from the network equals the total inflow to it plus the initial "fluid buffer" contents as defined by the vector $\mathbf{v}^{init}$.

$$\sum_{j=1}^{2M} \mathbf{v}_j^{init} + \sum_{t=1}^{T} u_{1,t} - \sum_{t=1}^{T} u_{2M+1,t} = 0$$

7) "Fluid buffer" contents cannot be negative.

$$x_{j,t} \geq 0, \quad j = 1, ..., 2M, \quad t = 1, ..., T$$

8) Also, the "material flows" $u_{j,t}$ cannot be negative either.

$$u_{j,t} \geq 0, \quad j = 1, ..., 2M+1, \quad t = 1, ..., T$$

9) Finally, the next constraint accounts for the non-preemptive nature of our scheduling policies; the non-preemptive character of our policies is implied by the structure of the state space $\mathcal{S}$ and the dynamics that are induced by this structure for the corresponding MDP formulation.

$$u_{2j,\hat{\tau}_j} = 1, \quad j \in \{1, \ldots, M : \mathbf{s}_{1+3(j-1)}^{init} = 1\}$$

Objective Function:

As already stated, we want to maximize the total outflow of the considered CRL over the employed time horizon T, assuming that the line is operated under the relaxed modeling assumptions that are expressed by the constraints of the considered LP, and its initial "fluid buffer" contents are set to the levels that are defined by the state $\mathbf{s}^{init}$ of the original CRL model. Hence, the objective function takes the form:

$$\max \sum_{t=1}^{T} u_{2M+1,t}$$

The induced scheduling policy: After we have solved the LP relaxation, the next step is to interpret the solution of the linear program to a scheduling policy for the underlying CRL. In particular, we want to use the solution of this LP as a "guide" in the selection of the next tangible state $\mathbf{s}$ among the set of tangible states that is defined by the tangible reach, $\mathcal{TR}(\mathbf{s}^{init})$, of the state $\mathbf{s}^{init}$ that constitutes the current decision point.

To effect this selection, let us denote by $\mathbf{u}_1^*$ the vector that is defined by the obtained optimal values for the variables $u_{1,1}, u_{2,\tau_1}, u_{3,1}, u_{4,\tau_2} \ldots, u_{2M,1}$, and by $\mathbf{v}$ the "fluid buffer"

vector that corresponds to any state $\mathbf{s} \in \mathcal{TR}(\mathbf{s}^{init})$. Then, the proposed scheduling policy will select the next tangible state, $\tilde{\mathbf{s}}$, through the following rule:

$$\tilde{\mathbf{s}} \in \arg \min_{\mathbf{s} \in \mathcal{TR}(\mathbf{s}^{init})} \sum_{j=0}^{M-1} |\mathbf{s}_{1+3j} - \mathbf{u}^*_{1,2+2j}| \qquad (2)$$

In more natural terms, the criterion of Eq. 2 seeks to select a tangible state $\mathbf{s} \in \mathcal{TR}(\mathbf{s}^{init})$ that has a server allocation w.r.t. the various processing stages $J_j,\ j = 1, \ldots, M$, that is most similar to the server allocation that is implied by the vector $\mathbf{u}^*_1$.

Furthermore, a secondary criterion that we have used to break any ties that are generated through the criterion of Eq. 2, is as follows:

$$\tilde{\mathbf{s}} \in \arg \min_{\mathbf{s} \in \mathcal{TR}(\mathbf{s}^{init})} |\mathbf{v} - \mathbf{v}^{init} - \mathbf{u}^*_1|_1 \qquad (3)$$

This new criterion perturbs the initial "buffer fluid" vector $\mathbf{v}^{init}$ by the "flow" vector $\mathbf{u}^*_1$, and eventually selects a tangible state $\mathbf{s} \in \mathcal{TR}(\mathbf{s}^{init})$ with a "fluid buffer" vector $\mathbf{v}$ that has the smallest l_1-distance from the aforementioned perturbation $\mathbf{v}^{init} + \mathbf{u}^*_1$; hence, this secondary criterion considers also state similarity in terms of buffer occupancy.

