Multi-Fidelity Modeling for Analysis and Optimization of Serial Production Lines

Yunyi Kang, *Student Member, IEEE,* Logan Mathesen, Giulia Pedrielli, *Member, IEEE*, Feng Ju, *Member, IEEE*, Loo Hay Lee, *Senior Member, IEEE*

***Abstract*—Recent advances in sensing, data analytics and manufacturing technologies (e.g., 3D printing, soft robotics, nanotechnologies) provide the potential to produce highly customized products by allowing flexible system design, endless device configurations, and unprecedented information flows. These opportunities also increase the complexity of controlling such systems optimally, which typically requires fast exploration of an increasingly large number of alternative operation strategies. Simulation and stochastic models have been particularly successful to support control and optimization of production systems, and methods have been developed to exploit them separately. Herein, we argue that the simultaneous use of these models can allow for better control and optimization by balancing the simulation accuracy, and related high computational costs, with the computational efficiency and lower accuracy of stochastic models.**

In this paper, we assume that high fidelity models have higher accuracy and computational costs, and we present a novel multi-fidelity approach, which utilizes several models at different levels of fidelity to efficiently and effectively estimate and optimize the performance of asynchronous serial production lines with machines suffering multiple failure types. Experimental results show that the multi-fidelity approach leads to better estimations, requiring less computational effort for optimization compared with the use of only high fidelity simulations.

***Index Terms*—Multi-fidelity modeling, simulation-optimization, manufacturing, serial production line**

I. Introduction

The design and operation of increasingly complicated manufacturing systems, and the related processes, require the development of methods for optimization and control that allow for the efficient evaluation and optimization of the system performance [1]. Nevertheless, simulation based and exact approaches have developed independently: (i) exact approaches generate solutions based on assumptions that hinder their implementation in the real system, (ii) simulation based approaches require less to no assumptions, but take a long time to run, and rarely provide finite time guarantees on the quality of the solution.

As a motivating example, consider a serial production system composed of machines with controllable processing time (inverse of capacity/speed). In theory, maintaining higher machine capacity will reduce the cycle time to produce a part. However, the higher the capacity the machine has, the more likely it will fail due to an accelerated degradation process. For instance, increasing the speed of a polishing machine will shorten the time to process the surface of a part, at the cost of increased bearing fault [2]. Therefore, machine capacity combinations should be carefully selected to maximize the overall system production rate, and such combinations should be identified quickly to enable online control of machine capacities given the dynamic production environment [3].

In the scope of fast optimization, models with high accuracy and low computational cost, coupled with optimization procedures, are of critical importance. However, such models are generally not available. Analytical models, when used for optimization, can lead to bad solutions, especially when the system is complex, due to an error that is typically not homogeneous across the solution space. Simulation models have great accuracy, but the large computational cost of running them effectively hinders their deployment into real time production control settings.

The research area developing methods that attempt to mix exact analytical and simulation based approaches is referred to as multi-fidelity modeling and optimization, having its foundational contributions, within the simulation and manufacturing literature, in [4], [5]. However, these methods were not designed for continuous solution spaces. In statistical learning theory, multi-fidelity has also attracted important attention, usually assuming that the models of different fidelity can be ranked, from lowest to highest accuracy, whether such rank is known or needs to be learned [6], [7], [8]. Our approach differs from the literature in both areas, by focusing on two aspects: (1) how to generate models at different fidelity when approximating a discrete event simulator, (2) how to develop statistical learning of different model biases to correct them as well as integrate them in a way that does not require models to be ranked according to their accuracy (whether the rank is known or needs to be learned), if not for the highest fidelity. This aspect is important: while a highly detailed simulator can be a high accuracy model, when approximations are performed, we cannot generally guarantee that the error of the models will be homogeneous in the space of the system parameters. As a result, models can have a different rank in different locations of the solution space.

The result of our effort is a novel optimization algorithm

This work is supported in part by grants NSF CMMI-1829238 and NSF CMMI-1922739.

Y. Kang, L. Mathesen, G. Pedrielli, and F. Ju are with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281, USA. Email: ykang37@asu.edu, lmathese@asu.edu, Giulia.Pedrielli@asu.edu, Feng.Ju@asu.edu

L.H. Lee is with the Department of Industrial and Systems Engineering, National University of Singapore, Singapore 119260, Singapore. Email: iseleelh@nus.edu.sg

Please send all correspondence to Dr. Feng Ju

that samples from several models at different, and potentially non-homogeneous, fidelity in order to efficiently provide an estimate of the optimal solution.

The main contribution of this paper is three-fold: (1) develop an automated procedure to produce approximations of a discrete event model, (2) develop novel models that intelligently mix low and high fidelity estimations for improved system performance evaluation; (3) new Bayesian optimization sampling procedures that effectively make use of the multi-fidelity information.

The remainder of the paper is organized as follows: Section II reviews the existing literature. Section III describes the system of interest and formulates the problem and assumptions. In Section IV the new multi-fidelity models, and the associated multi-fidelity optimization framework are presented. Section V provides the numerical studies on multi-fidelity prediction models and the optimization framework. Finally, Section VI is devoted to conclusions and future work.

II. Literature review

A. Performance evaluation of complex production systems

Within the production system research community, an important effort has been devoted to the design of state-based models to describe the evolution of the system in time [9], [10], [11], [12]. However, due to the complexity of the models, closed-form analytical expressions are available only for short production lines, for example, two machines and one buffer lines. For longer lines, researchers developed approximation methods, such as aggregation approaches [13], [14], decomposition methods [15], and numerical approaches [16]. State based models are helpful in predicting system performance when given a state sample path. Nevertheless, these approximation based models typically possess large and heterogeneous estimation bias, making them inappropriate for optimization and control purposes [17]. Besides analytical models, discrete event simulation (DES) has been one of the most successful tools for the analysis of production systems [18], [19], [20]. In fact, simulation models are usually constructed for the verification of analytical models, i.e., they represent the state of the art benchmark [21]. This is due to the fact that simulation models do not require particular assumptions. Therefore, it is feasible to use simulation to model complex systems in terms of the number of components, interactions, and policies which are typically hard to reproduce in analytical settings [22]. Nevertheless, simulation requires high computational effort. In particular, long simulation runs, as well as a large number of replications, are required to obtain accurate estimations of the desired output measure, since shorter simulation runs usually have larger bias and noise. Shorter simulations result in computational savings at the cost of accuracy loss [23], [24]. We refer to high accuracy high cost models as *high fidelity*, while low cost low accuracy representations are referred to as *low fidelity*.

Generally, the literature in performance estimation has mainly focused on the selection and improvement of single models for the target of higher accuracy and lower computational cost. Only recently, the research community has started to shed light on the opportunity coming from the integration of several models that are already available in order to achieve better performance than picking a single unique model [25], [4]. Nevertheless, no general approach has been proposed for this integration. While [25] focuses on a Kernel based estimation framework considering only two models that are assigned a static weight, [4] focuses on the optimization task again only considering two models. Therefore, a method enabling the integration of multiple models for achieving the best compromise between accuracy and computational effort is needed.

B. Black box optimization with sources at multiple fidelity

The problem to determine the combination of machine capacities that maximizes the desired performance of the entire production systems has long been researched in the manufacturing literature [26], [27], [28], [29]. In general, this problem falls within the category of non-linear non convex optimization over continuous domain. Hence, two branches of literature are relevant: (i) optimization of manufacturing systems; (ii) black box optimization. In both, we focus on approaches that consider models with multiple fidelity.

Within the manufacturing literature, an important contribution focusing on multi-fidelity models for the optimal design of manufacturing systems was proposed in [4], where the authors aim at identifying the optimal system configuration for a complex job shop using simulation (high fidelity model) together with a Jackson network (low fidelity model). The resulting algorithm, Multi-Fidelity Optimization with Ordinal Transformation and Optimal Sampling (MO^{2}TOS) works over a finite, discrete, solution space and it exploits the availability of the Jackson network to inform a new version of the Optimal Computing Budget Allocation (OCBA) algorithm [30]. The Ordinal Transformation (OT) uses the low fidelity model to derive a rank space for the alternatives based upon the associated low fidelity performance. The best solution is then searched for in this rank space using the high fidelity model and the optimal sampling to select the alternative to evaluate. Few challenges are not in the scope of the MO^{2}TOS framework: (i) it was not designed for continuous spaces; (ii) only a single low fidelity representation can be exploited; (iii) no approach is provided to derive low fidelity models.

Challenges (i) and (ii) have been addressed in several ways within the statistical learning and Bayesian Optimization communities. Indeed, multi-fidelity optimization has been theoretically investigated not only in discrete [31], but also continuous settings, which is the focus of this work. In general, continuous approaches for multi-fidelity include extended sequential methods [32], and methods for surrogate based optimization that make use of Co-Kriging meta-models in order to construct a prediction for the sources at different fidelity, and use the cross-correlation structure as a means to “transfer” information [33], [7]. The Co-Kriging model was explored in [34] to improve the efficiency of prediction and uncertainty modeling when multiple information sources exist. Co-kriging requires the knowledge of a ranking for the fidelity among the sources, such that an auto-regressive model can be used and a single multi-output Gaussian process constructed to predict the multiple sources. This fidelity ranking and auto regressive modeling is leveraged in several approaches [32], [7]. With quite restrictive assumptions [33] present an approach for the multi-armed bandit problem which minimizes both simple and cumulative regret under the assumption of a set of multi-fidelity models with a known fidelity hierarchy, where the maximum bias of an information source strictly decreases with its fidelity. Though these methods are examples of solutions to (i) and (ii), and allow multiple low fidelity models to be used, they are restrictive in their need for low fidelity models to be ranked in order of accuracy, and again give no approach on how to generate low fidelity approximations.

In our problem setting, we make no such assumption on the existence of any relationship or ranking between low fidelity representations. A similar setting can be found in the recent work on multiple information source optimization with knowledge gradient (misoKG) that extends the knowledge gradient methodology to multiple information sources [35]. There have been two other approaches proposed for the non-hierarchical information source setting including an expected improvement based algorithm proposed by Lam et al. in [36], and multi-task Bayesian optimization (MTBO) [37], which, to the authors' knowledge, was among the first approaches for non-hierarchical multi-fidelity optimization. MTBO proposes a joint Gaussian process to model all information sources by building upon the multi-task Gaussian process regression literature [38], [39], [40], extending the Co-Kriging approach. In MTBO the next sample point is chosen via the cost-sensitive entropy search, sampling points that reduce uncertainty in the optimum location, normalized by the query cost. Notably different from the Co-Kriging approach, the approach of Lam et. al. [36] is to model and maintain a Gaussian process estimation of each individual information source and to then coalesce these models into a single multi-fidelity Gaussian process via the Winkler's method [41]. A modified expected improvement function is then applied to this single multi-fidelity surrogate to determine what location to sample next. However it was shown in [35] that misoKG experimentally has superior performance to both previous algorithms. It is important to highlight that, while misoKG does not require a hierarchical ranking of low fidelity models, its sampling criteria requires to define a cost function associated to each information source. In essence these cost functions, which are generally assumed to be continuous over the input space, implicitly define a ranking over the low fidelity models. We show later that, when there exist multiple low fidelity models without a priori known cost functions, misoKG can be outperformed.

Concerning challenge (iii), no general approach is currently available to derive low fidelity models, and models of different fidelity are constructed by experts and assumed to be available for the optimizer [5], [42], [43]. Since this work looks into discrete event systems, it is relevant to propose ways to generate the models for the optimizer. It is important to develop generators that attempt to maximize the dependency between the low and high fidelity, while guaranteeing a substantial gain in computational effort. In this work, we take inspiration from the automata learning techniques [44], [45], and we propose a model driven approach for the generation of low fidelity for Discrete Event Systems.

III. Problem Description Notation & Terminology

The structure of the class of serial production system considered in this work is shown in Figure 1. The circles and rectangles are used to represent machines and buffers, respectively. The direction of arrows shows the flow of parts throughout the system.

