Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

DISCRETE EVENT OPTIMIZATION: SINGLE–RUN INTEGRATED SIMULATION–OPTIMIZATION USING MATHEMATICAL PROGRAMMING

Giulia Pedrielli

Centre for Maritime Studies
National University of Singapore
15 Prince George's Park
Singapore, SG 118414, SINGAPORE

Andrea Matta

Department of Industrial Engineering and Management
Shanghai Jiao Tong University
800 Dong Chuan Road
Shanghai, 200240, CHINA

Arianna Alfieri

Department of Management and Production Engineering
Politecnico di Torino
Corso Duca degli Abruzzi 24
10129 Torino, ITALY

ABSTRACT

Optimization of discrete event systems conventionally uses simulation as a black–box oracle to estimate performance at design points generated by a separate optimization algorithm. This decoupled approach fails to exploit an important advantage: simulation codes are white-boxes, at least to their creators. In fact, the full integration of the simulation model and the optimization algorithm is possible in many situations. In this contribution, a framework previously proposed by the authors, based on the mathematical programming methodology, is presented under a wider perspective. We show how to derive mathematical models for solving optimization problems while simultaneously considering the dynamics of the system to be optimized. Concerning the solution methodology, we refer back to retrospective optimization (RO) and sample path optimization (SPO) settings. Advantages and drawbacks deriving from the use of mathematical programming as *work models* within the RO (SPO) framework will be analyzed and its convergence properties will be discussed.

1 INTRODUCTION

Mathematical programming representations can be used to describe the dynamics of discrete event systems (DESs) using a set of equations and an objective function (Schruben 2000, Chan and Schruben 2008a), which drives the model to execute all the events as soon as possible and the optimal solution of such mathematical programming models represents the simulated system dynamics.

In case the DES becomes too complex, the mathematical programming approach might become impractical as no equation can describe certain behaviors or, in case they exist, the resulting mathematical model is computationally intractable (e.g., complex flows, presence of real time dispatching rules). Nevertheless, for many systems of interest, linear programming models are sufficient to describe the system dynamics (Chan and Schruben 2008a, Chan and Schruben 2008b, Alfieri and Matta 2012b).

The possibility to have an analytical description of system dynamics makes mathematical programming representation an appealing technique for analyzing in a formal way the simulated systems. Simulation output is generated by means of equations rather than a set of logical rules embedded in a computer code.

This enriches the types of analysis that can be performed on the simulation output (Pedrielli 2013, Matta et al. 2014, Chan and Schruben 2008a).

In addition to the aforementioned aspects, the use of mathematical programming for simulation allows to integrate simulation and optimization since the system dynamics constraints can be integrated in a mathematical programming model developed for optimization purposes. However, such integrated models are no longer LPs (as they usually are for simulation) and Mixed Integer Linear Programming (MILP) have to be considered with the related computational burden. Approximate LP models to simulate and optimize DESs can be developed to overcome this difficulty. Alfieri and Matta (2012a) propose an approximation consisting in modeling queues as *time buffers* (TB), i.e., temporal lags between two events, instead of the traditional *space buffers*. This approximation preserves the model linearity even when used for optimization and it allows to avoid the presence of two decoupled modules iteratively interacting with each other.

The output of the simulation module is the input for the optimization in an iterative procedure which continues until the optimal solution is found or a predefined stopping condition is satisfied (Spall 2003). For the optimization module, a vast choice of methods is available (Fu et al. 2005); the most common ones are response surface methodology, stochastic approximation, random search, sample path optimization (SPO) and its iterative version, retrospective optimization (RO).

In case mathematical programming models are used for simulation–optimization, then SPO and RO are commonly adopted (Healy and Schruben 1991, Plambeck et al. 1996, Robinson 1996). Examples of implementation of the integrated simulation–optimization framework with SPO as solution methodology can be found in (Alfieri and Matta 2012a) that solve the buffer allocation problem in open tandem queuing systems; Alfieri et al. (2013) approximately select the maximum number of jobs in a closed tandem queuing systems; in (Weiss and Stolletz 2015) the discrete buffer allocation problem is solved by means of a tailored Benders' decomposition technique. In a slightly different framework from the one proposed by the authors, Kolb and Gttlich (2015) solve the continuous buffer space allocation problem modeling a hybrid system, discrete in the machine states and continuous in the material flow. The same concept is exploited in (Tan 2015) for the optimal production flow rate control problem for a continuous material flow system with an unreliable station and deterministic demand.

In general, the motivation for proposing mathematical programming to integrate simulation and optimization stands in that the integration of the system dynamics with the optimization represents a fundamental step towards the realization of efficient algorithms which can rely on the information coming from a white–box model (refer to section 3). Also the ability to model a simulation–optimization problem as a mathematical programming model, within the Sample Path Optimization framework, enables the convergence study. In addition, the fact that simulation and optimization both work on the same sample path might lead to reduction in the required effort in terms of number of observations required to solve the problem (e.g., the number of jobs to simulate, the number of patients visiting hospital facilites). However, to make this statement applicable to the computational effort, more efficient algorithms need to be developed.

The recent contributions have several aspects in common. Similarly to (Plambeck et al. 1996), a single, fairly long, *run* is considered and mathematical programming techniques are used to solve sample–path integrated simulation–optimization problems. Several challenges are still to be tackled as well as opportunities of improvement can be reached. Convergence of the methodology is addressed only in the pioneer work of Robinson (1996), but no constraints are considered in that case and mathematical programming is not explicitly considered as simulation technique. In fact, Robinson (1996) does not refer to integrated simulation–optimization models and an external oracle producing consistent output is assumed to be available. Nevertheless, this new framework inherits issues from Robinson (1996) as the demonstration of the advantage of static sampling versus iterative sampling (Healy and Schruben 1991).

In light of the aforementioned aspects, this paper presents an overview of mathematical programming for integrated simulation–optimization by presenting the approach, the main results from the existing literature and describing the main challenges/opportunities. Furthermore, a convergence proof scheme is provided for the integrated framework proposed by the authors.

2 DEO: INTEGRATED SIMULATION–OPTIMIZATION

Discrete Event Optimization (DEO) refers to a framework for the integrated simulation–optimization of DES. At this stage of development of the framework, we consider systems such as supply chains or manufacturing systems. In this context, the topology of the DES we consider can be represented by a queueing network with the set of servers $\mathbb{J} = \{0,\dots,J+1\}$ and the set of possible transaction routes for job i ($i \in \mathbb{N} = \{0,\dots,n\}$) between servers, represented by $\mathbb{Q}_i = \{(j,j')|j,j' \in \mathbb{J}\}$, $\forall i$. For each pair (j,j'), the arc connecting j and j' belongs to $\mathbb{Q}_i$ if and only if job i can *directly* flow from node j to node j'. The *source* node, represented by index $j = 0$, is the server j having no predecessor. The *sink* node is, instead, the server j having no successor and it is indexed by $J+1$. The source node represents an infinite external arrival stream of customers, whereas the sink node is the output gate through which jobs are released from the network. We consider a general setting in which no explicit condition has to be imposed over the system layout (i.e., output can be linear, as well as merge and split points or cycles can be modeled).

