Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

MULTI-FIDELITY MODELS FOR DECOMPOSED SIMULATION OPTIMIZATION PROBLEMS

Nicla Frigerio
Andrea Matta

Department of Mechanical Engineering
via G. la Masa, 1
Politecnico di Milano
20156, Milano, ITALY

Ziwei Lin

Department of Industrial Engineering and Management
800 Dongchuan Rd, Minhang
Shanghai Jiao Tong University
200240, Shanghai, CHINA

ABSTRACT

Hierarchical problem decomposition methods are widely used in optimization when the scale of the problem is large. The master problem is hierarchically decomposed to several sub-problems and the detail level of the sub-problems increases during the optimization from bottom to top. When simulation is used to estimate unknown functions, models with different detail are used at each level. However, the simulation outputs used to solve the sub-problems of a hierarchy level are not used anymore at higher levels. An approach is proposed in this paper to reuse these experiment data to improve the efficiency of the simulation-optimization algorithm. A multi-fidelity surrogate model is built in each sub-problem to guide the search of the optimum. The performance of the approach is numerically assessed with the goal of understanding its potentialities and the effect of algorithm parameters over optimization results.

1 INTRODUCTION

Let us assume to have a master optimization problem in the form:

$$\min\{f(\mathbf{x}) \mid y(\mathbf{x}) \geq b \ , \mathbf{x} \in \mathbb{R}^d\} \tag{1}$$

where $f(\cdot)$ and $y(\cdot)$ are unknown functions of d-dimension vector of system parameters $\mathbf{x}$. Denote $\mathbf{x}^*$ the solution of the master problem in equation (1). When the scale of problem is large, hierarchical problem decomposition methods are widely used to decompose the master problem into several sub-problems that are sequentially optimized. In order to estimate unknown functions, simulation models with different detail are used at each level. Also, analytical models and meta-models can be used to approximate these functions. In this work, we focus on meta-model based simulation-optimization methods to solve large stochastic optimization problems.

1.1 Brief State of the Art

Meta-model based simulation-optimization has been originally developed for physical process (Box and Wilson 1992) and later applied in computer experiments. The search of optimal solution is guided by a meta-model previously built by regression or interpolation techniques on the base of real, or simulated, data. Two main classes of methods can be identified in the literature based on the type of meta-model used.

One class of meta-model based simulation-optimization sequentially uses a set of local meta-models to represent the complete problem in a limited area of interest. The classical Response Surface Methodology (RSM) (Myers and Montgomery 1995) sequentially fits a local linear function and selects new design points based on the gradient information derived from the fitted model. A second-order regression fit

is used to find the optimal solution. The local meta-model is also built in trust region algorithms for simulation-optimization (Osorio and Bierlaire 2013).

The second class of methods builds a global surrogate model to represent the complete problem over the whole solution space and to guide the simulation-optimization process. One of the most famous algorithm is the Efficient Global Optimization (EGO) algorithm firstly proposed by Jones et al. (1998) for deterministic problem and extended to stochastic problems with homogeneous noise by Huang et al. (2006). EGO builds a global meta-model using Kriging technique (Sacks et al. 1989) and updates it in each iteration. The exploitation and the exploration phases of the search are balanced according to the Expected Improvement (EI) criterion (Mockus et al. 1978). Other criteria can be used as alternative to the classical EI for balancing the exploitation and the exploration phases. Among the others, the most probable improvement criterion (Mockus 1989), and the maximum information gain criterion (Srinivas et al. 2012). Other algorithms use the built model to guide the sampling. For instance, Xu (2012) extends Adaptive Hyperbox Algorithm with points sampled according to the meta-model built by Stochastic Kriging (Ankenman et al. 2010).

1.2 Contribution

Meta-model based simulation-optimization methods are very flexible in application and very efficient in finding the optimal solution. The use of simulation budget is optimized during the search. However, these methods always focused onto the master problem in its entirety. On the other side, given a master problem as in equation (1), problem decomposition methods are widely used to efficiently find a solution. Simulation is often used to estimate unknown functions along the search, but the data acquired are used only in a certain hierarchy level. Similarly when analytical methods are used.