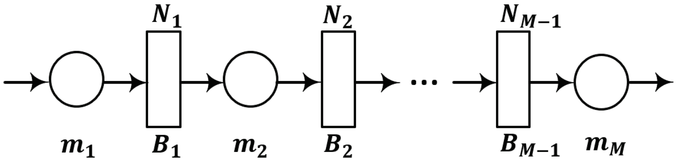


Fig. 1. Illustration of a serial production line

The characteristics of machines, buffers, and their interactions are listed below:

1) In the production system, there are M machines indexed by $k = 1, \ldots, M$, and $M-1$ buffers $(B_1, \ldots, B_{M-1})$ decoupling them;
2) All machines are independent with deterministic processing time, $\tau_k^p, k = 1, \ldots, M$, and corresponding capacity (processing speed) $c_k = 1/\tau_k^p v, k = 1, \ldots, M$.
3) There are two types of time-dependent failures considered, type 1 and type 2. For a machine k, which has been operating for time t since the last repair, the failure rate of type $u = \{1, 2\}$ is $\lambda_{k,u}(t) = \alpha_{u,1}^{(k)} t + \alpha_{u,2}^{(k)}$. Let $\alpha^{(\mathbf{k})} = [\alpha_{11}^{(k)}, \alpha_{12}^{(k)}, \alpha_{21}^{(k)}, \alpha_{22}^{(k)}]$ denote all the values of the failure rate for machine k. We can use an $M \times 4$ matrix to represent the parameter matrix of the failure rate, $\alpha = [\alpha^{(\mathbf{1})}; \alpha^{(\mathbf{2})}; \ldots; \alpha^{(\mathbf{M})}]$.
4) When a failure occurs, repair must be performed, which can fully recover the machine condition to as good as new. The repair time of machine k is exponential with rate R_k independently from the failure type.
5) Buffer B_k has finite capacity N_k, $k = 1, 2, \cdots, M-1$.
6) Machine $k, k = 1, 2, \cdots, M-1$ is blocked if it is up and buffer k becomes full. Machine M is never blocked.
7) Machine k is starved if it is up and buffer $k-1$ becomes empty, $k = 2, 3, \cdots, M$. Machine $k = 1$ is never starved.
8) The failure rate is linear in the capacity. For machine k, the failure rate, as a function of the machine capacity, is denoted as $\Lambda_{k,u}(t) = c_k \times \lambda_{ku}(t), k = 1, \ldots, M, t \geq 0$.

The system in Figure 1, is modeled in high-fidelity using the Event Relationship Graph (ERG) formalism [46], a directed weighted graph. The vertexes of the ERG represent the events that take place in the system and they may correspond to state changes. The directed arcs of the graph represent the "triggering" relationship between the connected pair of events. The state changes associated with each event vertex appear in braces. Arcs can have weights that represent a delay between the triggering (origin of the arc) and triggered event (destination of the arc). An arc can carry a condition (reported in parenthesis), that expresses a constraint that needs to be satisfied when the triggering event is executed for it to trigger the destination event.

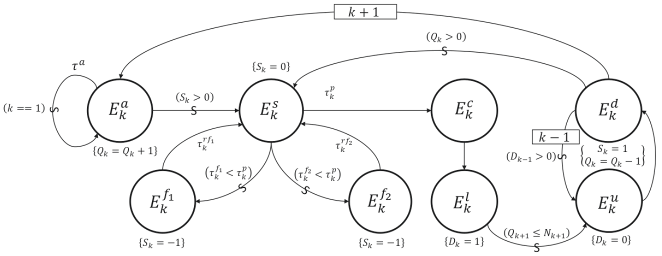


Fig. 2. ERG model for the serial production line

Figure 2 is the ERG representation of our manufacturing system with the 8 characteristics listed above. The model is characterized by eight types of events: $E_k^a, k = 1, \ldots, M$ represent the arrival of a job to the k-th queue, if the server is the first (condition $(k == 1)$ in Figure 2) then another arrival is scheduled with a delay τ^a. In any case, the queue level is increased $Q_k = Q_k + 1$. If a server is available (condition $S_k > 0$), a start event E_k^s is triggered for the server k of interest, with $k = 1, \ldots, M$. When processing starts at a station k, i.e., E_k^s is executed, if the processing time (delay) τ_k^p is larger than the time to the next failure for either of the two failures ($\tau_k^{f_1}$, $\tau_k^{f_2}$, that are generated from the related distribution at each execution), then the corresponding failure event is scheduled $E_k^{f_1}$ or $E_k^{f_2}$, respectively. At this point, the start event, E_k^s, will be scheduled to occur after a random exponential time $\tau_k^{rf_1}$, or $\tau_k^{rf_2}$, depending on the failure type that occurred. When a start event is executed, the server becomes busy ($S_k = 0$), and a completion event E_k^c is scheduled to occur with a delay τ_k^p. Upon completion, the server is blocked (event E_k^l), and the downstream buffer level is checked. If the downstream level is below capacity (condition $Q_{k+1} < N_{k+1}$), the server is unlocked by scheduling the unlocking event at the current time, E_k^u. The execution of E_k^u sets the number of blocked servers D_k at stage k back to 0 ($D_k = 0$), and a departure event is scheduled for the current time E_k^d. A departure schedules an arrival to the next stage $k+1$ as long as $k < M$; otherwise the part leaves the system. At every departure, the state of the upstream server $k-1$ is checked: if the server is blocked, i.e., $D_{k-1} > 0$, then an unlock event for the upstream stage E_{k-1}^u is scheduled for the current time.

The problem to be addressed in this paper is: considering the class of serial production lines with characteristics (1) - (8), and the related simulation model in Figure 2, develop a high accuracy/computationally efficient framework to evaluate and optimize the production rate controlling the machine capacity c_k.

We approach this problem by: (1) proposing a way to generate low fidelity models of the system in Figure 2, and (2) leveraging the high fidelity representation along with the several low fidelity approximations to sequentially sample solutions within the system configuration space (made of the capacities of each server).

IV. Multi-Fidelity Methodology

Formally, we assume the high fidelity simulator and its multiple low fidelity representations to be black-boxes, producing point-wise observations of the function $f^{HF}(\cdot)$ with $f_i^{LF}(\cdot)$, $i = 1, \ldots, n$, when n low fidelity models are given. We aim to optimize the high fidelity response by solving

$$\mathbf{x}^* \in \arg \min_{\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^d} f^{HF}(\mathbf{x}),$$

where $\mathbb{X}$ is a d-dimensional continuous solution space over which all fidelity models are defined. As each observation of $f^{HF}(\mathbf{x})$ is computationally expensive to collect, we aim to identify $\mathbf{x}^*$ using as few high fidelity observations as possible by augmenting them using the n low fidelity models, that provide access to lower quality information at greatly reduced computational costs. We propose novel methods to draw upon these multiple low fidelity information sources and ideally reduce the number of observations of $f^{HF}(\cdot)$ needed to identify $\mathbf{x}^*$. This is accomplished by first evaluating the low fidelities $f_i^{LF}(\mathbf{x}_h), i = 1, \ldots, n$, and the high fidelity function $f^{HF}(\mathbf{x}_h)$ in a set of locations $\{\mathbf{x}_h\}_{h=0}^{n_0}$. These values are then used to statistically model the observed relationship between the low and high fidelity sources, to build an accurate prediction of f^{HF}, which we use for simulation purposes.

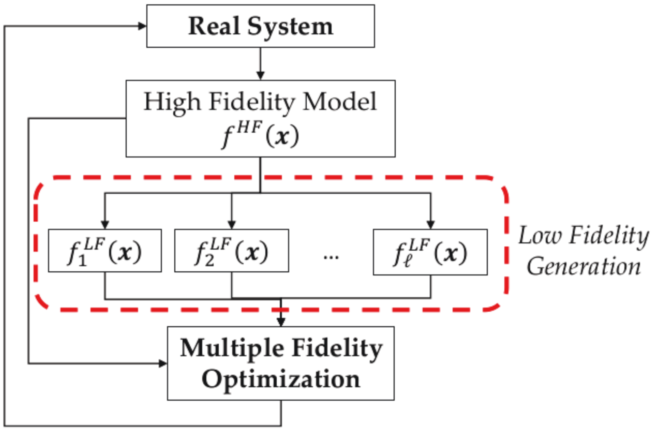


Fig. 3. The Multifidelity Optimization Approach for serial production lines

Figure 3 shows the outline of the approach. Section IV-A presents the alternative methods for generating low fidelity representations. Section IV-B focuses on our proposed multi-fidelity statistical models, while Section IV-C introduces the multi-fidelity optimization approaches.

A. Generating low fidelity models for Serial Production Lines

Once the high fidelity model is available (Figure 2), according to the procedure sketched in Figure 3, the *low fidelity generation* can be performed. In this work, we propose a first principle state-based approximation of the ERG dynamics in Figure 2 (Section IV-A1), and an ERG driven approach for model simplification (Section IV-A2).

1) Analytical model: We first build the analytical model to estimate the desired system performance by introducing a two-step aggregation approach. First, for each machine, we aggregate the multiple failure modes into one. In order to do so, we assume the failure time of each machine follows an exponential distribution, with the mean being the average time to failure from the multiple failures modes model specified in Section III, characteristic 3. We then apply the machine aggregation approach to derive the production rate resulting from a serial production line with geometric machines.

a) Deriving the operational time: Considering the type of failure described in characteristic 3 (Section III), the operational time for each machine $k = 1, \dots, M$ denoted as $\tilde{\tau}_k$ for the machines in the system can be expressed as follows:

$$\begin{aligned}\tilde{\tau}_k = \int_0^\infty & t_0(\lambda_1(t_0) + \lambda_2(t_0)) \times \\ & e^{-\int_0^{t_0}(\lambda_1(t)+\lambda_2(t))dt} dt_0.\end{aligned} \tag{1}$$

We can further simplify the equations leading to:

$$\begin{aligned}\tilde{\tau}_k &= \int_0^\infty t_0\Big((\alpha_{11}t_0 + \alpha_{12}) + (\alpha_{21}t_0 + \alpha_{22})\Big) \times \\ & e^{-\int_0^{t_0}(\alpha_{11}+\alpha_{21})t+(\alpha_{12}+\alpha_{22})dt} dt_0 \\ &= \gamma_1 e^{-\frac{\gamma_3^2}{2}} + \gamma_2(1 - \Phi(\gamma_3)),\end{aligned} \tag{2}$$

where

$$\begin{aligned}\gamma_1 &= \frac{2(\alpha_{12} + \alpha_{22})}{(\alpha_{11} + \alpha_{21})^3} e^{\frac{(\alpha_{12}+\alpha_{22})^2}{2(\alpha_{11}+\alpha_{21})}}, \\ \gamma_2 &= \frac{\sqrt{2\pi}}{(\alpha_{11} + \alpha_{21})^{\frac{5}{2}}} e^{\frac{(\alpha_{12}+\alpha_{22})^2}{2(\alpha_{11}+\alpha_{21})}}, \\ \gamma_3 &= \frac{\alpha_{12} + \alpha_{22}}{\sqrt{\alpha_{11} + \alpha_{21}}},\end{aligned} \tag{3}$$

$\Phi(\cdot)$ is the CDF of a standard normal distribution.