Let E_{ij}^{ξ} and e_{ij}^{ξ} denote the events occurring in the system and the related occurrence times, respectively, where $\xi \in \mathbb{T}$ is the event type (e.g., $\mathbb{T} = \{a,s,f\}$ as arrival, start of process or departure of a job from a server, respectively), and the pair (i,j) indicates the job i and the server j the event refers to. We assume that job i at server j undergoes a process activity with duration bounded by a start event E_{ij}^{s} occurring at time e_{ij}^{s} and a completion event E_{ij}^{f} occurring at time e_{ij}^{f}; the duration of the process is t_{ij} and, in case of stochastic DES, $\{t_{ij}\}$ may follow some known statistical distributions.

More generally, the flow of each job i is determined by the occurrence of a set of events $\mathbb{W}^i = \left\{E_{ij}^{\xi}, \xi \in \mathbb{T},\ (j,j') \text{ or } (j',j) \in \mathbb{Q}_i\right\}$, $i \in \mathbb{N}$. Each event E_{ij}^{ξ} in the set $\mathbb{W}^i$ has a set $\mathbb{W}_{ij}^{I\xi}$ of the input events, i.e., *triggering events*, and a set $\mathbb{W}_{ij}^{O\xi}$ of the output events, i.e., *triggered events*. Notice that elements in the sets $\mathbb{W}_{ij}^{I\xi}$ and $\mathbb{W}_{ij}^{O\xi}$ might not be in the set $\mathbb{W}^i$. An example of this is when the triggering (triggered) event is related to a job $i^{'} \neq i, i^{'} \in \mathbb{N}$. According to Matta et al. (2014), we provide a set of definitions which will be useful in the subsequent explanations.

Definition 1 (Event Relationship Graph Lite, ERGL) An ERGL is an oriented weighted graph where the set of nodes $\mathbb{W} = \{E_{ij}^{\xi}, i \in \mathbb{N}, j \in \mathbb{J}, \xi \in \mathbb{T}\}$ represents the set of events E_{ij}^{ξ} occurring in the system. Each node is assigned a value equal to the time e_{ij}^{ξ} when the associated event occurs. Directed arcs connect different event pairs $\left(E_{ij}^{\xi}, E_{i'j'}^{\xi'}\right)$ and the set of arcs $\mathbb{E} = \left\{\left(E_{ij}^{\xi}, E_{i'j'}^{\xi'}\right), i,i' \in \mathbb{N}, j,j' \in \mathbb{J}, \xi,\xi' \in \mathbb{T}\right\}$ represents the precedence relationships between events. Each arc can be assigned a weight $w_{E_{i'j'}^{\xi'}}^{E_{ij}^{\xi}}$ that can be continuous (positive or negative) or binary. In order to simplify the notation, we refer to weights as $w_{\xi',i',j'}^{\xi,i,j}$.

Definition 2 (Connected Events) Let e_{ij}^{ξ} and $e_{i'j'}^{\xi'}$ be the times when the events E_{ij}^{ξ} and $E_{i'j'}^{\xi'}$ occur, respectively. These two events are connected if and only if $\left(E_{i'j'}^{\xi'} \in \left(\mathbb{W}_{ij}^{I\xi} \cup \mathbb{W}_{ij}^{O\xi}\right) \bigwedge E_{ij}^{\xi} \in \left(\mathbb{W}_{i'j'}^{I\xi'} \cup \mathbb{W}_{i'j'}^{O\xi'}\right)\right)$. In particular, if $\left(E_{i'j'}^{\xi'} \in \mathbb{W}_{ij}^{I\xi} \bigwedge E_{ij}^{\xi} \in \mathbb{W}_{i'j'}^{O\xi'}\right)$, the connection establishes that event $E_{i'j'}^{\xi'}$ can trigger event E_{ij}^{ξ}. Similarly, if $\left(E_{ij}^{\xi} \in \mathbb{W}_{i'j'}^{I\xi'} \bigwedge E_{i'j'}^{\xi'} \in \mathbb{W}_{ij}^{O\xi}\right)$, event E_{ij}^{ξ} can trigger event $E_{i'j'}^{\xi'}$.

Definition 3 (Controlled ERGL) Given a set $\mathbb{W}^N \subseteq \mathbb{W}$ of natural events (events determined by *physical constraints*, e.g., a job cannot enter the system before its arrival times, and it cannot be processed by two servers at the same time), a controlled ERGL is the ordered set of events $\mathbb{W}^{CN}$ containing all the elements in $\mathbb{W}^N$ *and* the set $\mathbb{W}^C \subseteq \mathbb{W} \setminus \mathbb{W}^N$ of *control* events. Elements in $\mathbb{W}^{CN}$ are connected through natural arcs ($\mathbb{E}^N \subseteq \mathbb{E}$) and directed control arcs ($\mathbb{E}^C \subseteq \mathbb{E}$). The weight associated to each control arc can be either continuous, denoted as $s^{\xi ij}_{\xi' i' j'}$ and referred to as *time buffer*, or binary weight, with $\kappa^{\xi ij}_{\xi' i' j'}$ indicating the associated binary value.

The values associated to the nodes of the resulting graph and the control weights can be translated into decision variables in an optimization problem, while natural weights, are fixed input parameters. Under this new perspective, the optimization of a system corresponds to the search of the best set of control events $\mathbb{W}^C = \left\{E^{\xi}_{ij}\right\}$, and the set of arcs (and related weights) such that the resulting occurrence times $\left\{e^{\xi}_{ij}\right\}$ satisfy some target performance. Using the mathematical programming formalism, we can formulate the optimization problem as (Matta et al. 2014):

$$\min \quad \sum_{\upsilon \in \mathbb{W}} \alpha_{\upsilon} e_{\upsilon} + \sum_{\nu \in \mathbb{E}^C} \left(\beta_{\nu} s_{\nu} + \gamma_{\nu} \kappa_{\nu}\right) + \vartheta \varepsilon \tag{1}$$

$$\sum_{\nu \in \mathbb{W}^C} p\left(e_{\nu}\right) \leq \mu^* + \varepsilon \tag{2}$$

$$e^{f}_{ij} \geq e^{s}_{ij} + t_{ij} \qquad \forall E^{s}_{ij}, E^{f}_{ij} \in \mathbb{W}^N \tag{3}$$

$$e^{\xi'}_{i' j'} \geq e^{\xi}_{ij} - q\left(w^{\xi ij}_{\xi' i' j'}\right) \qquad \forall E^{\xi}_{ij}, E^{\xi'}_{i' j'} \in \mathbb{W}^{CN} \tag{4}$$