Selecting an appropriate time-horizon length T: One parameter that needs to be further specified for the complete definition of the "fluid" relaxation that was presented in the previous parts of this section, is the value of the parameter T, i.e., the time-horizon over which the line output will be maximized. This selection is driven by the realization that the optimal solution of the considered LP will essentially lead the system to an operational regime that provides the maximal possible output of the system as defined by the bottleneck stations of the line, and it will divert from this operational regime only towards the end of the operational horizon, in an effort to satisfy the termination condition of Constraint #6 above. Furthermore, some numerical experimentation reported in [12] has shown that as long as the selected T value is adequately large to let the line reach the aforementioned operational regime, the returned vector $\mathbf{u}^*_1$ will be quite insensitive to the exact T value. So, with these insights and findings, we propose to set $T = (\sum_{i=1}^{L} B_i)(\sum_{j=1}^{M} \hat{\tau}_j)$, since this is an upper bound of the time that is necessary to empty the line from the entire workload that is defined by the state $\mathbf{s}^{init}$ under any globally nonidling policy.

IV. Example

The CRL considered in this example is depicted in Figure 1. It consists of two workstations, WS_1 and WS_2, and an I/O port that interfaces the CRL with its operational environment. Each workstation has a single server, depicted as a grey ellipse in Figure 1, and two buffer slots, depicted by the corresponding rectangles. Parts visiting each of the two workstations are accommodated at one of the available buffer slots, and they are processed by the workstation server by having the server visiting the corresponding slot. A robotic manipulator supports the necessary material handling functions, and integrates the entire facility to a fully automated cell. Figure 1 also provides the process route for the parts that are processed through this CRL; since workstation WS_1 is visited twice by each part, the considered layout constitutes a re-entrant line. Furthermore, letting $J_j,\ j = 1, 2, 3$, denote the three processing stages of this CRL, under the notation that was introduced in the earlier parts of this section we also have that $W(J_1) = W(J_3) = WS_1$ and $W(J_2) = WS_2$. Finally, we assume that the processing times for each of the three processing stages are exponentially distributed with corresponding instantaneous rates μ_j and corresponding expected values $\tau_j = 1/\mu_j,\ j = 1, 2, 3$.

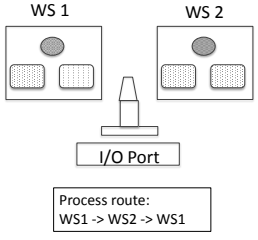

Fig. 1: An example CRL.

The state $\mathbf{s}$ of this CRL is a 7-dim integer vector. The first two components of vector $\mathbf{s}$, $\mathbf{s}_1$ and $\mathbf{s}_2$, report, respectively, the numbers of parts at workstation WS_1 executing processing stage J_1 and having completed the processing of this stage; components $\mathbf{s}_3,\ \mathbf{s}_4$ and $\mathbf{s}_5$ report the number of parts in workstation WS_2 respectively waiting for the execution of stage J_2, executing this stage, and having completed execution of this stage; finally, components $\mathbf{s}_6$ and $\mathbf{s}_7$ of state $\mathbf{s}$ report the number of parts at workstation WS_1 respectively waiting for the execution of stage J_3 and executing this stage.

On the other hand, the state $\hat{\mathbf{s}}$ of the FSA $\hat{\Phi}$ that characterizes the untimed dynamics of the considered CRLs w.r.t. deadlock and the defining logic of the necessary DAPs, is a 3-dim vector with: $\hat{\mathbf{s}}_1 \equiv \mathbf{s}_1 + \mathbf{s}_2$; $\hat{\mathbf{s}}_2 \equiv \mathbf{s}_3 + \mathbf{s}_4 + \mathbf{s}_5$; and $\hat{\mathbf{s}}_3 \equiv \mathbf{s}_6 + \mathbf{s}_7$. Furthermore, it is easy to check that, in the condensed state space $\hat{S}$ of the considered CRL, the only reachable unsafe state is the state $\hat{\mathbf{s}}^d = (2, 2, 0)$, i.e., the state where workstation WS_1 is fully loaded with parts executing or having completed processing stage J_1, and workstation WS_2 is also fully loaded. Obviously, state $\hat{\mathbf{s}}^d$ is also a deadlock state, since none of the loaded parts can advance to its next requested workstation. This deadlock can be avoided, in a maximally permissive manner, by enforcing the constraint

$$\hat{\mathbf{s}}_1 + \hat{\mathbf{s}}_2 \leq 3 \qquad (4)$$

which constitutes a representation of the maximally permissive DAP Δ^* for the considered CRL.