The failure rate for machine k results $P_k = 1/\tilde{\tau}_k$. Besides, the repairing rate R_k for machine k follows the descriptions of the characteristic 4) in Section III.

b) Deriving the production rate: Let us first consider a two-machine-one-buffer line, defined by characteristic 1-8. A continuous time and discrete state Markov chain, with states defined as a combination of machine state and buffer level is proposed to model the line. According to [12], the production rate, denoted as $\hat{f}^{LF}(\mathbf{x}), \mathbf{x} \in \mathbb{X}$, with $\mathbb{X} \subseteq \mathbb{R}_+^M$, can be expressed as follows:

- If $x_1 \neq x_2$ (i.e., the two machines have different capacity),

$$\hat{f}^{LF}(\mathbf{x}) = \begin{cases} \frac{x_2\eta_2K_1e^{\theta_1N_1}+x_1\eta_1K_1e^{\theta_2N_1}+x_1\eta_1K_1e^{-\theta_2N_1}}{K_1e^{\theta_1N_1}+K_1e^{\theta_2N_1}+K_1e^{-\theta_2N_1}}, \\ \qquad \text{if } x_1 < x_2, \\ \frac{x_1e_1K_1e^{-\theta_1N_1}+x_2\eta_1K_2e^{\theta_2N_1}+x_2\eta_2K_1e^{-\theta_2N_1}}{K_1e^{-\theta_1N_1}+K_1e^{\theta_2N_1}+K_1e^{-\theta_2N_1}}, \\ \qquad \text{if } x_1 > x_2, \end{cases} \tag{4}$$

where

$$\begin{aligned}\eta_1 &= \frac{R_1}{P_1 + R_1}, \eta_2 = \frac{R_2}{P_2 + R_2}, \\ Q &= \sqrt{[\frac{x_1(1+\eta_2)}{R_2} - \frac{x_2(1+\eta_1)}{R_1}]^2 + 4x_1x_2P_1P_2}, \\ \varsigma_1 &= \frac{x_1^2R_1(R_1+R_2+P_2) + x_2^2R_2(R_1+R_2+P_1)}{2x_1x_2(x_1-x_2)(R_1+R_2)} \\ & -\frac{(R_1+R_2+P_1+P_2)}{2(x_1-x_2)} - \frac{(R_2P_1+P_2R_1)}{2(x_1-x_2)(R_1+R_2)}, \\ \varsigma_2 &= \frac{(x_1R_1+x_2R_2)Q}{2x_1x_2(x_2-x_1)(R_1+R_2)}, \\ K_1 &= [x_1(R_1+R_2+P_2) - x_2(R_1+R_2+P_1)]R_1Q \\ & \qquad + R_1Q^2, \\ K_2 &= \begin{cases} x_2R_2P_1[(x_1-x_2)(R_1-R_2) - (x_2P_1+x_1P_2) \\ \qquad -Q], \text{ if } x_1 < x_2, \\ x_1R_1P_2[(x_1-x_2)(R_1-R_2) - (x_2P_1+x_1P_2) \\ \qquad -Q], \text{ if } x_1 > x_2, \end{cases} \\ K_3 &= \begin{cases} \frac{\eta_2(x_2-x_1\eta_1)K_1+x_1\eta_1(1-\eta_2)K_2}{x_1\eta_1(\eta_2-1)}, \text{ if } x_1 < x_2, \\ \frac{\eta_1(x_1-x_2\eta_2)K_1+x_2\eta_2(1-\eta_1)K_2}{x_2\eta_2(\eta_1-1)}, \text{ if } x_1 > x_2. \end{cases}\end{aligned} \tag{5}$$

- If $x_1 = x_2$,

$$\hat{f}^{LF}(\mathbf{x}) = \begin{cases} \frac{x_1R_1R_2[P_1(R_2+P_2)-P_2(R_1+P_1)e^{-\varsigma_3N_1}]}{(P_1+P_2)(R_1+R_2)(R_2P_1-P_2R_1e^{-\varsigma_3N_1})}, \\ \qquad \text{if } \frac{P_1}{R_1} \neq \frac{P_2}{R_2}, \\ \frac{x_1^2R_2^2(R_1+R_2)+x_1R_1R_2N_1(P_2+R_2)^2}{[x_1(R_1+R_2)R_1N_1(P_2+R_2)](P_2+R_2)^2}, \\ \qquad \text{if } \frac{P_1}{R_1} \neq \frac{P_2}{R_2}, \end{cases} \tag{6}$$

where

$$\varsigma_3 = \frac{(P_1+P_2+R_1+R_2)(P_1R_2-P_2R_1)}{x_1(P_1+P_2)(R_1+R_2)}. \tag{7}$$

For a production line with more than two machines, there is no closed-form expression for the production rate. Therefore, we use the aggregation-based recursive method to obtain the production performance of the serial production lines. To facilitate the presentation of the procedure, we introduce new notations $x_i^b(u)$ and $x_i^f(u)$, representing the capacity in the u^{th} backward and forward aggregation iteration of machine i, respectively. The detailed aggregation procedure is shown as follows:

Procedure 1.

$$\begin{aligned}x_i^b(u+1) =& \frac{x_i}{\eta_i x_i^f(u)} \hat{f}^{LF}(x_i^f(u), x_{i+1}^b(u+1)| \\ & P_i, R_i, P_{i+1}, R_{i+1}, N_i), i = 1, \dots, M-1, \\ x_i^f(u+1) =& \frac{x_i}{\eta_i x_i^b(u+1)} \hat{f}^{LF}(x_{i-1}^f(u+1), x_i^b(u+1)| \\ & P_{i-1}, R_{i-1}, R_i, P_i, N_{i-1}), i = 2, \dots, M,\end{aligned} \tag{8}$$

Fixing the initial conditions for the procedure to $x_i^f(0) = x_i, i = 2, \dots, M-1$, with boundary conditions being $x_1^f(u) =$

$x_1, x_M^b(u) = x_M, u = 0, 1, \ldots$, then x_i^f is known to converge to a unique solution (result presented in [12], Theorem 11.3, Page 350).

$$x_i^f := \lim_{u \to \infty} x_i^f(u),\ x_i^b := \lim_{u \to \infty} x_i^b(u). \tag{9}$$

And, the production rate becomes:

$$\hat{f}^{LF}(\mathbf{x}) = x_1^b e_1 = x_M^f e_M \tag{10}$$

Equation (8) shows the updated aggregate capacity for machine i at iteration $u+1$ using the $\hat{f}^{LF}$ estimator obtained from Equations (1)-(7) with the updated parameters at iteration u. The initial parameters at iteration $u = 0$ are the same as the capacity value for each machine. Since the first machine and the last machine are the initial machines for the forward and backward aggregation, respectively, the capacity of these two machines remains unchanged throughout the procedure. The convergence of the integration shown in Equation (9) provides the stopping criteria for the iterative procedure. Finally, the production rate can be obtained using the aggregate capacity of all the machines upon convergence of the x_M^f from the forward aggregation, or the x_1^b from the backward aggregation, times the efficiency of the corresponding machines, which is illustrated in Equation (10).

2) Simulation Models: Considering the characteristics (1)-(8), and the related discrete event simulation (DES) in Figure 2, the fidelity of the output resulting from the execution of the model will be controlled by means of two approaches: (1) *simulation parameters driven approach*; (2) *graph based approach*.

a) Simulation parameters driven approach: The first approach is grounded in the simulation output analysis literature [47], [48], [49], and returns the output generated from the simulation of a low number of jobs. As a result, the output from the simulation model will be substantially impacted by initialization bias when compared with the high fidelity model. However, given that the system is initialized under the same condition across all the replications and considering the short simulation length, the variance of the output estimator is negligible, to the point that we will consider a deterministic simulation response. Although the underlying system logic is unaffected, the truncated simulation runs lead to a large initialization bias, thus methods aimed at recovering this model bias should be effective. We executed a preliminary testing and analysis of long and short run (high and low fidelity) simulations, and found a positive correlation with varying magnitudes between the models.

b) Graph reduction approach: The second approach takes as input the graph in Figure 2 and reduces it in order to decrease the simulation time. Let E_k^ξ and e_k^ξ denote the events occurring in the system and their occurrence times, respectively, where $\xi \in \mathbb{T}$ is the event type (i.e., $a, s, f_1, f_2, c, l, u, d$ as in Figure 2), and k indicates that the event belongs to machine $k = 1, 2, \ldots, M$. Let $\mathbb{W} = \{E_k^\xi | \xi \in \mathbb{T}, k \in \{1, \ldots, M\}\}$ be the set of all the events. For each event E_k^ξ, let $\mathbb{I}\left(E_k^\xi\right)$ be the set of input events for E_k^ξ, and let $\mathbb{O}\left(E_k^\xi\right)$ be the set of output events for E_k^ξ. We can define the set of arcs in the model as:

$$\mathbb{E} = \left\{\left(E_k^\xi, E_j^{\xi'}\right) : E_j^{\xi'} \in \mathbb{O}\left(E_k^\xi\right)\right\}_{k=1,\ldots,M;\xi\in\mathbb{T}} \cup \left\{\left(E_j^{\xi'}, E_k^\xi\right) : E_j^{\xi'} \in \mathbb{I}\left(E_k^\xi\right)\right\}_{k=1,\ldots,M;\xi\in\mathbb{T}}.$$

Finally, for each arc, we can define a weight (delay) and a condition forming the pair $\left(w_{k,j}^{\xi,\xi'}, \mathcal{C}_{k,j}^{\xi,\xi'}\right)$, thus generating the label set $\mathbb{L}$ by considering all arcs in $\mathbb{E}$. In this work, we define two operators to reduce the ERGs:

- $\mathcal{A}_1$: Aggregate failure event types to form a single event. Formally, the event set after aggregation can be expressed as:

$$\begin{aligned}\mathbb{W}_{\mathcal{A}_1} =& \mathbb{W} \setminus \{E_k^\xi | \xi \in \{f_1, f_2\}, k \in \{1, \ldots, M\}\} \cup \\ & \{E_k^{\bar{f}} | k \in \{1, \ldots, M\}\}, \\ \mathbb{E}_{\mathcal{A}_1} =& \mathbb{E} \setminus \mathbb{E}_{\text{Elim}} \cup \mathbb{E}_{\text{New}}, \\ \mathbb{L}_{\mathcal{A}_1} =& \mathbb{L} \setminus \mathbb{L}_{\text{Elim}} \cup \mathbb{L}_{\text{New}},\end{aligned}$$

$$\begin{aligned}\mathbb{E}_{\text{Elim}} = & \left\{\left(E_k^{f_j}, E_j^{\xi'}\right) : E_j^{\xi'} \in \mathbb{O}\left(E_k^{f_j}\right), j = 1, 2\right\} \cup \\ & \left\{\left(E_j^{\xi'}, E_k^{f_j}\right) : E_j^{\xi'} \in \mathbb{I}\left(E_k^{f_j}\right), j = 1, 2\right\}, \\ \mathbb{E}_{\text{New}} = & \left\{\left(E_k^{\bar{f}}, E_j^{\xi'}\right) : E_j^{\xi'} \in \mathbb{O}\left(E_k^{\bar{f}}\right)\right\} \cup \\ & \left\{\left(E_j^{\xi'}, E_k^{\bar{f}}\right) : E_j^{\xi'} \in \mathbb{I}\left(E_k^{\bar{f}}\right)\right\}.\end{aligned}$$

In the definitions, $E_k^{\bar{f}}$ is the aggregated failure event in Figure 4. The two failure events, $E_k^{f_1}, E_k^{f_2}$, are aggregated for each server to form the aggregated node $E^{\bar{f}_k}$. In this way, the total number of nodes, and executions, will be reduced by k events. As a result, the arcs are updated eliminating the $4k$ connections reducing the size of the set $\mathbb{E}_{\text{New}}$ to $2k$. These arcs need to have weights, representing the failure time, the condition for failure, and the repair time, recomputed. In particular, the set of new weights $\mathbb{L}_{\text{New}}$ is derived by changing the distribution used to generate the time to the next failure $\tau_k^{f_j}, j = 1, 2,\ k = 1, \ldots, M$. To do so, we use the parameters obtained in the analytical model, and set $\tau^{\bar{f}_k} \sim \text{expo}(\pi_k)$ for each server k, where π_k is obtained through Equation (2).