The idea of this work is to use meta-models at different levels of fidelity to guide the search of the optimal solution for each sub-problem and to re-use local meta-models for the creation of more accurate meta-models in the next hierarchy levels. At each level of problem decomposition, we create a surrogate model by using the Extended Kernel Regression (EKR) method proposed recently in the literature (Matta et al. 2015; Lin et al. 2016) for combining High-Fidelity (HF) simulation data and a Low-Fidelity (LF) model. As briefly described in section 2, the method is extended to include multiple LF models (Lin et al. 2018). Let us assume to have a set of simulation models $a \in \mathbb{A}$ each representing a portion of the system, i.e., a portion of the master problem. Therefore, we have a set of models that can be considered as HF representations of the sub-problems. In the proposed approach, a meta-model belonging to a certain hierarchy level in the decomposition is created by using simulation at that level as HF model, and the meta-models created at a lower hierarchy as LF models. When combined, an a-priori hierarchy among meta-models of the same level cannot be defined because they might not caught completely the behavior of a superior system. In (Lin et al. 2018), LF models do not have to be hierarchical.

In particular, this work explores the potentialities of a simulation-optimization algorithm for the Buffer Allocation Problem (BAP) in transfer lines where simulation data are re-used along the optimization. A problem decomposition approach is adopted (e.g., Weiss and Stolletz (2015), Shi and Gershwin (2016)). The algorithm starts with the optimization of two-machine lines by creating a meta-model of a two-machine sub-system. Then, at following optimization levels, the algorithm combines simulation outputs with the meta-models created at previous stages. For example, in order to create the meta-model of the sub-system including machines $s = \{1,2,3\}$, the 3-machine line simulation is combined with two meta-models representing sub-systems with machines $s = \{1,2\}$ and $s = \{2,3\}$, respectively. The algorithm provides a local solution with a bottom-up approach toward the solution of the master problem.

The paper is structured as follows: section 2 briefly describes the EKR method, section 3 is dedicated to the BAP, section 4 details the algorithm implemented to solve BAP, and section 5 discusses and analyses the obtained results. Section 6 concludes the paper.

2 A NOTE ON EXTENDED KERNEL REGRESSION

Extended Kernel Regression (EKR) method, recently proposed in the literature (Matta et al. 2015; Lin et al. 2016; Lin et al. 2018), enables the creation of a global surrogate model combining a High-Fidelity (HF) model with one or more Low-Fidelity (LF) models. The HF model is highly accurate but slow in execution, whilst LF models are faster but they may provide biased estimates. The method can autonomously identify which LF models are more helpful in different regions of the domain using the estimated local prediction error, and it assigns area-based weights accordingly.

In more details, let us consider an experiment plan (DOE) of n design points $\boldsymbol{x}_i^0 \in \mathbb{R}^d | i = 1, \cdots, n$ where d is the problem dimension. And let us assume that at each design point the performance were estimated by: (i) a HF model $y_h(\cdot)$, and (ii) a set of m LF models $y_{l_j}(\cdot) | j = 1, \cdots, m$. Given an unknown point $\boldsymbol{x}$ to be predicted, the EKR method takes the LF estimates at this point (i.e., $y_{l_j}(\boldsymbol{x}) | j = 1, \cdots, m$) and applies a correction based on the DOE (i.e., both HF $y_{l_j}(\boldsymbol{x}_i^0) | i = 1, \cdots, n$ and LF $y_h(\boldsymbol{x}_i^0) | i = 1, \cdots, n; j = 1, \cdots, m$) collected data. Local polynomial regression technique is used to fit a polynomial function onto the corrected estimates for each LF model. Gaussian Kernel function is used to assign higher weights to design points that are close to the unknown point and lower weights to design points that are far away. In a final step, the estimates provided by LF models are combined. Higher weights are assigned to LF models with lower local prediction error. The final estimate can be seen as a function $\hat{y}(\mathbf{x}) = \Phi(y_h(\mathbf{x}_i^0), y_{l_j}(\mathbf{x}_i^0), y_{l_j}(\mathbf{x}) | j = 1, \cdots, m; i = 1, \cdots, n)$. The EKR method requires the fitting of parameters Θ_1, θ_2 which are obtained by minimizing the RMSE (Root Mean Square Error) of the design points calculated by leave-one-out cross-validation score method. In this paper, *"fmincon"* Matlab function is used for the optimization of these parameters.