Figure 2 depicts the reachable and safe state space of the considered CRL, S_{rs}, under the maximally permissive DAP of Eq. 4. A complete characterization of the depicted states is provided in Table I. It can be checked through Table I that the double-circled states in Figure 2 enable only events in the set $\{e^d_j : j = 1, 2, 3, \}$, and therefore, they constitute the tangible states of the depicted STD. Figure 2 also provides the transition probabilities out of these tangible states, as defined by the exponential races that take place in each of these states. On the other hand, the single-circled states in Figure 2 enable some resource allocation event of type e^l_j, e^a_j or $e^p_j,\ j = 1, 2, 3$, and they are the vanishing states of the corresponding STD. Transitions out of these states must be arbitrated by the adopted scheduling policy. However, the specification of this scheduling policy can be simplified by the fact that the timed performance of the considered CRL is determined by the sojourn times spent by this line at the tangible states only. This realization enables a "thinning" of the depicted STD by removing the vanishing states and their interconnected transitions depicted by dashed lines, on the basis that the remaining STD structure does not alter

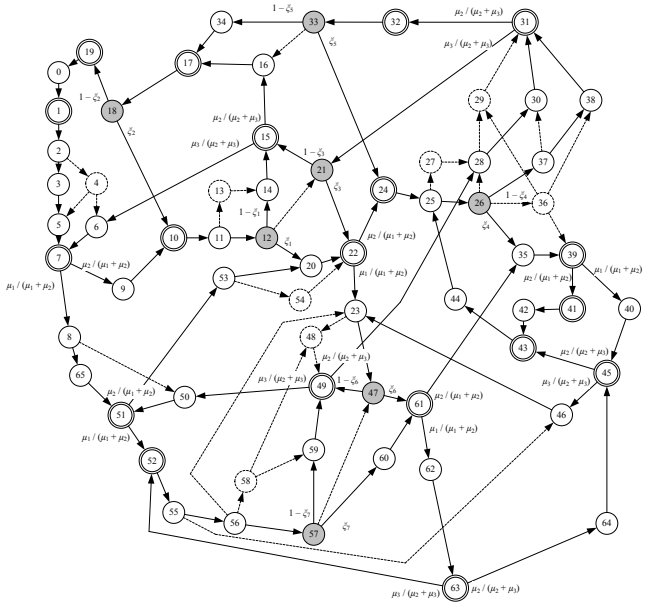

Fig. 2: The reachable and safe state space S_{rs} for the CRL of Figure 1, and some further structure that defines the MDP characterizing the corresponding throughput-maximization problem.

the reachability between the various tangible states and the vanishing states that result from the execution of an e_j^d type of event.[10] The vanishing states that enable choice in the remaining STD structure are the seven states colored in grey in Figure 2. More specifically, in the remaining STD, each of these seven states possesses two enabled transitions, and the effected choice between these two transitions can be represented by setting the corresponding variable $\xi_k,\ k = 1, \ldots, 7$, either to the value of 0 or 1. These settings of the variables ξ_k define completely the next tangible marking $\mathbf{s}''$ to be reached from state $\mathbf{s}'$, and at the same time, they do not introduce any "coupling" in the corresponding transitioning logic that is effected at any pair of markings $\mathbf{s}'_1$ and $\mathbf{s}'_2$ that constitute decision points for the underlying MDP. Hence, each possible binary pricing of the variables ξ_k depicted in Figure 2 defines a complete deterministic stationary policy for the considered MDP. In order to fully specify this MPD, we must also specify the immediate-reward function. Under the previously introduced notation, this function is defined by associating with every pair $(\mathbf{s}', \mathbf{s}'')$ an immediate reward equal to the occurrence probability of event $e_3^d (\equiv e^u)$ in state $\mathbf{s}''$.

The application of the scheduling methodology of Section III to the considered CRL requires the solution of the following LP formulation at each of the seven vanishing states $\mathbf{s}_l,\ l \in \{12, 18, 21, 26, 33, 47, 57\}$:

$$\max_{x,u} \quad \sum_{t=1}^{T} u_{7,t}$$

[10] A complete methodology that formalizes and supports computationally this "thinning" process, is presented in [22].

TABLE I: The state description for the STD of Figure 2.