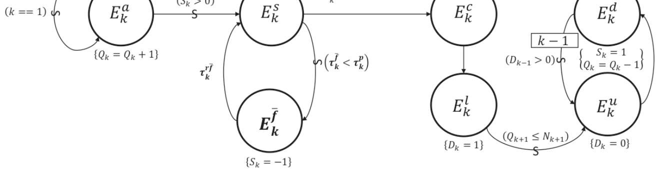


Fig. 4. ERG model for operator $\mathcal{A}_1$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.3025143, IEEE Transactions on Automatic Control

- $\mathcal{A}_2$: Aggregate all the events related to all the machines except the first and the last one in the system into one event. Formally, the graph after aggregation can be expressed as:

$$\begin{aligned}\mathbb{W}_{\mathcal{A}_2} =& \mathbb{W} \setminus \{E_k^{\xi} | \xi \in \mathbb{T}, m_k \in \mathbb{Q}\} \cup \{E_{\Phi}^{\xi}\},\\ \mathbb{E}_{\mathcal{A}_2} =& \mathbb{E} \setminus \mathbb{E}_{\text{Elim}} \cup \mathbb{E}_{\text{New}},\\ \mathbb{L}_{\mathcal{A}_2} =& \mathbb{L} \setminus \mathbb{L}_{\text{Elim}} \cup \mathbb{L}_{\text{New}},\end{aligned}$$

$$\begin{aligned}\mathbb{E}_{\text{Elim}} = & \left\{ \left(E_k^{\xi}, E_k^{\xi'} \right) : E_k^{\xi},\ E_k^{\xi'} \in \mathbb{O}\left(E_k^{\xi} \right) \right\} \cup \\ & \left\{ \left(E_k^{\xi'}, E_k^{\xi} \right) : E_k^{\xi'},\ E_k^{\xi} \in \mathbb{I}\left(E_k^{\xi} \right) \right\}, k \in \mathbb{Q},\\ \mathbb{E}_{\text{new}} = & \left\{ \left(E_{\mathcal{A}_2}^{\xi}, E_{\mathcal{A}_2}^{\xi'} \right) : E_{\mathcal{A}_2}^{\xi},\ E_{\mathcal{A}_2}^{\xi'} \in \mathbb{O}\left(E_{\mathcal{A}_2}^{\xi} \right) \right\} \cup \\ & \left\{ \left(E_{\mathcal{A}_2}^{\xi'}, E_{\mathcal{A}_2}^{\xi} \right) : E_{\mathcal{A}_2}^{\xi'},\ E_{\mathcal{A}_2}^{\xi} \in \mathbb{I}\left(E_{\mathcal{A}_2}^{\xi} \right) \right\}, \xi, \xi' \in \mathbb{T}',\\ & \mathbb{T}' = \{a, s, c, l, u, d\}.\end{aligned}$$

Notice that, with such an approach, all the events associated with machines in $\mathbb{Q}$, are eliminated. For the production system analyzed in this work, we aggregated all the events of machine in $\mathbb{Q} = \{2, \dots, M-1\}$. We use a single virtual unit to mimic the part flow through the machines in $\mathbb{Q}$ of the physical system, with events denoted as $E_{\mathcal{A}}^{\xi}$. Figure 5 shows the resulting model. Furthermore, comparing with the event types set $\mathbb{T}$ in the high fidelity model, the aggregated virtual unit contains no failure events, with the set $\mathbb{T}' = \{a, s, c, l, u, d\}$. In this model, the processing time $\tau_{\mathcal{A}}^{p}$ is assumed to be random. In particular, we consider the lower bound of the processing time as the sum of the unit processing time of the individual machines; subsequently, we model the processing time $\tau_{\mathcal{A}}^{p}$ as a lognormal distribution to scale up the total processing time of the aggregated machine based on the aforementioned lower bound, which can be expressed as follows:

$$\tau_{\mathcal{A}}^{p} \sim (\sum_{k=2}^{M-1} \tau_k^p) \times (1 + LogN(0,1)),$$

where $LogN(0,1)$ is the log normal distribution with parameters $(0,1)$. The events of the other machines in the system remain unchanged. It can be found that, compared with $\mathcal{A}_1$, this operator has a net reduction of all the events of $(M-3)$ machines, which is far less than the total number of events resulting from the use of operator $\mathcal{A}_1$.

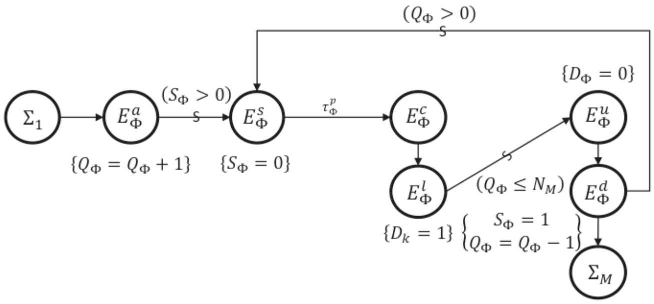


Fig. 5. ERG model for operator $\mathcal{A}_2$

B. Multi-Fidelity Statistical Models

The analytical model as well as the low fidelity simulation models, presented in Section IV-A, can be used in combination with the high fidelity simulation model to produce estimations of the performance response across the solution space. In particular, our strategy is to produce a prediction for the high fidelity model that relies on the bias estimation of each low fidelity model throughout the solution space. To produce the bias prediction, we assume that the bias of each low fidelity model is adequately modeled as a realization of a Gaussian process. Formally, let us refer to $B_i^{LF}(\mathbf{x})$ as the bias of the i-th low fidelity model. Subsequently, we investigate several ways to use the bias information. In particular, let us refer to $f_i^{LF}(\mathbf{x})$ as the response from the i^{th} low fidelity model of solution $\mathbf{x} \in \mathbb{X}$, and let us assume, as justified previously, that $f_i^{LF}(\mathbf{x})$ can be evaluated with no noise. Also, $f^{HF}(\mathbf{x})$, representing the response from the very long simulation run, can be assumed as deterministic. As a result, every point estimate of the bias is noiseless. We propose two different model types for the prediction of the high fidelity response that make different use of the low fidelity information:

- $\mathcal{M}_i$ uses the results from the i-th low fidelity and the high fidelity simulation to derive the bias prediction model $\hat{B}_i^{LF}(\mathbf{x}) | \mathbf{X}, \boldsymbol{f}$, where $\mathbf{X}, \boldsymbol{f}$ represent the sampled locations and the corresponding high fidelity simulation value. The resulting model to predict the response of unsampled points is $\hat{f}^{HF}(\mathbf{x}) = \hat{f}_i^{LF}(\mathbf{x}) + \hat{B}_i^{LF}(\mathbf{x})$;
- $\mathcal{M}_g$ considers the results of multiple types of low fidelity models. Let's assume that n low fidelity models are generated from the analytical and simulation models, with predicted response $\hat{f}_i^{LF}(\mathbf{x})|\mathbf{X}$ and bias $\hat{B}_i^{LF}(\mathbf{x})|\mathbf{X}, \boldsymbol{f}$ for model i, where $\mathbf{X}$ represents the sampled locations, $\boldsymbol{f}$ represents the corresponding high fidelity simulation value and $i \in \{1, \dots, n\}$. We further consider the control variate $\bar{f}_i^{LF} = \hat{f}_i^{LF}(\mathbf{x}) + \hat{B}_i^{LF}(\mathbf{x})$, and the corresponding weight $\beta_i(\mathbf{x})$ for model i. Therefore, we can weight all the n models by the MSE-optimal coefficient $\boldsymbol{\beta}^*(\mathbf{x}) = \{\beta_1, \dots, \beta_n\}$, for the estimator on the sampled points:

$$\begin{aligned}\hat{f}^{HF}(\mathbf{x}) = & \qquad \bar{f}^{HF}(\mathbf{x}) + \\ & \boldsymbol{\beta}^*(\mathbf{x}) \begin{bmatrix} \bar{f}_1^{LF}(\mathbf{x}) - E\left[\bar{f}_1^{LF}(\mathbf{x})\right] \\ \vdots \qquad\qquad \vdots \\ \bar{f}_n^{LF}(\mathbf{x}) - E\left[\bar{f}_n^{LF}(\mathbf{x})\right] \end{bmatrix}, \quad (11)\end{aligned}$$

  where $\bar{f}^{HF}$ represents the expensive high fidelity estimator obtained with the simulation model. Assuming Gaussian processes for the responses, we can have an analytical form for the MSE, allowing the computation of the optimal coefficient $\boldsymbol{\beta}$ throughout the solution space.

It is apparent that we will have a single statistical model of type $\mathcal{M}_g$, and as many statistical models of the type $\mathcal{M}_i$ as low fidelity models we constructed or were given.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.3025143, IEEE Transactions on Automatic Control

C. Multi-Fidelity Optimization

Our novel approach to optimization in the context of multi-fidelity models extends standard Bayesian Optimization (BO) to integrate multiple information sources when making sampling decisions over the costly high fidelity models. In particular, we present two approaches: (1) the *model driven* multi-fidelity optimization (MD-MFO); and (2) the *sampling driven* multi-fidelity optimization (SD-MFO). Figure 6 depicts our multi-fidelity approach within the context of a standard surrogate based simulation optimization (SSO). At the k^{th} iteration, SSO fits a surrogate prediction model from observed input/output data (Figure 6, top). Over this surrogate model, a "sampling criteria" or acquisition function $a(\mathbf{x})$ can be constructed and maximized to determine the next location to sample $\mathbf{x}_{k+1}$ (Figure 6, center). The location $\mathbf{x}_{k+1}$ is then sampled (Figure 6, bottom) and the SSO framework proceeds refitting the surrogate prediction model. Figure 6 highlights the additional low fidelity and bias prediction surrogate models that are fit in our multi-fidelity context.

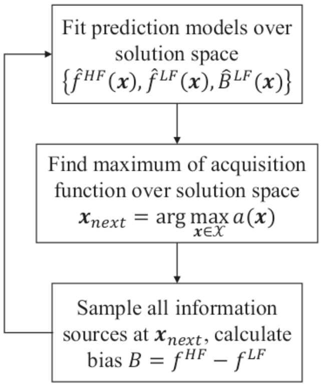

Fit prediction models over solution space
$\{\hat{f}^{HF}(\boldsymbol{x}), \hat{f}^{LF}(\boldsymbol{x}), \hat{B}^{LF}(\boldsymbol{x})\}$
Find maximum of acquisition function over solution space
$\boldsymbol{x}_{next} = \arg\max_{\boldsymbol{x}\in\mathcal{X}} a(\boldsymbol{x})$
Sample all information sources at $\boldsymbol{x}_{next}$, calculate bias $B = f^{HF} - f^{LF}$

Fig. 6. General Bayesian optimization approach in the multi-fidelity context.

1) Model Driven Multi-Fidelity Optimization MD-MFO: The basic idea of MD-MFO is to use the high fidelity predictions generated from the low fidelity models from Section IV-B, $\{\mathcal{M}_i\}, i = 1, \ldots, n$, to feed an acquisition function/sampling criteria. The sampling location at the k^{th} $\mathbf{x}_k$ iteration is generated using the model $\mathcal{M}_g$ to evaluate the sampling criterion. Equivalently, the MD-MFO approach uses $\mathcal{M}_g$ as the surrogate model handed from the top to the center box of Figure 6. In fact, equation (11) shows how $\mathcal{M}_g$ is constructed as a mixture of the high fidelity model, $\bar{\theta}^{HF}$, and *all* of the low fidelity models, $\hat{\theta}_i^{LF}$. The weight of the i^{th} low fidelity mixture component is assigned by the observed relationship between the i_{th} low fidelity and high fidelity information source, which is modeled by the parameter $\beta_i^*(\mathbf{x})$.

Regarding the overall MD-MFO approach, we use the *expected improvement* acquisition function in (12) as the sampling criteria over the mixture model $\mathcal{M}_g$ [50].

$$a(\mathbf{x}) = EI(\mathbf{x}) = \max\left\{\left[\Delta_{HF}(\mathbf{x})\Phi\left(\frac{\Delta_{HF}(\mathbf{x})}{s(\mathbf{x})}\right)\right.\right. \tag{12}$$
$$\left.\left. +s^2(\mathbf{x})\phi\left(\frac{\Delta_{HF}(\mathbf{x})}{s(\mathbf{x})}\right)\right], 0\right\}$$

where $s(\mathbf{x}) = \sqrt{\text{Var}(\hat{f}^{HF}(\mathbf{x}))}$, $\Delta_{HF}(\mathbf{x}) = f^{HF}(\mathbf{x}_*) - \hat{f}^{HF}(\mathbf{x})$, and $\mathbf{x}_*$ is the best observed high fidelity sample and $\hat{f}^{HF}$ is the prediction produced by model $\mathcal{M}_g$. Choosing $\mathbf{x}_{k+1} \in \arg\max_{\mathbf{x}\in\mathbb{X}} EI_k(\mathbf{x})$ yields a sequence of $\{\mathbf{x}_k, k = 1, 2, \ldots\}$, where h denotes expected improvement iteration, that is influenced by *all* of the low fidelity sources. For MD-MFO, we can view the influence of the low fidelity information as being exerted during the model building step of calculating $\hat{f}^{HF}$ from $\mathcal{M}_g$.