Equation (1) is the objective function, having as decision variables the event times e_{υ}, and the control parameters (weights in the ERG graph). Function (1) can consider a single or multiple objectives depending on the values of α_{υ}, β_{ν} and γ_{ν} that are known function coefficients. The term $\vartheta \varepsilon$ serves the purpose to penalize finite sample path solutions that do not meet the desired performance (i.e., violate the constraint (2) if the decision variable ε is not considered). This penalization approach has impact on the implementation of the algorithm but not on the asymptotic properties. Equation (2) is the performance constraints, where μ^* is the target performance and p is any function of the control event times. The natural dynamics linking event times if no control is added are represented by constraints (3), stating that customer i cannot leave stage j (e^{f}_{ij}) before accessing the server (e^{s}_{ij}) and completing the service (t_{ij}). Parameters t_{ij} form the collection of realizations of random variables characterizing the queueing process (e.g., arrival times, processing times). These values translate the weights between the nodes E^{s}_{ij} (starting event) and E^{f}_{ij} (departure event). According to the definitions in the previous section, we assume to know the probabilistic characterization of the input stochastic processes. Hence, we can generate t_{ij} as realizations of known random variables. Constraints (4) refer to control constraints. Variables $e^{\xi}_{i,j}$ and $e^{\xi'}_{i' j'}$ represent the time occurrence of two events relating job i' on stage j' and job i on stage j that are linked by control $q(w^{\xi ij}_{\xi' i' j'})$. If the relationship between the two event times is boolean, function q has the form $(1 - \kappa^{\xi ij}_{\xi' i' j'}) \cdot M$, where $\kappa^{\xi ij}_{\xi' i' j'}$ is a binary decision variable and M is a large number. Instead, in case of continuous relationship, q is a function of the continuous variable $s^{\xi ij}_{\xi' i' j'}$ time buffer.

The main feature of DEO models is that they are based on events rather than on states, which generally grow faster than events. Notice that when $\beta_{\nu} = 0$ and $\gamma_{\nu} = 0$ $\forall \nu$ in equation (1) and the performance constraint is not present, the model becomes a simulation model while when $\alpha_{\upsilon} = 0$ $\forall \upsilon$ is null, we have an optimization model.

Notice that, since $i = 1, \ldots, n$, the model is a function of n. Hence, it is apparent that, as the simulation length increases (i.e., n increaes), also the number of decision variables and constraints increases.

3 SOLUTION APPROACH

The DEO models in (Weiss and Stolletz 2015, Kolb and Gttlich 2015, Tan 2015) are solved in the framework of SPO (Robinson 1996). In SPO, a single sample path is generated and the problem is solved using techniques from deterministic optimization. The length of the sample size impacts on the quality of the obtained results.

SPO can be interpreted as a single–iteration RO, i.e., as a single large deterministic problem solved to optimality (Healy and Schruben 1991). Specifically, RO is an iterative technique requiring the solution of a sequence of sample-path problems of increasing size (Jin and Schmeiser 2003). At each step l, a sample-path problem P_l is solved using the information obtained from the previous iterations. The solution at iteration l has an error tolerance ε_l from the optimum and, combined with the solutions from previous iterations, it is used to compute the new retrospective candidate. The algorithm terminates when a stopping condition is met, otherwise the sample size is increased and the error tolerance decreased. Large error tolerance values at the beginning of the procedure allow for rough but fast solutions that are then used in the subsequent iterations as warm start solutions.

The use of mathematical programming for solving sample–path problems affects the implementation of RO. Nevertheless, the retrospective one could be an interesting framework to investigate to seek more efficient algorithms for the solution of DEO models. Indeed, the complexity of DEO models increases with the size of the sample path due to the events times decision variables (i.e., $\{e_{ij}^{\xi}\}$). As a result, the computational time for solving long sample-path might be too large because of the model size. Under this perspective, a warm start, as in the RO approach, could speed up the solution algorithm: the algorithm might be iteratively stopped before the optimal solution is reached and the suboptimal solution might be used as starting point for the next iteration executed with an increased sample path. Using mathematical programming techniques could help to control the tolerance error in solving the deterministic sample-path problem.

Independently from the iterative aspect, the solution of the deterministic problem strongly impacts on the efficiency, in terms of computation time, of the algorithm. In (Alfieri and Matta 2012a, Alfieri et al. 2013) an LP model is solved by state–of–the–art algorithms such as the simplex-based methods or interior points methods. Nevertheless, the LP structure could still be exploited to develop more efficient solution algorithms. For example, gradient–based search methods (Spall 2003, Edelkamp and Schroed 2011) or column–row generation procedures (Muter et al. 2012) could be developed and tailored for the DEO framework. The two approaches could also be considered together in a unique solution framework in which the reduced LP of the column–row generation algorithm is solved by gradient–based methods.

4 ASYMPTOTIC PROPERTIES

Herein, we present the asymptotic analysis of our proposed integrated simulation–optimization algorithm. The basic idea to prove the convergence is to consider our algorithm as a case of Sample Path Optimization and use the results in (Robinson 1996) that provides the asymptotic characterization of sample path optimization algorithms. In order to do so, the proposed analysis is made of thee main parts: 1) analysis of the second order properties in the context of simulation and optimization; 2) analysis of the feasible region asymptotic behavior; 3) application of the results in (Robinson 1996) to the integrated simulation–optimization model. The second order properties of the considered optimization models and related simulation models guarantee the regularity conditions at the basis for constraint classification and existence results. The second part of the analysis is required since Robinson (1996) does not consider stochastically constrained problems. Once part 1) and 2) are characterized, we can apply the main results in (Robinson 1996) and prove convergence in our setting.

4.1 Second Order Properties

In this section, we will refer to the time buffer approximate models presented in section 2, i.e., we will assume $\gamma_\nu = 0\ \forall \nu \in \mathbb{E}^C$. In particular, we will analyze optimization models ($\alpha_\upsilon = 0, \forall \upsilon$) and will assume the function of the involved time buffers is simply the sum of the time buffer components, i.e., $\beta_\nu = 1\ \forall \nu \in \mathbb{E}^C$. As it will be shown in section 5, this type of objective function reflects several applications.