$\mathbf{s}$	$\mathbf{s}_1\mathbf{s}_2$	$\mathbf{s}_3\mathbf{s}_4\mathbf{s}_5$	$\mathbf{s}_6\mathbf{s}_7$	$\mathbf{s}$	$\mathbf{s}_1\mathbf{s}_2$	$\mathbf{s}_3\mathbf{s}_4\mathbf{s}_5$	$\mathbf{s}_6\mathbf{s}_7$
0	0 0	0 0 0	0 0	33	0 0	0 0 1	1 0
1	1 0	0 0 0	0 0	34	0 0	0 0 0	2 0
2	0 1	0 0 0	0 0	35	1 0	1 0 1	1 0
3	1 1	0 0 0	0 0	36	0 0	0 1 1	1 0
4	0 0	1 0 0	0 0	37	0 0	1 0 0	2 0
5	1 0	1 0 0	0 0	38	0 0	0 1 0	2 0
6	0 0	0 1 0	0 0	39	1 0	0 1 1	1 0
7	1 0	0 1 0	0 0	40	0 1	0 1 1	1 0
8	0 1	0 1 0	0 0	41	1 0	0 0 2	1 0
9	1 0	0 0 1	0 0	42	0 1	0 0 2	1 0
10	1 0	0 0 0	1 0	43	0 1	0 0 2	0 1
11	0 1	0 0 0	1 0	44	0 1	0 0 2	0 0
12	0 0	1 0 0	1 0	45	0 1	0 1 1	0 1
13	0 1	0 0 0	0 1	46	0 1	0 1 1	0 0
14	0 0	1 0 0	0 1	47	0 0	1 1 0	1 0
15	0 0	0 1 0	0 1	48	0 1	0 1 0	0 1
16	0 0	0 0 1	0 1	49	0 0	1 1 0	0 1
17	0 0	0 0 0	1 1	50	0 0	1 1 0	0 0
18	0 0	0 0 0	1 0	51	1 0	1 1 0	0 0
19	0 0	0 0 0	0 1	52	0 1	1 1 0	0 0
20	1 0	1 0 0	1 0	53	1 0	1 0 1	0 0
21	0 0	0 1 0	1 0	54	1 0	0 1 1	0 0
22	1 0	0 1 0	1 0	55	0 1	1 0 1	0 0
23	0 1	0 1 0	1 0	56	0 1	1 0 0	1 0
24	1 0	0 0 1	1 0	57	0 0	2 0 0	1 0
25	0 1	0 0 1	1 0	58	0 1	1 0 0	0 1
26	0 0	1 0 1	1 0	59	0 0	2 0 0	0 1
27	0 1	0 0 1	0 1	60	1 0	2 0 0	1 0
28	0 0	1 0 1	0 1	61	1 0	1 1 0	1 0
29	0 0	0 1 1	0 1	62	0 1	1 1 0	1 0
30	0 0	1 0 0	1 1	63	0 1	1 1 0	0 1
31	0 0	0 1 0	1 1	64	0 1	1 0 1	0 1
32	0 0	0 0 1	1 1	65	1 1	0 1 0	0 0

s.t.

$$u_{2,t} + u_{6,t} \leq 1,\ t = 1, \ldots, T$$

$$u_{4,t} \leq 1,\ t = 1, \ldots, T$$

$$x_{1+2i,1} - u_{1+2i,1} + u_{2+2i,\hat{\tau}_{i+1}} = \mathbf{v}_{1+2i}^{init},\ i = 0, \ldots, 2$$

$$x_{2i,1} - u_{2i,1} + u_{1+2i,1} = \mathbf{v}_{2i}^{init},\ i = 1, \ldots, 3$$

$$x_{1+2i,t} - x_{1+2i,t-1} - u_{1+2i,t} + u_{2+2i,t+\hat{\tau}_{i+1}-1} = 0,$$
$$i = 0, \ldots, 2;\ t = 2, \ldots, T$$

$$x_{2i,t} - x_{2i,t-1} - u_{2i,t} + u_{1+2i,t} = 0,\ i = 1, \ldots, 3;\ t = 2, \ldots, T$$

$$u_{2+2i,1} \leq \mathbf{v}_{1+2i}^{init} + \mathbf{v}_{2+2i}^{init},\ i = 1, 2$$

$$u_{2+2i,t} - x_{1+2i,t-1} \leq 0,\ i = 1, 2;\ t = 2, \ldots, T$$

$$x_{1,t} + x_{2,t} + x_{5,t} + x_{6,t} \leq 2,\ t = 1, \ldots, T$$

$$x_{3,t} + x_{4,t} \leq 2,\ t = 1, \ldots, T$$

$$x_{1,t} + x_{2,t} + x_{3,t} + x_{4,t} \leq 3,\ t = 1, \ldots, T$$

$$\sum_{t=1}^{T} u_{7,t} - \sum_{t=1}^{T} u_{1,t} = \sum_{j=1}^{6} \mathbf{v}_j^{init}$$