2) Sampling Driven Multi-Fidelity Optimization SD-MFO: An alternative to the model driven approach is to focus on the sampling function $a(\mathbf{x})$ (Figure 6, center box). Specifically, SD-MFO aims to combine *several* surrogate prediction models, each embedding low fidelity and bias information. As a result, several surrogate models are passed from the top box to the center box in Figure 6. Let us assume that n Gaussian processes are used to produce predictions $\hat{f}_i(\mathbf{x})$ across $\mathbf{x} \in \mathbb{X}$ for $i = 1, 2, \ldots, n$. Given that the surrogate model forms selected are Gaussian processes, we know that conditional predictions at a given location $\hat{f}_i(\mathbf{x})|\mathbf{x} \in \mathbb{X}$ are normally distributed. Moreover, if we assume that these n processes are independent, then the i resulting conditional predictions can easily be combined under the idea of assigning a single unique score to each point $\mathbf{x}$ in the solution space. Assume that, at any given location, an acquisition function, or sampling criteria, can yield a sampling score $I_i(\mathbf{x})$ that is well defined when only one model is considered. It is possible to design a random function $G\left(I_1\left(\mathbf{x}\right), \ldots, I_n\left(\mathbf{x}\right)\right)$ and an associated density F_G. These two ingredients define the novel concept of *joint score functions*. In the case where $I_i\left(\mathbf{x}\right)$ is the improvement function, i.e., $I_i\left(\mathbf{x}\right) = \max\left(f^* - \hat{f}_i\left(\mathbf{x}\right), 0\right)$, this idea leads to the novel *Joint Expected Improvement (JEI)* defined as:

$$\begin{aligned} JEI\left(\mathbf{x}\right) = & \quad \max\left(E_G\left[G\left(I_1\left(\mathbf{x}\right), \ldots, I_n\left(\mathbf{x}\right)\right)\right], 0\right) \\ = & \quad \max\left(\int_0^\infty g\left(\mathbf{x}\right) \cdot \mathrm{d}F_G(g), 0\right). \end{aligned} \tag{13}$$

And the point is selected which satisfies:

$$\mathbf{x}_k \in \arg\max_{\mathbf{x}\notin\mathbb{S}} \left(JEI\left(\mathbf{x}\right)\right). \tag{14}$$

Nevertheless, how to derive G and its distribution is all but trivial. In this manuscript, we propose two competing strategies and provide the underlying motivations. Specifically, we introduce: (a) the *average joint expected improvement* (aJEI), and (b) the *consensus joint expected improvement* (cJEI).

a) aJEI: A first way to embed the predictions generated by the n low fidelity models, is to consider the point that maximizes the Expected Average Improvement. This corresponds to the following G random function:

$$G\left(\mathbf{x}\right) \quad = \tfrac{1}{n}\sum_{i=1}^n f^* - \hat{F}_i\left(\mathbf{x}\right). \tag{15}$$

Conditional upon the location $\mathbf{x}$, $G_{\mathbf{x}}$ is a normal random variable obtained as the scaled sum of independent normal random variables, centered around f^*, i.e., $G_{\mathbf{x}} \sim \mathcal{N}\left(\mu_{G_{\mathbf{x}}}, \sigma^2_{G_{\mathbf{x}}}\right)$, where:

$$\mu_{G_{\mathbf{x}}} = \frac{1}{n}\sum_{i=1}^{n}\left(f^* - \mu_i\left(\mathbf{x}\right)\right), \tag{16}$$

$$\sigma^2_{G_{\mathbf{x}}} = \frac{1}{n^2}\sum_{i=1}^{n}\sigma_i^2\left(\mathbf{x}\right). \tag{17}$$

At this point, it can be observed that we can simply apply equation (13) to derive aJEI for all locations $\mathbf{x}$ in the feasible space $\mathbb{X}$.

Theorem 1. *(Variance Reduction under aJEI) Consider a number of predictive models $n = 2$, without loss of generality, express $\sigma_2 = \Psi\sigma_1$. In this setting, a reduction in posterior variance is achieved if and only if:*

$$\frac{\sqrt{3}}{3} \leq \Psi \leq \sqrt{3}. \tag{18}$$

Proof. Consider the following:

$$\begin{aligned} &\frac{\sigma_1^2+\sigma_2^2}{4} \leq \sigma_1^2 \quad \& \quad \frac{\sigma_1^2+\sigma_2^2}{4} \leq \sigma_2^2 \\ \iff &\frac{1+\Psi^2}{4} \leq 1 \quad \& \quad \frac{1+\Psi^2}{4} \leq \Psi^2 \\ \iff &\frac{\sqrt{3}}{3} \leq \Psi \leq \sqrt{3}. \end{aligned} \tag{19}$$

Thus when the distributions have similar variances this results in a higher overall confidence in the distribution. However, if they vary substantially, then the distribution with lower accuracy tends to dilute the accuracy of the other. □

b) cJEI: The second approach we take to combine the n predictive models biases the sampling towards locations that maximize the "agreement" among different predictors, we refer to this second approach as *consensus Joint Expected Improvement*. We consider the random improvement at location $\mathbf{x}$ of the i^{th} random model with known distribution $F_{I_i(\mathbf{x})}\left(y\right)$, where y represents the realization of the improvement. Let us consider the probability ρ that all the models achieve an improvement at most y. Under the independence assumption, we obtain:

$$\rho = \prod_{i=1}^{n} F_{I_i(\mathbf{x})}\left(y\right) \tag{20}$$

With a reasoning similar to the derivation of Bayesian posteriors, the distribution of G, conditional upon the point $\mathbf{x}$, will satisfy:

$$F_{G_{\mathbf{x}}} \propto \rho, \tag{21}$$

where the proportionality constant is known or can be estimated as $\frac{1}{2\pi\sqrt[2]{\prod_{i=1}^{n}\sigma_i^2}}$. Ignoring such a constant, $G_{\mathbf{x}} \sim \mathcal{N}\left(\mu_{G_{\mathbf{x}}}, \sigma^2_{G_{\mathbf{x}}}\right)$, where:

$$\mu_{G_{\mathbf{x}}} = \frac{\sum_{i=1}^{n}\mu_{\hat{F}_i}/\sigma^2_{\hat{F}_i}}{\sum_{i=1}^{n}1/\sigma^2_{\hat{F}_i}}, \tag{22}$$

$$\sigma^2_{G_{\mathbf{x}}} = \sqrt{\frac{1}{\sum_{i=1}^{n}1/\sigma^2_{\hat{F}_i}}}. \tag{23}$$

V. Numerical Study

In this section, we present the numerical analysis of the proposed approach. In particular, we first, separately and empirically, validate the modeling and optimization methods. Finally, we propose a case study as a proof of concept of the overall approach.

A. Statistical Modeling Validation

A first test is carried out over two dimensions on a box-constrained domain $(x_1, x_2) \in [0.5, 1.5] \times [0.5, 1.5]$. The high fidelity model is a Gaussian Process, $GP(\mu, \Sigma)$, with $\mu = [0.5, 0.5]^T, \Sigma = [0.5, 0; 0, 0.5]$. The prediction resulting from 3000 randomly sampled points is shown in Figure 7(a). As synthetic low fidelity models, we use two Gaussian processes, one with positive bias $GP(1.5\mu, \Sigma)$ and one with negative bias $GP(-0.5\mu, \Sigma)$. The bias process associated to the low fidelity models is referred to as B_1^{LF} and B_2^{LF}, respectively.

In order to test the quality of our multi-fidelity model, we estimate $\mathcal{M}_1$ and $\mathcal{M}_2$, and $\mathcal{M}_g$ sampling 50 locations at random in the solution space.

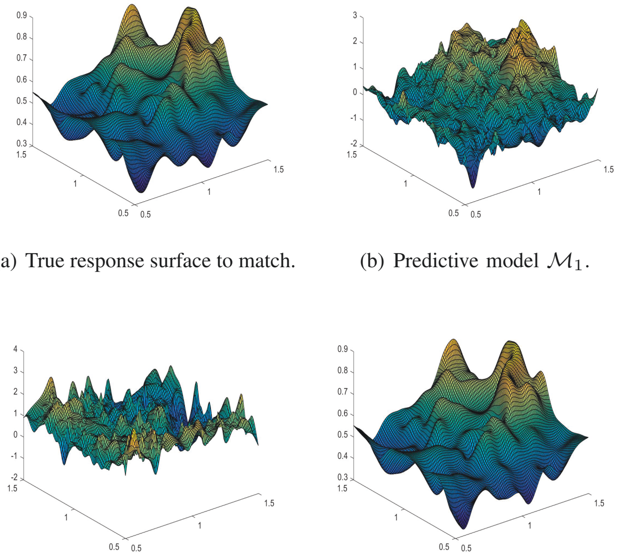
The plot of the true response surface and the corresponding predictions for our proposed modeling methods are reported in Figure 7. In Figure 7(a)-7(d), the horizontal axes refer to the location coordinate, and the vertical axis represents the prediction produced by the Gaussian process model. In Figure 7 it can be observed that the bias-adjusted models $\mathcal{M}_i, i = 1, 2$ (Figures 7(b)-7(c), respectively) reflect the behavior of the high fidelity model. For example, the positions of the peaks and the valleys of the prediction in Figure 7(b) match the true response in Figure 7(a). For the predictions considering multiple models, Figure 7(d), the prediction is very close to the true response surface.

We test the predictive capabilities of our three proposed models, $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_g$, against a baseline approach which only makes use of high fidelity information and ignores all low fidelity models. This baseline, denoted as $\mathcal{M}_{HF}$, considers the same 50 sample points from the true high fidelity response surface used for the models $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_g$, and builds a predictive Gaussian process. In fact, the models $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_g$, augment the same 50 high fidelity response samples with the low fidelity response values at those locations. We compare the effectiveness of our modeling method in improving predictive capabilities when high fidelity information is augmented with low fidelity information.

Specifically, we use the following performance metric to capture the error measures of each model:

$$\delta_{model} = \frac{|\widehat{f}^{HF} - f^{HF}|}{f^{HF}} \times 100\%, \tag{24}$$

where $\hat{f}^{HF}$ denotes the prediction obtained from the one of the models ($\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_g$, or $\mathcal{M}_{HF}$) and f^{HF} denotes the true function value. Such a measure shows the percentage deviation of the prediction model compared with the corresponding true function values. The mean, standard deviation, and maximum of δ_{model} are reported in Table I. It can observed from the plot and the measures that the mixture model, $\mathcal{M}_g$ performs the



(a) True response surface to match. (b) Predictive model $\mathcal{M}_1$.

(c) Predictive model $\mathcal{M}_2$. (d) Predictive model $\mathcal{M}_g$.

Fig. 7. Plot of the multi-fidelity models in the theoretical examples

best among all the different types of prediction models including the baseline approach of $\mathcal{M}_{HF}$, where only high fidelity samples are considered. $\mathcal{M}_g$ is able to effectively leverage the extra low fidelity response information (in combination with the 50 high fidelity response samples) to produce a highly accurate response surface with only 50 high fidelity samples and each low fidelity model.

TABLE I
THEORETICAL MODEL ACCURACIES (IN %)

Model	Mean δ_{model}	Std Dev δ_{model}	Max δ_{model}
$\mathcal{M}_1$	76	59	399
$\mathcal{M}_2$	114	89	598.5
$\mathcal{M}_g$	$7.04e^{-6}$	$6.51e^{-6}$	$4.26e^{-5}$
$\mathcal{M}_{HF}$	6.7	6.5	42.1

B. Multi-Fidelity Optimization Validation

In the scope of testing the optimization approach, we selected the d-dimensional functions reported in Table II, where the correlation between the high fidelity and each low fidelity model is also reported. Testing was carried out in four dimensions ($d = 4$) over a box-constrained domain from the Cartesian product of intervals $\prod_1^d [0.1, 1]$ with a high fidelity function evaluation budget of $b = 100$, i.e., allowing 100 evaluations of the high fidelity source.

TABLE II
THEORETIC HIGH AND LOW FIDELITY TEST FUNCTIONS.