In this study, we exploit the mathematical programming framework and the concepts presented in Yao and Shanthikumar (1991). In particular, *SIL(sp)*, *SICX(sp)*, *SICV(sp)*, *SDL(sp)*, *SDCX(sp)*, and *SDCV(sp)* represent monotone convexity and concavity notions. They refer to stochastic increasing and linear, stochastic increasing and convex, stochastic increasing and concave, stochastic decreasing linear, stochastic decreasing and convex, stochastic decreasing and concave, in the sample path (*sp*) sense, respectively (Yao and Shanthikumar (1991)).

We indicate with $\mathbf{F}_n \subset \mathbb{X}_n \times \mathbb{R}_+^n \times \mathbb{R}_+$ the *feasible* region for the approximate optimization problem (for a finite sample path of size n), where $\mathbb{X}_n$ is the domain for the time buffer s_v, $\mathbb{R}_+^n$ is the domain for the finishing times and $\mathbb{R}_+$ the domain for the ε. Since the results will basically focus on the time buffer s_v rather than the event times e_v, it is useful to define the projection of the feasible region $\mathbf{F}_n$ onto the time buffer space $\mathbb{X}_n$. We will refer to this set representing the sample path feasible time-buffer configurations as Σ_n. Moreover, to simplify the notation and focus on the sample size n, we will refer to the time buffers as s_n, dropping the v subscript.

The primal (on the left) and the dual (on the right) approximate optimization models, in their matrix forms, are the following:

$$
\begin{array}{rlr}
\min\ \mathscr{S}_n = \mathbf{1}^{'}\mathbf{s}_n + \vartheta \cdot \varepsilon & \max\ \mathbf{b}^1(\tau,\upsilon)\mathbf{u}_{\mathscr{D}} + \mathbf{b}^2(\tau,\upsilon)\mathbf{u}_{\mathscr{P}} - \mu^* \cdot \nu & (5) \\
\text{s.t.} & & \\
\mathbf{A}^{\mathscr{D}} e_\mathrm{v} \geq \mathbf{b}^1(\tau,\upsilon) & \mathbf{A}^{\mathscr{D}'}\mathbf{u}_{\mathscr{D}} \leq \mathbf{0} & (6) \\
\mathbf{A}^{\mathscr{P}}\left[e_\mathrm{v}|\mathbf{s}_n\right] \geq \mathbf{b}^2(\tau,\upsilon) & \mathbf{A}^{\mathscr{P}'}\mathbf{u}_{\mathscr{P}} \leq \mathbf{1} & (7) \\
\varepsilon - \sum_{\mathrm{v}\in\mathbb{W}^C} p\left(e_\mathrm{v}\right) \geq -\mu^* & \nu \leq \vartheta & \\
e_\mathrm{v} \geq 0, \varepsilon \geq 0,\ \mathbf{s}_n \in \mathbb{X}_n & \mathbf{u} \geq 0, \nu \geq 0 &
\end{array}
$$

The vector $\mathbf{v}^{'} = [e_\mathrm{v}|\mathbf{s}_n]^{'}$, where $[\cdot|\cdot]'$ is the row vector obtained by the concatenation of two column vectors, is the set of decision variables of the primal model, i.e., the time buffer $\mathbf{s}$ and the event times $\mathbf{e}$, while $\mathbf{u}^{'} = [\mathbf{u}_{\mathscr{D}}|\mathbf{u}_{\mathscr{P}}]$ and ν represent the dual variables.

The matrix $\mathbf{A} = \left[\mathbf{A}^{\mathscr{D}}|\mathbf{A}^{\mathscr{P}}\right]$ is an $l \times m$ dimension matrix, where l represents the number of constraints not including the performance constraint(s) (2) and m the number of decision variables. According to the definitions provided in section 2, constraints (6) and (7) are the same as (3) and (4), respectively.

The m-dimensional vector of the right hand side $\mathbf{b} = \left[\mathbf{b}^1|\mathbf{b}^2\right] = \{b_1, b_2, \ldots, b_m\}$ consists of the realization of random variables, in compact form, A_i $(i = 1,\ldots,n)$ and B_{ij} $(i = 1,\ldots,n\ j = 1,\ldots,J)$. These random variables are assumed to follow univariate distributions (following the approach in Shaked and Shanthikumar (2007)) $A_i \sim \mathscr{V}^A(\upsilon)$ and $B_{ij} \sim \mathscr{V}^B(\tau)$, respectively. The link between the realizations and the parameters of the sampling distribution is made explicit through the notation $\mathbf{b}(\tau,\upsilon)$. In section 5, some examples will show that these stochastic variables can model arrival times as well as processing times of jobs in a production system. Nevertheless, several processes can apply to the same definition.

The objective of the primal problem, $\mathscr{S}_n(\mathbf{s}_n(\mathbf{b}(\tau,\upsilon),\mu^*), \varepsilon(\mathbf{b}(\tau,\upsilon),\mu^*))$, is a function of the time buffer $\mathbf{s}_n$, which is itself a function of the right hand side $\mathbf{b}$.

The same modeling approach can be applied for the simulation model.

$$
\begin{array}{rrl}
 & \min \chi = \mathbf{1}^{'} e_\mathrm{v} & \max \mathbf{q}(\tau,\upsilon)^{'}\mathbf{u} \\
\text{s.t.} & & \\
 & \mathbf{A}e_\mathrm{v} \leq -\mathbf{q}(\tau,\upsilon) & \mathbf{A}^{'}[\mathbf{u}] \leq \mathbf{1} \\
 & e_\mathrm{v} \geq 0 & \mathbf{u} \geq 0
\end{array}
$$

$\mathbf{A}$ is the same matrix as in the approximate optimization model while vector $\mathbf{q}(\tau,\upsilon)$ is the concatenation of vectors $\mathbf{t}$ and $\mathbf{s}_n$, i.e., $\mathbf{q}(\tau,\upsilon) = [-\mathbf{t}(\tau,\upsilon)|\mathbf{s}_n(\tau,\upsilon)]$. The notation $\mathbf{q}(\tau,\upsilon)$ has exactly the same interpretation of $\mathbf{b}(\tau,\upsilon)$. Event times $\{e_{ij}\}$ are the variables of the primal model, while $\mathbf{u}$ represents the vector of dual variables. We will refer to the primal objective function as $\chi(\mathbf{s}_n,\mathbf{t})$. Furthermore, when interested in studying the behavior of function χ with respect to only $\mathbf{s}_n$ or $\mathbf{t}$ we will use $\chi(\mathbf{s}_n,\cdot)$ ($\mathbf{s}_n$ fixed, $\mathbf{t}$ variable)

and $\chi(\cdot,\mathbf{t})$ ($\mathbf{s}_n$ variable, $\mathbf{t}$ fixed).
The formulation just presented allows to explicitly indicate the set of control constraints (equations (4)) having the time buffers as right hand side and the natural dynamics constraints (equations (3)) not containing it. Again, the dependency of both $\mathbf{t}$ and $\mathbf{s}_n$ on the parameters τ, υ is made explicit through the notation $\mathbf{t}(\tau,\upsilon)$ and $\mathbf{s}_n(\tau,\upsilon)$.