$$x_{i,t} \geq 0,\ i = 1, \ldots, 6;\ t = 1, \ldots, T$$

$$u_{i,t} \geq 0,\ i = 1, \ldots, 7;\quad t = 1, \ldots, T$$

$$u_{2j,\hat{\tau}_j} = 1,\ j \in \{1, \ldots, 3 : \mathbf{s}_{1+3(j-1)}^{init} = 1\}$$

TABLE II: Comparing the policy specified for the example CRL of Figure 1 by the methodology that is presented in this work, to the optimal policy for this line.

$\mathbf{s}$	Vanishing State			Tangible Reach			Opti-	Sel. Crit.
	$\mathbf{s}_1\mathbf{s}_2$	$\mathbf{s}_3\mathbf{s}_4\mathbf{s}_5$	$\mathbf{s}_6\mathbf{s}_7$	$\mathbf{s}_1\mathbf{s}_2$	$\mathbf{s}_3\mathbf{s}_4\mathbf{s}_5$	$\mathbf{s}_6\mathbf{s}_7$	mal	of Eq. 2
12	00	100	10	10	010	10	YES	1.0626
				00	010	01	NO	1.4439
18	00	000	10	10	000	10	YES	0.7693
				00	000	01	NO	2.3220
21	00	010	10	10	010	10	YES	0.7285
				00	010	01	NO	1.2715
26	00	101	10	10	011	10	NO	1.7492
				00	010	11	YES	1.2365
33	00	001	10	10	001	10	YES	0.8282
				00	000	11	NO	1.7219
47	00	110	10	00	110	01	YES	0.6529
				10	110	10	NO	1.3471
57	00	200	10	00	110	01	YES	0.6826
				10	110	10	NO	1.5370

The parameters $\mathbf{s}_{\cdot}^{init}$ and $\mathbf{v}_{\cdot}^{init}$ that appear in the right-hand-side of the above formulation, are determined by the considered vanishing state $\mathbf{s}_l$ according to the defining logic of these parameters that was discussed during their introduction in Section III.

Table II presents the obtained scheduling policy for this example CRL when $\tau_j = 1.0,\ \forall j$. Each primary row in Table II corresponds to one of the considered vanishing states $\mathbf{s}_l$, and the first two parts of the row provide a complete characterization of state $\mathbf{s}_l$ and its tangible reach, $\mathcal{TR}(\mathbf{s}_l)$. On the other hand, the row entry in the column entitled "Optimal" provides the choice for the next tangible state $\mathbf{s}'_l \in \mathcal{TR}(\mathbf{s}_l)$ specified by the optimal policy that is obtained through the solution of the throughput-maximizing MDP formulation for this CRL. Finally, the last column of Table II provides the values for the "action"-selection criterion of Equation 2 that are obtained from the solution of the corresponding LP relaxations. It can be checked that, for each state $\mathbf{s}_l,\ l \in \{12, 18, 21, 26, 33, 47, 57\}$, the minimum value for this criterion corresponds to the tangible state $\mathbf{s}'_l \in \mathcal{TR}(\mathbf{s}_l)$ that is the optimal choice according to column "Optimal". Hence, for this example CRL, our scheduling methodology returns an optimal policy.[11]

V. Conclusions

This paper has extended the scheduling methodology for complex stochastic networks that is based on the solution of a pertinent "fluid" relaxation of the addressed scheduling problem, to the class of stochastic networks that model resource allocation with blocking and deadlocking effects. For better specificity, the results were detailed in the operational context of a re-entrant line with finite buffering capacity at each workstation, and the particular scheduling problem addressed was the maximization of the long-term throughput. The presented methodology was substantially enabled by a pre-established ability to control the underlying resource allocation for deadlock freedom, and by the expression of the corresponding DAP as a set of linear inequalities on the system state. At the same time, the developed LP relaxation differs considerably from similar past developments, since it must account for the applied DAP and the blocking effects that take place in the considered CRLs.

[11] Additional investigations and results regarding the application of the presented scheduling methodology to the example CRL of Fig. 1 can be found in [12]. That paper also reports additional results regarding (i) the performance of the presented method when applied to larger CRL configurations, and (ii) the computational times required for the solution of the employed LP formulations.