Model	Functional Form	Correlation
HF Model	$-2.5 \prod_{i=1}^d \sin(\pi x_i) - \prod_{i=1}^d \sin(5\pi x_i)$	
LF Model 1	$-2 \prod_{i=1}^d \sin(\pi x_i)$	0.87
LF Model 2	$-0.8 \prod_{i=1}^d \sin(5\pi x_i)$	0.49
LF Model 3	$2 \prod_{i=1}^d \sin(\pi x_i)$	-0.87
LF Model 4	$0.8 \prod_{i=1}^d \sin(5\pi x_i)$	-0.49

We tested five competing algorithms.

1) ***CV BO***: This *model driven* multi-fidelity optimization (*MD-MFO*) approach adopts a standard expected improvement acquisition function calculated using the $\mathcal{M}_g$ model in equation (11);
2) ***aJEI***: This sampling driven multi-fidelity optimization (*SD-MFO*) is executed jointly over multiple bias adjusted low fidelity models $\mathcal{M}_i$, $i = 1, \ldots, n$, utilizing the aJEI acquisition function to reconcile the predictions;
3) ***cJEI***: This sampling driven multi-fidelity optimization (*SD-MFO*) is executed jointly over multiple bias adjusted low fidelity models $\mathcal{M}_i$, $i = 1, \ldots, n$, utilizing the cJEI acquisition function to reconcile the multiple predictions;
4) ***HF BO***: The EGO algorithm uses expected improvement defined only using the high fidelity source, and ignoring any low fidelity information [50].
5) **misoKG**: The misoKG algorithm takes sampling decisions by trading the value of information resulting from sampling the i^{th} low fidelity source at location $\mathbf{x}$, against the cost of that information, where such cost is provided by the user. The value of information is the expected gain in the quality of the best observation, which is then normalized by the cost of sampling. This is formalized through the use of Gaussian processes to estimate model discrepancies and the maximization of the MKG acquisition function, an extension of the knowledge gradient [35]. We chose misoKG as it was empirically shown to outperform both expected improvement based multi-fidelity in [36] and multi task Bayesian Optimization (MTBO) [37], which can all handle un-ranked low fidelity models.

We macro-replicated each of the five algorithms 30 times and observed the average performance and the associated standard error. In particular, we assess two performance metrics: (1) the average Euclidean distance of the proposed solution from the true minimum $\sum_{j=1}^{30} ||\hat{\mathbf{x}}_j^* - \mathbf{x}^*||_2/30$; (2) the average absolute function value error $\sum_{j=1}^{30} |f(\hat{\mathbf{x}}_j^*) - f(\mathbf{x}^*)|/30$, where $f(\cdot)$ is the high fidelity function evaluation and $\hat{\mathbf{x}}_j^*$ is the identified minimum from the j^{th} macro-replication. The two metrics (with standard error) are reported in Table III for each of the four tested algorithms.

In Figure 8, we report the average distance as a function of the simulation budget, and the mean cumulative regret, defined as: $\sum_1^i \sum_{j=1}^{30} (f(\hat{\mathbf{x}}_{ij}) - f(\mathbf{x}^*))/30$, where i indicates the number of high fidelity samples taken, i.e., $i = 1, \ldots, b$. Note that the benchmark algorithm, *HF BO*, which excludes all low fidelity information, serves as a control experiment to observe how embedding the low fidelity information impacts the algorithms performance. Figures 8(a)-8(b) show the progress against the total number of high fidelity evaluations allowed. Figure 8(a) shows the Euclidean distance of the best observed location relative to the true minimum as the simulation budget is exhausted, and Figure 8(b) shows the average cumulative regret as the cumulated gap between the best observed function value and the true minimum.

Table III and Figure 8(a) show that all algorithms perform well over the theoretical test environment with budget $b = 100$.

TABLE III
OPTIMIZATION ALGORITHM RESULTS AFTER 100 SAMPLES ON 4 DIMENSIONAL THEORETIC PROBLEM.

Algorithm	$\overline{\vert\vert\hat{x}_k^* - x^*\vert\vert}$	Std Error	$\overline{\vert f(\hat{x}_k^*) - f(x^*)\vert}$	Std Error
CV BO	0.0388	0.0105	0.1550	0.0268
aJEI	0.0212	0.0085	0.0853	0.0206
cJEI†	**0.0098**†	**0.0084**	**.0264**†	**0.0161**
HF BO	0.0213	0.0088	0.0893	0.0210
misoKG	0.1680	0.0200	1.0803	0.0987

† statistical best performance at $\alpha_{sig} = 0.01$

From Figures 8(a) and 8(b), we observe that misoKG does not perform well in this test environment. There are two related aspects of misoKG that affect algorithm performance: 1) for each information source ℓ misoKG requires a cost function $c_\ell(\mathbf{x})$ to be defined by a user as input for the algorithms to run, and 2) misoKG proceeds by sampling a single source ℓ at location $\mathbf{x}$ by maximizing the MKG acquisition function, which is defined for each $(\ell, \mathbf{x})$ pair. The MKG acquisition function is defined as:

$$MKG_i(\ell, \mathbf{x}) = \mathbb{E}_i \left[\frac{f_i^* - y_{i+1}^*}{c_\ell(\mathbf{x})} | \ell_{i+1} = \ell, \mathbf{x}_{i+1} = \mathbf{x} \right]$$

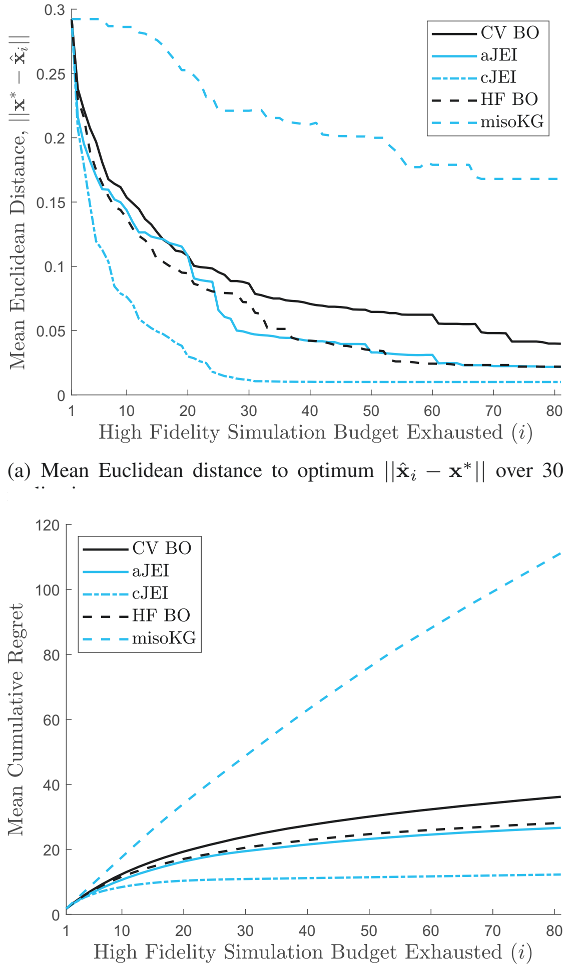
where $f_i^* = \min_{\mathbf{x} \in \mathbb{X}} \hat{f}_i^{HF}(\mathbf{x})$ is the value of the minimum high fidelity prediction at the i^{th} iteration, and $c_\ell(\mathbf{x})$ is a normalizing term for the potential information gain of sampling $(\ell, \mathbf{x})$. Each $(\ell, \mathbf{x})$ sample decision from MKG is a greedy maximization of improvement in high fidelity prediction relative to the sample/source cost. Thus, information sources are viewed as competitors and each iteration samples from the "best" source. Our approach views the sources cooperatively, and makes sampling decisions using all sources collectively. When there are several information sources, as in this case, we observe that misoKG suffers by making sampling decisions based upon the MKG normalized acquisition function, which only considers a single source at a time.

In particular, the advantage of sampling sources simultaneously appears to be confirmed by the performance of the consensus Joint Expected Improvement, which coalesces a sampling decision from all the models simultaneously, and outperforms all other algorithms. Over the 30 macro-replications cJEI accurately identifies the true global minimum of the high fidelity simulator, on average using only 30 high fidelity evaluations. After 10 simulations, the cJEI approach identifies solutions that are substantially closer to the true global minimum, than either of the algorithms that explicitly fit a Gaussian process (GP) to the high fidelity responses. Considering that every GP fit in the cJEI algorithm uses low fidelity information, the performance of cJEI highlights the effectiveness of fitting bias corrected models and how simultaneous consideration of all low fidelity proposals yields better results.

C. Case study

In this part of the analysis, we present the experimental results of the cJEI algorithm, the best performer among the algorithms previously analyzed, against common state-of-the-art global optimization algorithms that are embedded into commercial simulation software and are therefore popular within the manufacturing community: particle swarm optimization (PSO), genetic algorithm (GA), simulated annealing (SA), and pattern search (PS). All these algorithms make use of only high fidelity information. These algorithms are widely used in commercial simulation software such as Rockwell Arena, Simio, Matlab-Simulink. We use the implementations for the aforementioned algorithms available within the *MatLab global optimization toolbox*. Testing is conducted using the high fidelity simulator of the production line described in Section III, along with the analytical and simulation-based low fidelity models presented in Sections IV-A1 and IV-A2 respectively. Due to the complexity of the serial production line, the true global optimum $(\mathbf{x}^*, f(\mathbf{x}^*))$ is not known for any number of machines $M \geq 2$ and for multiple failures as it is the case for us. As a result, the only metric available to judge the competing optimization algorithms is $f(\hat{\mathbf{x}}^*)$, i.e., the best observed objective function value (corresponding to the system throughput).

In order to analyze the performance of the proposed approach we generate random instances for 5-machine production lines. For the given system $M = 5$, we randomly draw the remaining static parameters for the system, $N_k \in \{3, \ldots, 8\}$,



(a) Mean Euclidean distance to optimum $||\hat{\mathbf{x}}_i - \mathbf{x}^*||$ over 30

(b) Mean cumulative regret from optimal value over 30 replications.

Fig. 8. Optimization algorithm progression as simulation budget is exhausted.

TABLE IV
EXPERIMENTAL PARAMETER SETTINGS

Case	α	N	MTTR
1	[0.36,0.22,0.41,0.11; 0.46,0.02,0.40,0.22; 0.37,0.08, 0.26,0.41;0.49,0.17,0.40,0.45; 0.47,0.05,0.21,0.09]	[3,4, 5,5]	[2.83,3.44,2.00, 2.60,2.29]
2	[0.28,0.32,0.36,0.08; 0.35,0.10,0.44,0.44; 0.35,0.43, 0.22,0.26;0.22,0.22,0.23,0.07; 0.38,0.12,0.23,0.12]	[4,4, 6,4]	[2.87,2.05,3.10, 2.87,2.84]
3	[0.33,0.02,0.34,0.33; 0.28,0.35,0.38,0.02; 0.37,0.14, 0.32,0.15;0.41,0.23,0.25,0.28; 0.43,0.16,0.27,0.20]	[8,3, 4,3]	[3.10,3.41,2.58, 3.02,3.79]
4	[0.33,0.40,0.26,0.44; 0.50,0.09,0.38,0.01; 0.32,0.03, 0.49,0.23; 0.48,0.40,0.46,0.10; 0.22,0.31,0.25,0.38]	[4,8, 3,4]	[3.93,3.09,3.95, 3.43,3.40]
5	[0.27,0.05,0.42,0.23; 0.25,0.45,0.28,0.22; 0.29,0.32, 0.37,0.31;0.28,0.15,0.28,0.17; 0.24,0.09,0.49,0.49]	[6,7, 6,4]	[2.44,3.74,2.41, 3.84,2.98]
6	[0.39,0.23,0.42,0.27; 0.37,0.33,0.50,0.42; 0.32,0.45, 0.45,0.04;0.42,0.41,0.42,0.36; 0.36,0.07,0.49,0.21]	[6,6, 5,5]	[3.79,2.66,3.64, 2.08,2.22]
7	[0.35,0.35,0.44,0.20; 0.22,0.15,0.47,0.12; 0.34,0.48, 0.21,0.31;0.49,0.13,0.36,0.46; 0.24,0.27,0.43,0.34]	[6,6, 3,4]	[2.15,3.56,2.88, 3.45,3.96]
8	[0.34,0.29,0.36,0.39; 0.41,0.32,0.33,0.15; 0.49,0.17, 0.27,0.04;0.49,0.07,0.30,0.05; 0.27,0.21,0.47,0.18]	[3,5, 5,6]	[3.75,3.94,3.74, 3.06,2.47]
9	[0.30,0.09,0.46,0.49;0.21,0.36,0.37,0.46;0.40,0.28, 0.41,0.20;0.41,0.42,0.34,0.48;0.44,0.50,0.25,0.27]	[4,5, 4,3]	[2.02,3.00,2.99, 2.27,2.28]
10	[0.23,0.35,0.49,0.01; 0.35,0.42,0.38,0.37;0.29,0.47, 0.41,0.28;0.24,0.20,0.40,0.23; 0.33,0.32,0.35,0.34]	[4,4, 7,4]	[3.54,2.04,3.27, 3.050,3.00]

$R_k \in (0.25, 0.5), \alpha_{j1}^{(k)} \in (0.2, 0.5),\ \alpha_{j2}^{(k)} \in (0, 0.5), j = 1, 2, k = 1, \ldots, 5$. The parameters are generated following the cases discussed in [21], which considers the randomness in the manufacturing systems, such as the failure rate, repairing rate and the transitions among multiple failure modes. The selections of the parameters are also representative to widely cover the application scenarios in the real systems. The resulting testing conditions are reported in Table IV.