In the following, we will consider only random variable B_{ij}, i.e., $A_i = 0$, $\forall i$. This will simplify the notation in $\mathbf{t}(\tau)$, $\mathbf{s}_n(\tau)$, $\mathbf{b}(\tau)$ and $\mathbf{q}(\tau)$. Moreover, the parameter of interest τ will be defined in a convex set T (e.g., an interval real line).
In order to proceed with the analysis, we need to make some assumptions on the form of the performance function $p(e_\nu)$ and on the system meeting the target performance as $n \to \infty$.

Assumption 1 $p(e_\nu)$ is a convex function of e_ν and we estimate $\mathbb{E}(p(e_\nu))$ through a sample average of the realizations of the function values in the sample path.

Assumption 2 The system under analysis is stationary and the target performance μ^* is such that $\mu^* \geq \mu_{min}$ being μ_{min} the best performance that can be reached by the system in a steady state.

Property 1 (Second order properties for $\mathscr{S}_n$) If $\{\mathbf{b}(\tau)\} \in$ *SICX(sp)*, then $\{\mathscr{S}_n^*(\mathbf{b})\} \in$ *SICX(sp)*.

Proof. The proof simply relies on the strong duality that holds in the case of LP models as approximate time buffer models are. According to strong duality, we can prove that the function is increasing and convex, namely (increasing):

$$\mathscr{S}_n^*(\mathbf{b}_1) = \mathbf{b}_1^{'}\mathbf{u}^{(1)*} - \mu^*\nu^{(1)*} \leq \mathbf{b}_2^{'}\mathbf{u}^{(1)*} - \mu^*\nu^{(1)*} \leq \mathbf{b}_2^{'}\mathbf{u}^{(2)*} - \mu^*\nu^{(2)*} = \mathscr{S}_n^*(\mathbf{b}_2),$$

whereas, for convexity:

$$\begin{aligned}\mathscr{S}_n^*(\beta\mathbf{b}_1 + (1-\beta)\mathbf{b}_2) &= (\beta\mathbf{b}_1^{'} + (1-\beta)\mathbf{b}_2^{'})\mathbf{u}^{(1)*} - \mu^*\nu^{(1)*}\\ &= \beta\mathbf{b}_1^{'}\mathbf{u}^{(1)*} + (1-\beta)\mathbf{b}_2^{'}\mathbf{u}^{(1)*} - \mu^*\nu^{(1)*}\\ \leq \beta\left(\mathbf{b}_1^{'}\mathbf{u}^{(2)*} - \mu^*\nu^{(2)*}\right) &+ (1-\beta)\left(\mathbf{b}_2^{'}\mathbf{u}^{(3)*} - \mu^*\nu^{(3)*}\right)\\ &= \beta\mathscr{S}_n^*(\mathbf{b}_1) + (1-\beta)\mathscr{S}_n^*(\mathbf{b}_2).\end{aligned}$$

The SICX property is a result from Shaked and Shanthikumar (1988) (Proposition 3.2, pag. 433). □

Property 2 (Second order properties for χ) If the time buffer sequence $\{\mathbf{s}_n(\tau)\} \in$ *SICX(sp)*, then the approximate simulation objective function $\chi^*(\mathbf{s}_n,\cdot) \in$ *SDCX(sp)*. If the processing time sequence $\{\mathbf{t}(\tau)\} \in$ *SICX(sp)*, then the approximate simulation objective function $\chi^*(\cdot,\mathbf{t}) \in$ *SICX(sp)*.

Proof. The same reasoning of Property 1 applies. □

Corollary 1 The average performance $\{\hat{\mu}(\mathbf{s}_n,\mathbf{t})\} \in$ *SICX(sp)* in the processing times $\mathbf{t}$. The average performance $\{\hat{\mu}(\mathbf{s}_n,\mathbf{t})\} \in$ *SDCX(sp)* in the time buffer $\mathbf{s}_n$.

Proof. According to Assumption 1, the expected performance is estimated by convex operations. Since χ^* is *SICX(sp)* in the processing times and *SDCX(sp)* in the time buffers (Property 2) and both *SICX(sp)* and *SDCX(sp)* are closed with respect to monotonic convex operations (Yao and Shanthikumar 1991), then the average performance is *SICX(sp)* with respect to the processing times and *SDCX(sp)* with respect to the time buffer. □

4.2 Constraints Characterization

Let $\Lambda(\mathbf{s},B)$ and $\hat{\Lambda}(\mathbf{s}_n,\mathbf{b})$ be the expected value of the target performance μ and its estimator, namely:

$$\Lambda(\mathbf{s},B) \triangleq E_B[\mu], \qquad \hat{\Lambda}(\mathbf{s}_n,\mathbf{b}) \triangleq \frac{1}{n}\sum_{i=1}^{n} p(e_V(i)) \tag{8}$$

Let $\lambda(\mathbf{s},B)$ and $\hat{\lambda}(\mathbf{s},\mathbf{b})$ be the difference between the expected value of the actual performance and the target performance and its estimator, namely:

$$\lambda(\mathbf{s},B) \triangleq \Lambda(\mathbf{s},B) - \mu^*, \qquad \hat{\lambda}(\mathbf{s}_n,\mathbf{b}) \triangleq \hat{\Lambda}(\mathbf{s}_n,\mathbf{b}) - \mu^*. \tag{9}$$

Estimates defined in (8) and (9) are sample averages. The expected values are functions of the time buffer $\mathbf{s}$ and of the collection of random variables B. We will denote the expected values with $\Lambda(\mathbf{s},B)$ and $\lambda(\mathbf{s},B)$ when we want to stress that the described property is related to the considered random variables (i.e., the distribution taken into account), whereas the notation $\Lambda(\mathbf{s},\cdot)$ and $\lambda(\mathbf{s},\cdot)$ will be adopted in case the property is independent from the distributions.

Lemma 1 The following holds for the approximate optimization model: (i) $\Lambda(\mathbf{s},B)$ is Lipschitz continuous in the domain of $\mathbf{s}$ for P_B almost all $\tau \in \mathbb{R}^{||\tau||}$. Then there exists a function $\Pi : \mathbb{R}^{||\tau||} \to \mathbb{R}$ such that $||\Lambda(\mathbf{s}_1,B) - \Lambda(\mathbf{s}_2,B)|| \leq \Pi(\tau)||\mathbf{s}_1 - \mathbf{s}_2||$, for P_B almost all $\tau \in \mathbb{R}^{||\tau||}$ and such $\Pi(\tau)$ is integrable; (ii) The moment generating function of $\Pi(\tau)$, denoted as $M_{\Pi(\tau)}(l)$, is finite for all the l in a neighborhood of 0.