Our future work will seek to provide a more structured characterization of the dynamics that are encoded in the proposed "fluid" relaxation by means of some formal, DES-theoretic modeling frameworks, and it will employ this characterization in order to further analyze the structure of the optimal solutions of the corresponding LP. We shall also try to bring into this analysis the notion of "robustness" that has been pursued w.r.t. some other "fluid" relaxations in [8]. Finally, from a more application-oriented standpoint, we shall address the necessary details for extending the scheduling methodology presented in this work to the more general RAS classes that are studied in [4].

Acknowledgement

This work was partially supported by the NSF grant ECCS-1405156.

References

[1] P. R. Kumar, "Scheduling semiconductor manufacturing plants," *IEEE Control Systems Magazine*, vol. 14–6, pp. 33–40, 1994.
[2] S. Meyn, *Control Techniques for Complex Networks*. Cambridge, UK: Cambridge University Press, 2008.
[3] J. G. Dai and W. Lin, "Maximum Pressure Policies in Stochastic Processing Networks," *Operations Research*, vol. 53, pp. 197–218, 2005.
[4] S. Reveliotis, "Logical Control of Complex Resource Allocation Systems," *NOW Series on Foundations and Trends in Systems and Control*, vol. 4, pp. 1–224, 2017.
[5] J. G. Dai, "Stablity of fluid and stochastic processing networks," Dept. of Mathematical Sciences, Unversity of Aarhus, Denmark, Tech. Rep. Miscellanea, No. 9, 1998.
[6] C. Maglaras, "Discrete-review policies for scheduling stochastic networks: trajectory tracking and fluid-scale asymptotic optimality," *Ann. Appl. Prob.*, vol. 10, pp. 897–929, 2000.
[7] D. Bertsimas and J. Sethuraman, "From fluid relaxations to practical algorithms for job shop scheduling: The makespan objective," *Mathematical Programming*, vol. 92, pp. 61–102, 2002.
[8] D. Bertsimas, E. Nasrabadi, and I. C. Paschalidis, "Robust fluid processing networks," *IEEE Trans. on Automatic Control*, vol. 60, pp. 715–728, 2015.
[9] C. G. Cassandras, Y. Wardi, B. Melamed, G. Sun, and C. G. Panayiotou, "Pertubation analysis for online control and optimization of stochastic fluid models," *IEEE Trans. on Automatic Control*, vol. 47, pp. 1234–1248, 2002.
[10] T. Bountourelis and S. A. Reveliotis, "Optimal node visitation in stochastic digraphs," *IEEE Trans. on Automatic Control*, vol. 53, pp. 2558–2570, 2008.
[11] P. R. Kumar, "Re-entrant lines," *Queueing Systems*, vol. 13, pp. 87–110, 1993.
[12] M. Ibrahim and S. Reveliotis, "Throughput maximization of capacitated re-entrant lines through fluid relaxation," ISyE, Georgia Tech, Tech. Rep. (submitted for publication), 2017.
[13] H. Chen and D. D. Yao, *Fundamentals of Queueing Networks*. NY, NY: Springer, 2001.
[14] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems (2nd ed.)*. NY, NY: Springer, 2008.
[15] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modeling with Generalized Stochastic Petri Nets*. John Wiley & Sons, 1994.
[16] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.
[17] P. Glasserman and D. Yao, *Monotone Structure in Discrete-Event Systems*. NY,NY: John Wiley & Sons, Inc., 1994.
[18] S. A. Reveliotis, "The destabilizing effect of blocking due to finite buffering capacity in multi-class queueing networks," *IEEE Trans. on Autom. Control*, vol. 45, pp. 585–588, 2000.
[19] R. Li, "Performance optimization of complex resource allocation systems," Ph.D. dissertation, ISyE, Georgia Tech, Atlanta, GA, 2016.
[20] G. Weiss, "On optimal draining of re-entrant fluid lines," *IMA volumes in mathematics and its applications*, vol. 71, pp. 91–91, 1995.
[21] ——, "Scheduling and control of manufacturing systems-a fluid approach," vol. 37, pp. 577–586, 1999.
[22] R. Li and S. Reveliotis, "Designing parsimonious scheduling policies for complex resource allocation systems through concurrency theory," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 26, pp. 511–537, 2016.