We analyze the impact of the total number of simulations in high fidelity required to obtain a satisfactory solution; allowing for a fair comparison of the performance between the proposed cJEI algorithm and the state of the art competitors. In order to do so, we performed experiments over Case 1 in Table IV. We performed 50 macro-replications and the obtained results are reported in Table V for all the algorithms with the statistical significance. Figure 9 pictures the confidence intervals around the optimal estimated function value $\hat{f}_b^*$, where b represents the number of allowed simulations in high-fidelity and it is set to $b = \{10, 20, 30, 50, 75, 150\}$. First, we observe, from Figure 9, that the incremental improvement in performance decreases for increasing simulation budget. In particular, the improvement for cJEI is minor if we go from $b = 50$ to $b = 75$ and even more from 75 to 150. Since simulations are expensive, we wish to use the lowest possible budget. Also, we can observe that the proposed algorithm is never dominated by any of the competitors independently from the budget used.

D. Randomized Testing

In practice, the budget should be set such that it is as small as possible while still providing a high quality solution. Based on the preliminary results in Table V and Figure 9, we chose to further investigate two budget values $b = 30$ and $b = 50$; $b = 30$ is the smallest value that shows good performance, while $b = 50$ is the smallest value that shows solution convergence. In order to explore the performance of the proposed approach over random system configurations, we proceed testing the same algorithms over all the cases in Table IV with the two alternative budget values.

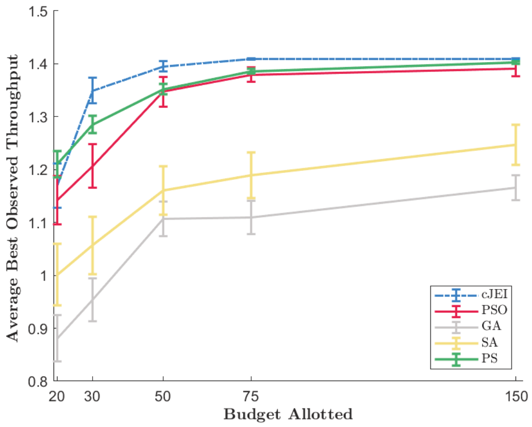


Fig. 9. Mean performance of cJEI against competitors across alternative allotted sampling budget scenarios.

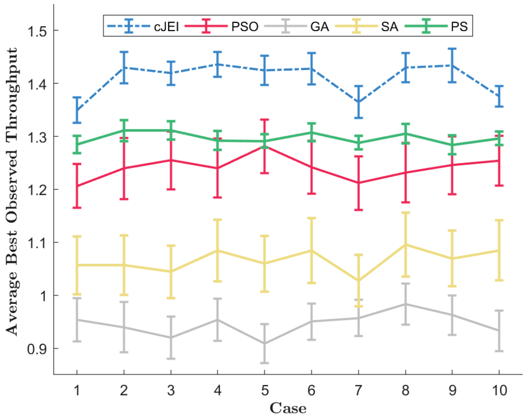


Fig. 10. Mean performance of cJEI against competitors across randomized cases with 30 sampling budget, $b = 30$.

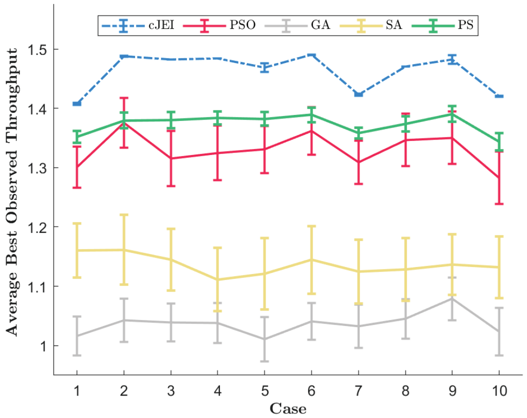


Fig. 11. Mean performance of cJEI against competitors across randomized cases with 50 sampling budget, $b = 50$.

TABLE V
cJEI PERFORMANCE WITH VARYING BUDGET

Algorithm	b	$\hat{f}^*$	
		average	std err
cJEI	10	1.029	0.0275
	20	1.1699	0.0208
	30	**1.3495**†	**0.0121**
	50	**1.3915**†	**0.0051**
	75	**1.4095**†	**0.0006**
	150	**1.4097**†	**0.0006**
PSO	10	1.0023	0.0258
	20	1.1426	0.0229
	30	1.2064	0.0206
	50	1.3472	0.0142
	75	1.3793	0.0069
	150	1.3909	0.0071
GA	10	0.8726	0.0217
	20	0.8807	0.0219
	30	0.9537	0.0205
	50	1.0165	0.0164
	75	1.1091	0.0156
	150	1.1654	0.0116
SA	10	0.8765	0.0249
	20	1.0012	0.029
	30	1.0566	0.0272
	50	1.1601	0.0228
	75	1.1893	0.0215
	150	1.2468	0.0191
PS	10	1.1122	0.0181
	20	1.2099	0.0123
	30	1.285	0.0082
	50	1.3518	0.005
	75	1.3858	0.0024
	150	1.4024	0.0015

† statistically better mean than all State of the Art at $\alpha_{sig} = 0.05$

The obtained results for $b = 30$ are displayed in Table VI and Figure 10, while the results for $b = 50$ are displayed in Table VII and Figure 11. While the optimal solution location is unknown, we notice that, under all the tested cases with both $b = 30$ and $b = 50$, the proposed approach outperforms state of the art algorithms, thus establishing the empirical relevance of the proposed algorithm. The differences between Figure 10 and Figure 11 echo the results seen in Figure 9, with cJEI's confidence intervals collapsing to indicate convergence of the method. While the advantage of cJEI compared to the alternative algorithms (i.e., the performance improvement with respect to the benchmark algorithms) appears to be varying in both budget scenarios across the randomized system conditions, the performance appears to be consistent across the random case selected.

VI. CONCLUSION

In this work, a multi-fidelity optimization approach is designed to estimate and optimize the production performance of a serial production system with asynchronous machines in multi-failure modes. In particular, analytical models and simulation models are used. High fidelity simulation experiments are run with both a large number of replications and large run length. For low fidelity simulation models, the simulation length is shortened, combined with an aggregation approach to further reduce the computational cost. This enables running a large number of evaluations with the low fidelity models, which can be used to predict the high fidelity result with very few high fidelity expensive simulations. Experiment results show that the multi-fidelity model could provide higher accuracy than individual models, and is computationally quicker than the high fidelity simulation model. Moreover, we propose novel methods for utilizing multiple fidelity models to optimize over a high fidelity model and show applications in determining the capacity combination to maximize a system production rate in limited computation time scenarios. Our proposed consensus joint expected improvement method has demonstrated the ability to search a large number of solutions with tight time budget and the results outperform existing solution methods.

Future work can be extended to more complicated systems, such as assembly lines and production networks. For the decision variables, more factors can be included, such as production quantities, which can provide additional information to improve the modeling accuracy for machine's up time. Furthermore, different types of low fidelity models, such as empirical and statistical models, can be further investigated and incorporated in the proposed decision framework.

REFERENCES

[1] Y. Lu and F. Ju, "Smart manufacturing systems based on cyber-physical manufacturing services (cpms)," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15 883–15 889, 2017.

[2] J. R. Stack, T. G. Habetler, and R. G. Harley, "Effects of machine speed on the development and detection of rolling element bearing faults," *IEEE Power Electronics Letters*, vol. 99, no. 1, pp. 19–21, 2003.

[3] L. Hao, K. Liu, N. Gebraeel, and J. Shi, "Controlling the residual life distribution of parallel unit systems through workload adjustment," *IEEE Transactions on Automation Science and Engineering*, 2015.

[4] J. Xu, S. Zhang, E. Huang, C.-H. Chen, L. H. Lee, and N. Celik, "Mo2tos: Multi-fidelity optimization with ordinal transformation and optimal sampling," *Asia-Pacific Journal of Operational Research*, vol. 33, no. 03, p. 1650017, 2016.

[5] N. Celik, S. Lee, K. Vasudevan, and Y.-J. Son, "Dddas-based multi-fidelity simulation framework for supply chain systems," *IIE Transactions*, vol. 42, no. 5, pp. 325–341, 2010.

[6] B. Peherstorfer, K. Willcox, and M. Gunzburger, "Survey of multifidelity methods in uncertainty propagation, inference, and optimization," *SIAM Review*, vol. 60, no. 3, pp. 550–591, 2018.

[7] A. I. Forrester, A. Sóbester, and A. J. Keane, "Multi-fidelity optimization via surrogate modelling," in *Proceedings of the royal society of london a: mathematical, physical and engineering sciences*, vol. 463, no. 2088. The Royal Society, 2007, pp. 3251–3269.

[8] T. Robinson, K. Willcox, M. Eldred, and R. Haimes, "Multifidelity optimization for variable-complexity design," in *11th AIAA/ISSMO multidisciplinary analysis and optimization conference*, 2006, p. 7114.

[9] H. Papadopolous, C. Heavey, and J. Browne, *Queueing theory in manufacturing systems analysis and design*. Springer Science & Business Media, 1993.

[10] J. A. Buzacott and J. G. Shanthikumar, *Stochastic models of manufacturing systems*. Prentice Hall Englewood Cliffs, NJ, 1993, vol. 4.

[11] S. B. Gershwin, *Manufacturing systems engineering*. Prentice Hall, 1994.

[12] J. Li and S. M. Meerkov, *Production systems engineering*. Springer Science & Business Media, 2008.

[13] S.-Y. Chiang, C.-T. Kuo, J.-T. Lim, and S. Meerkov, "Improvability of assembly systems i: Problem formulation and performance evaluation," *Mathematical Problems in Engineering*, vol. 6, no. 4, pp. 321–357, 2000.

[14] F. Ju, J. Li, G. Xiao, J. Arinez, and W. Deng, "Modeling, analysis, and improvement of integrated productivity and quality system in battery manufacturing," *IIE transactions*, vol. 47, no. 12, pp. 1313–1328, 2015.

[15] S. B. Gershwin and M. H. Burman, "A decomposition method for analyzing inhomogeneous assembly/disassembly systems," *Annals of Operations Research*, vol. 93, no. 1-4, pp. 91–115, 2000.

[16] S. Yang, C. Wu, and S. J. Hu, "Modeling and analysis of multi-stage transfer lines with unreliable machines and finite buffers," *Annals of Operations Research*, vol. 93, no. 1-4, pp. 405–421, 2000.