Proof. Note that $\Pi(\tau)$ is a stochastic function since it is related to the distance between random variables $||\mathbf{s}_1 - \mathbf{s}_2||$. Corollary 1 proves that $\Lambda(\mathbf{s},B)$ is convex in the parameters τ of distribution $\mathscr{V}^B$ of B, hence it is always possible to define function Π. In addition, if the system is stationary and $\mu^* \geq \mu_{min}$, then $\Lambda(\mathbf{s},B) < \infty$ *with probability 1*, hence function Π is integrable. Corollary 1 also guarantees that, if the difference $||\mathbf{s}_1 - \mathbf{s}_2||$ is finite, the difference $||\Lambda(\mathbf{s}_1,B) - \Lambda(\mathbf{s}_2,B)||$ is also finite.
For convexity, finiteness and compactness of the set Σ (section 4), the moment generating functions of $\Pi(\tau)$ is finite in a neighborhood of 0 (Billingsley 1999). □

We are now ready to characterize the relationship between $\hat{\lambda}(\cdot,\cdot)$ and $\lambda(\cdot,\cdot)$. First, we separately analyze $\hat{\lambda}(\cdot,\cdot)$ and $\lambda(\cdot,\cdot)$, in order to verify their properties. We then study the distance between the two functions as the sample path increases.

Proposition 1 Let the function $\pi = E\left[\Pi(\tau)\right]$ be the expectation of $\Pi(\tau)$ and π_n be the sample average approximation of $E\left[\Pi(\tau)\right]$, i.e., $\pi_n \triangleq \frac{1}{n}\sum_{j=1}^{n}\Pi_j(\tau)$: 1) If function $\lambda(\mathbf{s})$ is bounded on Σ, $\hat{\lambda}(\mathbf{s}_n,\cdot)$ is P_B–almost surely bounded on Σ. 2) If function $\lambda(\mathbf{s},\cdot)$ is Lipschitz continuous on Σ, $\hat{\lambda}(\mathbf{s}_n,\cdot)$ is P_B–almost surely Lipschitz continuous on Σ.

Proof. Let $\mathbf{s}_0 \in \Sigma$ be a time buffer configuration. Lemma 1 leads to the following chain of inequalities for the infinite sample path problem:

$$\begin{array}{rcl} E\left[\lambda(\mathbf{s},B)\right] \leq & E\left[|\lambda(\mathbf{s}_0,B)| + \Pi(\tau)||\mathbf{s}-\mathbf{s}_0||\right] & \\ 0 \leq & |\lambda(\mathbf{s}_0,B)| + E\left[\Pi\right]\max_{\mathbf{s}_1,\mathbf{s}_2\in\Sigma}||\mathbf{s}_1-\mathbf{s}_2|| & = \\ = & |\lambda(\mathbf{s}_0,B)| + \pi\max_{\mathbf{s}_1,\mathbf{s}_2\in\Sigma}||\mathbf{s}_1-\mathbf{s}_2|| & < \infty \end{array} \tag{10}$$

The inequalities $|\lambda(s_0,B)| < \infty$ and $\Pi < \infty$ hold because of Lemma 1, whereas $\max_{\mathbf{s}_1,\mathbf{s}_2\in\Sigma}||\mathbf{s}_1-\mathbf{s}_2||$ (the *deviation* of the set Σ) is finite because the set is compact. This proves equation (10) holds, i.e., $|\lambda(s_0,B)|$ and π are both finite. The chain of equalities $E\left[\lambda(\mathbf{s},B)\right] = E\left[\varepsilon\right] = 0$ is guaranteed by any target performance satisfying $\mu^* \geq \mu_{min}$. In fact, from Lemma 4, as $n \to \infty$, we are guaranteed the solution set is not empty and $\varepsilon^* = 0$ exists.

The same result can be proved for function $\hat{\lambda}_n(\mathbf{s},\mathbf{b})$. We can rewrite the chain of inequalities for the finite sample path problem. Simply by considering that in the second inequality we have α instead of 0. In the finite sample path case, we cannot guarantee that $\hat{\varepsilon}^* = 0$, i.e., in general, $E\left[\hat{\lambda}(\mathbf{s},\mathbf{b})\right] = E\left[\hat{\varepsilon}\right] = 0$ does not hold.

Lipschitz continuity holds as a consequence of Corollary 1. Indeed, function $\lambda(\mathbf{s},B)$ is increasing convex in the processing times, realization of the random variables in B, and it is decreasing convex in the multidimensional array $\mathbf{s}$ (the time buffer). □

Lemma 2 $\sup\left\{|\lambda(\mathbf{s},B)-\hat{\lambda}(\mathbf{s},B)| : \mathbf{s}\in\Sigma\right\} \rightarrow 0$ as $n\rightarrow\infty$ almost surely. As a result, $\hat{\lambda}(\mathbf{s})$ converges to $\lambda(\mathbf{s})$ *uniformly* on Σ almost surely.

Proof. The proof relies on the result from Proposition 1. □

Lemma 2 shows that the sample average estimator is an unbiased estimator of the performance expected value and, as a result, the sample path constraint set converges to the true constraints set (Shapiro 2003).

4.3 Asymptotic Convergence

In the following, we will use the notation $\mathbf{s}$ and $\mathscr{S}$ to indicate the infinite sample path solution and objective function, respectively, while $\mathbf{s}_n$ and $\mathscr{S}_n$ will indicate the optimal time buffer and the objective function of the finite sample path case, as in the previous sections. In order to apply the fundamental results in (Robinson 1996), we need to show that Assumptions *A*–*B* in Definitions 2.2 and 2.3 in (Robinson 1996) hold and this is proved by

Property 3 (Assumption *A*, Definition 2.2 in (Robinson 1996), page 517) Function $\mathscr{S}_n$ satisfies the following conditions: (a) for each $1\leq n<\infty$, $\mathscr{S}_n$ is lower semi–continuous. (b) $\mathscr{S}_n \rightarrow_{epi} \mathscr{S}$, i.e., it epiconverges to $\mathscr{S}$.