TABLE VI
COMPARISON OF THE ALGORITHMS UNDER RANDOMIZED CASES WITH $b = 30$

case	cJEI		PSO		GA		SA		PS	
	f bar	std err f	f bar	std err f	f bar	std err f	f bar	std err f	f bar	std err f
1	**1.349**	0.012	1.206	0.021	0.954	0.021	1.057	0.027	1.285	0.008
2	**1.430**	0.015	1.239	0.029	0.940	0.024	1.057	0.028	1.311	0.010
3	**1.419**	0.011	1.255	0.027	0.920	0.020	1.044	0.025	1.311	0.009
4	**1.436**	0.012	1.240	0.028	0.954	0.020	1.084	0.029	1.292	0.009
5	**1.425**	0.014	1.281	0.025	0.909	0.018	1.060	0.026	1.291	0.006
6	**1.428**	0.015	1.242	0.025	0.950	0.017	1.085	0.031	1.308	0.008
7	**1.365**	0.015	1.212	0.025	0.957	0.017	1.027	0.024	1.288	0.006
8	**1.430**	0.014	1.232	0.028	0.983	0.020	1.095	0.030	1.305	0.009
9	**1.434**	0.016	1.245	0.028	0.963	0.019	1.070	0.026	1.284	0.009
10	**1.376**	0.010	1.254	0.024	0.933	0.019	1.085	0.028	1.297	0.007

TABLE VII
COMPARISON OF THE ALGORITHMS UNDER RANDOMIZED CASES WITH $b = 50$

case	cJEI		PSO		GA		SA		PS	
	f bar	std err f	f bar	std err f	f bar	std err f	f bar	std err f	f bar	std err f
1	**1.408**	0.001	1.301	0.017	1.017	0.016	1.160	0.023	1.352	0.005
2	**1.488**	0.001	1.376	0.021	1.043	0.018	1.162	0.029	1.380	0.007
3	**1.482**	0.001	1.315	0.024	1.039	0.016	1.145	0.026	1.381	0.007
4	**1.484**	0.001	1.325	0.023	1.038	0.017	1.112	0.027	1.384	0.005
5	**1.469**	0.003	1.331	0.020	1.011	0.019	1.122	0.030	1.383	0.006
6	**1.490**	0.001	1.362	0.020	1.041	0.015	1.145	0.029	1.389	0.006
7	**1.423**	0.001	1.309	0.018	1.033	0.018	1.125	0.027	1.358	0.005
8	**1.471**	0.0001	1.347	0.022	1.045	0.017	1.128	0.026	1.374	0.006
9	**1.482**	0.004	1.351	0.022	1.079	0.018	1.137	0.026	1.391	0.007
10	**1.420**	0.001	1.283	0.022	1.024	0.020	1.132	0.026	1.344	0.007

[17] T. Tolio, A. Matta, and S. B. Gershwin, "Analysis of two-machine lines with multiple failure modes," *Iie Transactions*, vol. 34, no. 1, pp. 51–62, 2002.

[18] B. R. Sarker and J. A. Fitzsimmons, "The performance of push and pull systems: a simulation and comparative study," *International Journal of Production Research*, vol. 27, no. 10, pp. 1715–1731, 1989.

[19] C. D. Paternina-Arboleda and T. K. Das, "Intelligent dynamic control policies for serial production lines," *IIE Transactions*, vol. 33, no. 1, pp. 65–77, 2001.

[20] H. A. Vergara and D. S. Kim, "A new method for the placement of buffers in serial production lines," *International Journal of Production Research*, vol. 47, no. 16, pp. 4437–4456, 2009.

[21] M. Colledani and S. B. Gershwin, "A decomposition method for approximate evaluation of continuous flow multi-stage lines with general markovian machines," *Annals of Operations Research*, vol. 209, no. 1, pp. 5–40, 2013.

[22] G. Fishman, *Discrete-event simulation: modeling, programming, and analysis*. Springer Science & Business Media, 2013.

[23] L. W. Schruben, "Simulation modeling for analysis," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 20, no. 1, p. 2, 2010.

[24] L. Schruben, "Simulation modeling, experimenting, analysis, and implementation," in *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World*. IEEE Press, 2013, pp. 678–690.

[25] Z. Lin, A. Matta, N. Li, and J. G. Shanthikumar, "Extended kernel regression: a multi-resolution method to combine simulation experiments with analytical methods," in *Proceedings of the 2016 Winter Simulation Conference*. IEEE Press, 2016, pp. 590–601.

[26] L. Hao, K. Liu, N. Gebraeel, and J. Shi, "Controlling the residual life distribution of parallel unit systems through workload adjustment," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 1042–1052, 2017.

[27] H. Li and A. K. Parlikad, "Study of dynamic workload assignment strategies on production performance," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13 710–13 715, 2017.

[28] D. Gyulai, B. Kádár, A. Kovács, and L. Monostori, "Capacity management for assembly systems with dedicated and reconfigurable resources," *CIRP Annals-Manufacturing Technology*, vol. 63, no. 1, pp. 457–460, 2014.

[29] P. Renna, "A decision investment model to design manufacturing systems based on a genetic algorithm and monte-carlo simulation," *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 6, pp. 590–605, 2017.

[30] C.-H. Chen, J. Lin, E. Yücesan, and S. E. Chick, "Simulation budget allocation for further enhancing the efficiency of ordinal optimization," *Discrete Event Dynamic Systems*, vol. 10, no. 3, pp. 251–270, 2000.

[31] Y. Peng, J. Xu, L. H. Lee, J.-Q. Hu, and C.-H. Chen, "Efficient simulation sampling allocation using multi-fidelity models," *IEEE Transactions on Automatic Control*, 2018.

[32] D. Huang, T. T. Allen, W. I. Notz, and N. Zeng, "Global optimization of stochastic black-box systems via sequential kriging meta-models," *Journal of global optimization*, vol. 34, no. 3, pp. 441–466, 2006.

[33] K. Kandasamy, G. Dasarathy, J. B. Oliva, J. Schneider, and B. Póczos, "Gaussian process bandit optimisation with multi-fidelity evaluations," in *Advances in Neural Information Processing Systems*, 2016, pp. 992–1000.

[34] M. C. Kennedy and A. O'Hagan, "Predicting the output from a complex computer code when fast approximations are available," *Biometrika*, vol. 87, no. 1, pp. 1–13, 2000.

[35] M. Poloczek, J. Wang, and P. I. Frazier, "Multi-information source optimization," in *Advances in Neural Information Processing Systems*, 2017, accepted for publication. Code is available at http://github.com/misokg.

[36] R. Lam, D. L. Allaire, and K. E. Willcox, "Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources," in *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2015, p. 0143.

[37] K. Swersky, J. Snoek, and R. P. Adams, "Multi-task bayesian optimization," in *Advances in neural information processing systems*, 2013, pp. 2004–2012.

[38] E. V. Bonilla, K. M. Chai, and C. Williams, "Multi-task gaussian process prediction," in *Advances in neural information processing systems*, 2008, pp. 153–160.

[39] P. Goovaerts *et al.*, *Geostatistics for natural resources evaluation*. Oxford University Press on Demand, 1997.

[40] M. Seeger, Y.-W. Teh, and M. Jordan, "Semiparametric latent factor models," Tech. Rep., 2005.

[41] R. L. Winkler, "Combining probability distributions from dependent information sources," *Management Science*, vol. 27, no. 4, pp. 479–488, 1981.

[42] L. Y. Hsieh, E. Huang, C.-H. Chen, S. Zhang, and K.-H. Chang, "Application of multi-fidelity simulation modelling to integrated circuit packaging," *International Journal of Simulation and Process Modelling*, vol. 11, no. 3-4, pp. 259–269, 2016.

[43] R. Chen, J. Xu, S. Zhang, C.-H. Chen, and L. H. Lee, "An effective learning procedure for multi-fidelity simulation optimization with ordinal transformation," in *Automation Science and Engineering (CASE), 2015 IEEE International Conference on*. IEEE, 2015, pp. 702–707.

[44] S. Cassel, F. Howar, B. Jonsson, and B. Steffen, "Extending automata learning to extended finite state machines," in *Machine Learning for Dynamic Software Analysis: Potentials and Limits*. Springer, 2018, pp. 149–177.

[45] P. Mars, *Learning algorithms: theory and applications in signal processing, control and communications*. CRC press, 2018.

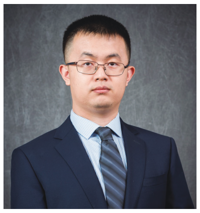
[46] L. Schruben, "Simulation modeling with event graphs," *Commun. ACM*, vol. 26, no. 11, pp. 957–963, Nov. 1983. [Online]. Available: http://doi.acm.org/10.1145/182.358460

[47] J. P. Kleijnen, "Design and analysis of simulation experiments," in *International Workshop on Simulation*. Springer, 2015, pp. 3–22.

[48] W. Xie, B. L. Nelson, and R. R. Barton, "Multivariate input uncertainty in output analysis for stochastic simulation," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 27, no. 1, p. 5, 2016.

[49] A. M. Law, "Statistical analysis of simulation output data: the practical state of the art," in *Proceedings of the 2015 Winter Simulation Conference*. IEEE Press, 2015, pp. 1810–1819.

[50] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

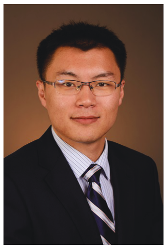


Yunyi Kang received the B.S. degree from the Department of Industrial Systems and Engineering, the Hong Kong Polytechnic University, Kowloon, Hong Kong, China, and the M.S. degree from the Department of Industrial Systems and Engineering, Rutgers University, New Brunswick, NJ, USA, in 2013 and 2015 respectively. He is now working towards the Ph.D. degree at the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA. He was a recipient the best student paper finalist in IEEE CASE.

His research interests are in modeling, analysis and real-time control of production systems.

Logan Mathesen Logan Mathesen is a National Science Foundation Fellow currently pursing his doctorate in Industrial Engineering at Arizona State University. His research focuses on design and analysis of stochastic optimization methods, with particular interest in methods to leverage independent optimizations routines to achieve high-dimensional global optimization over highly non-linear systems. His applications areas currently include design of optimal manufacturing systems and falsification/verification of hybrid cyber physical systems.

Giulia Pedrielli Giulia Pedrielli received her Master, and Ph.D in Mechanical Engineering from Politecnico di Milano, Italy, in 2009 and 2013, respectively. In 2011-2012 she was a PhD Scholar at University of California Berkeley under the mentorship of Professor L.W. Schruben. She pursued a Post-Doc at National University of Singapore from 2014-2016. She is currently Assistant Professor for the School of Computing Informatics and Decision Systems Engineering (CIDSE) in Arizona State University. She develops her research activity in the field of stochastic simulation and simulation optimization with a particular interest Bayesian Optimization. Her application areas span from Logistics and Supply Chain, Manufacturing, Autonomous Vehicles and Bio-productions. She serves as Associated Editor for the Journal of Simulation and the Journal of Flexible Service and Manufacturing.



Feng Ju (M15) received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2010, and the M.S. degree in electrical and computer engineering and Ph.D. degree in industrial and systems engineering from the University of Wisconsin, Madison, WI, USA, in 2011 and 2015, respectively. He is an Assistant Professor with the School of Computing, Informatics & Decision Systems Engineering, Arizona State University, Tempe, AZ, USA. His current research interests include modeling, analysis, continuous improvement, and optimization of manufacturing systems. Dr. Ju is also a member of the Institute for Operations Research and the Management Sciences and Institute of Industrial Engineers. He was a recipient of multiple awards, including the best paper award in IFAC MIM and best student paper finalist in IEEE CASE and IFAC INCOM.

Loo Hay Lee Loo Hay Lee received the S.M. and Ph.D. degrees in engineering science from Harvard University, Cambridge, MA, USA, in 1994 and 1997, respectively. He is an Associate Professor with the Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore. He is also appointed by the Shanghai Municipal Education Commission as an Eastern Scholar Professor for the Shanghai Maritime University. Dr. Lee has served as an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IIE Transactions, and the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. He is currently the Co-Editor for the Journal of Simulation and is a Member in the advisory board for OR Spectrum. He has co-led a team with Dr. Chew to receive the grand prize for the Next-Generation Container Port Challenge in 2013 by proposing a revolutionary double-storey container terminal design, called SINGA Port.