Proof. The function $\mathscr{S}_n$ is lower semi–continuous *iff* the related epigraph, $E_{\mathscr{S}_n}$, is closed (Attouch 1984). The epigraph of the function $\mathscr{S}_n$ represents the set of solutions having a value of the objective function smaller than some predefined $\phi > \mathscr{S}_n^*$, i.e., $E_{\mathscr{S}_n} = \{(\mathbf{s}_n,\phi) : \mathscr{S}_n(\mathbf{s}_n)\leq\phi\}$. In particular, let $\mathbf{s}_n$ be a feasible solution and ϕ be a real positive value. The function $\mathscr{S}_n : \Sigma_n \rightarrow \mathbb{R}_+$ is linear in $\mathbf{s}_n$ and it is defined over the compact set Σ_n. As a result, there exists an arbitrarily small $\delta(\phi)\in\mathbb{R}$ such that the neighborhood of (feasible) solutions $\mathbf{s}_n+\delta$ are outside the epigraph $E_{\mathscr{S}_n}$, thus $E_{\mathscr{S}_n}$ is a closed set. This proves that $\mathscr{S}$ is lower semi–continuous.

The function $\mathscr{S}_n \rightarrow_{epi} \mathscr{S}$ if, in addition to lower–semicontinuity, it uniformly converges to the infinite sample path function $\mathscr{S}$ on compact subsets of the function domain Σ. Since the random variables in B are such that $P(B_{ij}\geq\infty) = 0$, then the objective function is finite $\mathscr{S}\leq\infty$ a.s., hence the function is proper. In addition, the objective function is linear in $\mathbf{s}$, convex in the processing times (Property 1) and it is defined over a compact set Σ_n. Since, from Lemma 2, we know that $\Sigma_n\rightarrow\Sigma$ uniformly, we have uniform convergence. □

Property 4 (Assumption *B*, Definition 2.3 in (Robinson 1996), page 517) $\mathscr{S}$ is proper and the set of minimizers of the finite sample path optimization problem $\mathbb{S}_n^* = \{\mathbf{s}_n\in\Sigma_n | \mathscr{S}_n = \mathscr{S}_n^*\}$ is not empty and it is compact.

Proof. The problem in (5) always admits a feasible solution for construction. Let μ_{min} be the best performance that can be reached by the system in a steady state. If the system is stationary, a finite μ_{min} exists. As a result, the problem solution set is non–empty. Under Assumption 2, as the sample path size goes to ∞, the optimal solution is characterized by $\varepsilon^* = 0$. □

Theorem 1 (Convergence of $\mathscr{S}$, from Theorem 3.2 in (Robinson 1996), page 519) The minimizer $\mathbf{s}_n^*$ epi–converges to the infinite sample path solution $\mathbf{s}^*$ with probability 1. The related objective function value $\mathscr{S}_n^*$ converges uniformly to the infinite sample path objective function value $\mathscr{S}^*$.

Proof. From Property 4, the optimal solution to the infinite–sample path optimization problem exists and it is finite, $\mathbf{s}^*$. The ε-problem is developed in a way such that a minimum to the sample path problem always exists, i.e., for each n, the sample–path solution $\mathbf{s}_n$ is such that $s_n \in \mathbb{S}_n^*$. As $n \to \infty$ this solution converges to a limiting value $\mathbf{s}$.

Let $\mathscr{S}_n^*$ be the value of the objective function at the optimum: $\mathscr{S}_n^* = \inf \mathscr{S}_n$. Function $\mathscr{S}_n$ satisfies epi–convergence (Property 3 and 4). Given Lemma 2, we can use epi–convergence to prove that $\mathbb{S}_n^*$ converges to $\mathbb{S}^*$, i.e., $\mathbf{s}_n^*$ converges to $\mathbf{s}^*$.

Let Γ be the set defined as $\Gamma = \{\mathbf{b} : \sup\{b_l\} = \infty\}$ (note that Γ has measure 0 under Assumption 2). For every realization $\mathbf{b}$ of the processing times such that $\mathbf{b} \notin \Gamma$, the following holds (this result is in Theorem 3.2 in (Robinson 1996), page 519):

(a) $\mathscr{S}^* \leq \lim\sup \mathscr{S}_n^*$;
(b) if $\mathbf{s}_n$ is a sequence converging to $\mathbf{s}$ and if, for each n, $\mathbf{s}_n \in \mathbb{S}_n^*$, then $\mathbf{s} \in \mathbb{S}^*$.

Epi–convergence results in $\mathbf{s} \in \mathbb{S}^*$, hence the sample path optimal solution converges to the infinite sample path optimal solution. □

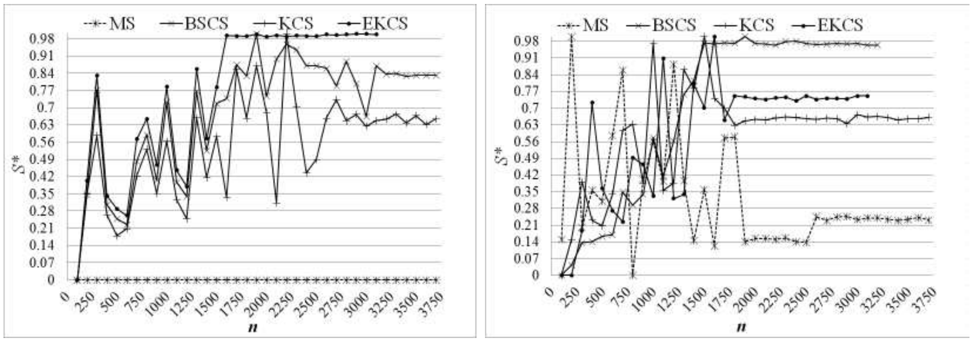
5 NUMERICAL ANALYSIS

Herein, we analyze the impact of the complexity and the size of the problem together with the main parameters. In order to do so, we consider the multi–stage systems in Table 1.

Table 1: Test Problems

System	Objective	Performance
Multi–Stage	Minimize Buffer Capacity	System Throughput
Kanban–Controlled	Minimize Kanban Tokens	Service Level
Base–Stock–Controlled	Minimize Base Stock Level	Service Level
Extended–Kanban–Controlled	Minimize the Base Stock Level & Kanban Tokens	Service Level

We solved these problems by approximating buffer capacities, kanban tokens and inventory levels with the related time-buffers. More details concerning the implementation of the framework to these cases are available in (Pedrielli et al. 2015, Alfieri and Matta 2012a). Figure 1 shows the convergence of the objective ($\mathscr{S}^*$) function and the related computational effort when the systems in Table 1 are subject to different saturation conditions. In particular, in Figure 1(a), the multi–stage system is required to produce at a rate of 0.65 [jobs/time unit], and, equivalently, we set an arrival rate into the pull system equal to 0.65 [jobs/time unit]. Figure 1(b) refers to a rate of 0.9 [jobs/time unit]. All the Test Problems refer to three-stage systems, i.e., $J = 3$. From Figure 1, we can observe that the multi–stage buffer allocation problem (MS in Figure 1(a)), shows the fastest convergence, weather base–stock and kanban systems appear to be more oscillating. This phenomenon can be brought back to the higher influence of the initial conditions on kanban and base–stock systems, which cause the solution to be more influenced by the initial generation of random variables negatively affecting the rates. Figure 1(b) shows that the convergence is faster, with respect to the previous cases; this is due to the fact that more stringent constraints dramatically reduce the solution space making the procedure converge faster. We also observed the required solution time. Systems of increasing complexity also require an increased computational power (e.g., EKCS), whereas more constrained systems correspond to less computational time.



(a) $\mu^* = 0.65$

(b) $\mu^* = 0.90$

Figure 1: Empirical Convergence

In these experiments, even for a small number of entities, the computational effort shows an exponential growth. As already stated in section 3, techniques that can still make use of information coming from the mathematical model, but use faster solution algorithms, will be beneficial to the framework.

6 CONCLUSIONS

In this paper, we gave a comprehensive overview of the integrated simulation–optimization framework based on mathematical programming representations. The main definitions and guidelines to develop integrated simulation–optimization models are provided together with the algorithm to solve them. The asymptotic properties and examples presented encourage the design and analysis of efficient methodologies for the solution of integrated simulation–optimization models. Indeed, if, on the one hand, the presented algorithm shows empirical convergence, its implementation requires to solve mathematical programming models. More efficient techniques can be used which exploit the mathematical modeling while providing a more efficient procedure (i.e., leading to possibly faster convergence rates). Such procedures might be applied not only to the approximate but also to the original problems. In particular, time buffer models could be interpreted as low fidelity versions of their integer analogue. As such, the approximate integer solution from the time buffer model, might indeed be used as initial solution for the integer counterpart. Nevertheless to make such framework effective, algorithmic efficiency needs to be improved.

REFERENCES

Alfieri, A., and A. Matta. 2012a. “Mathematical programming formulations for approximate simulation of multistage production systems”. *European Journal of Operational Research* 219 (3): 773 – 783.

Alfieri, A., and A. Matta. 2012b. “A Time-Based Decomposition Algorithm for Fast Simulation with Mathematical Programming Models”. In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher: Berlin, Germany.

Alfieri, A., A. Matta, and G. Pedrielli. 2013. “Mathematical Programming formulations for approximate simulation optimization of closed–loop systems”. *Annals of Operations Research*.

Attouch, H. 1984. *Variational Convergence of Functions and Operators*. Pitman, London.

Billingsley, P. 1999. *Convergence of probability measures; 2nd ed.* Wiley Series in Probability and Statistics. Hoboken, NJ: Wiley.

Chan, W., and L. Schruben. 2008a. “Optimization models of Discrete–Event System Dynamics”. *Operations Research* 56 (5): 1218–1237.

Chan, W. K., and L. W. Schruben. 2008b. “Mathematical programming models of closed tandem queueing networks”. *ACM Transactions on Modeling and Computer Simulation* 19 (1).

Edelkamp, S., and S. Schroed. 2011. *Heuristic Search: Theory and Applications*. Morgan Kaufmann.

Fu, M., F. Glover, and J. April. 2005. “Simulation optimization: a review, new developments, and applications”. In *Proceedings of the 2005 Winter Simulation Conference*.

Healy, K., and L. W. Schruben. 1991. “Retrospective simulation response optimization”. In *Proceedings of the 23rd conference on Winter simulation*.

Jin, J., and B. Schmeiser. 2003. “Simulation-based Retrospective Optimization of Stochastic Systems: a Family of Algorithms”. In *Proceedings of the 2003 Winter Simulation Conference*.

Kolb, O., and S. Gttlich. 2015. “A continuous buffer allocation model using stochastic processes”. *European Journal of Operational Research* 242 (3): 865 – 874.

Matta, A., G. Pedrielli, and A. Alfieri. 2014. “ERG Lite: Event Based Modeling for SimulationOptimization of Control Policies in Discrete Event Systems”. In *Proceedings of the 2014 Winter Simulation Conference*.

Muter, I., S. Birbil, and K. Bulbul. 2012. “Simultaneous Column-and-Row Generation for large-scale linear programs with column-dependent-rows”. *Mathematical Programming* 142(1-2):47–82.

Pedrielli, G. 2013. *Discrete Event Systems Simulation–Optimization: Time Buffer Framework*. Ph. D. thesis, Mechanical Engineering Department, Politecnico di Milano, Italy.

Pedrielli, G., A. Matta, and A. Alfieri. 2015. “Integrated Simulation–Optimization of Pull Control Systems”. *International Journal of Production Research*:To Appear.

Plambeck, E. L., B. ruey Fu, S. M. Robinson, and R. Suri. 1996. “Sample-path optimization of convex stochastic performance functions”. *Mathematical Programming* 75:137–176.

Robinson, S. 1996. “Analysis of Sample–Path Optimization”. *Mathematics of Operations Research* 21:513–528.

Schruben, L. W. 2000. “Mathematical Programming Models of Discrete Event System Dynamics”. In *Proceedings of the 2000 Winter Simulation Conference*.

Shaked, M., and J. Shanthikumar. 2007. *Stochastic orders*. Springer series in statistics. Springer.

Shaked, M., and J. G. Shanthikumar. 1988. “Stochastic Convexity and Its Applications”. *Advances in Applied Probability* 20 (2): pp. 427–446.

Shapiro, A. 2003. “Monte Carlo sampling methods”. *Handbooks in operations research and management science* 10:353–425.

Spall, J. C. 2003. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley.

Tan, B. 2015. “Mathematical programming representations of the dynamics of continuous-flow production systems”. *IIE Transactions* 47 (2): 173–189.

Weiss, S., and R. Stolletz. 2015. “Buffer allocation in stochastic flow lines via sample-based optimization with initial bounds”. *OR Spectrum*:1–34.

Yao, D., and J. G. Shanthikumar. 1991. “Strong stochastic convexity: Closure properties and applications”. *Journal of Applied Probability* 28 (1): pp. 131–145.

AUTHOR BIOGRAPHIES

GIULIA PEDRIELLI is Research Fellow for the Centre for Maritime Studies at the National University of Singapore. Her research focuses on stochastic simulation-optimization in both single and multiple–objectives framework. Her email address is cmsgp@nus.edu.sg.

ANDREA MATTA is Distinguished Professor at the Institute of Industrial Engineering at Shanghai Jiao Tong University, where he currently teaches stochastic models and simulation. His research area includes analysis and design of manufacturing and health care systems. His email address is matta@sjtu.edu.cn.

ARIANNA ALFIERI is Associate Professor at Politecnico di Torino, where she currently teaches production planning and control and system simulation. Her research area includes scheduling and planning in production and transportation systems. Her email address is arianna.alfieri@polito.it.