3 BUFFER ALLOCATION PROBLEM DESCRIPTION

The system under study is a transfer line composed by S single-server station and $S-1$ finite buffers between stations. The first station has unlimited supply and the last station is never blocked. Processing times at servers follow a general distribution. The stations may be subject to operation-dependent failures where times to failure and repair times are generally distributed and included in the processing time distribution. Also, blocking after service is assumed and transportation times are negligible or already included in the processing times. Buffer capacity needs to be allocated along the line (Buffer Allocation Problem or BAP) in order to minimize the total buffer capacity while reaching a certain throughput target. Assuming a superior limit B_s to the capacity x_s allocated at buffer behind station s, we obtain the following BAP:

$$\min \left\{ \sum_{s=1}^{S-1} x_s \mid TH(\mathbf{x}, \mathbf{u}) \geq TH_{target} \; ; \; 0 \leq x_s \leq B_s \right\} \tag{2}$$

where $TH(\cdot)$ is a not linear function of decision variables $\mathbf{x} = \{x_s\}$ and other system parameters $\mathbf{u}$. Along the paper, we drop the relation with system parameters $\mathbf{u}$ for ease of representation. The performance of the line can be computed using a sample-path optimization algorithm. We assume the line processes W parts where W_0 parts correspond to the warm-up phase such that, with proper values for W and W_0, the simulation algorithm provides an accurate estimate of the throughput function, i.e., $TH_{SIM}(\mathbf{x}) = TH(\mathbf{x})$.

3.1 Brief Review of BAP Solving Algorithms

The BAP is an NP-hard problem firstly described in Koenigsberg (1959). The classical primal problem as in equation (2) considers the total allocated buffer capacity as objective function and the throughput satisfaction as a constraint. The dual problem maximizes the throughput under a constrained buffer capacity. A recent and comprehensive review of BAP can be found in Weiss et al. (2018b). As performance evaluation methods, the literature uses Markov chain analysis, decomposition and aggregation methods, and simulation. Solving methods are classified into three classes: explicit solutions, iterative optimization methods, and integrated optimization methods. The class of explicit solutions provides a set of rules or

established formulas describing the BAP. Iterative optimization algorithms are composed by two parts: a searching algorithm used to select the candidate buffer allocations, and a method used to evaluate the performance of the candidate solutions. For instance, Hillier (2000) enumerates a set of the most promising solutions, Matta et al. (2012) use an analytical method to improve the performance of stochastic kriging in the proposed optimization algorithm, Kose and Kilincci (2015) combine simulated annealing and genetic algorithm for exploring and exploiting the search spaces, Shi and Gershwin (2016) guide the search with the gradient information obtained from decomposition method, Jafari and Shanthikumar (1989), Diamantidis and Papadopoulos (2004) and Huang et al. (2002) solve the BAP using dynamic programming. Nested partition and branch-and-bound are also frequently used (Shi and Men 2003; Dolgui et al. 2007). Most references belong to the first two classes. The integrated optimization methods formulate the BAP into an MILP able to find a sample-exact solution (Matta 2008; Helber et al. 2011; Alfieri and Matta 2012; Stolletz and Weiss 2013). Instead of using sampling, analytical results are also utilized to build an expression of performance so the problem can be transformed to MILP (Soyster et al. 1979).

In problem decomposition methods, the effort dedicated to simulation or analytical methods developed along the algorithms is exploited only for solving the related sub-problem. For example, Weiss and Stolletz (2015) and Weiss et al. (2018a) propose a very efficient algorithm to solve BAP based on an exact sample-based MIP formulation and Benders Decomposition. Their algorithm decomposes the system in several sub-systems which are locally solved using Benders Decomposition to create lower bounds for the respective buffer capacities. Shi and Gershwin (2016) also decompose the system in several sub-systems without starting from 2-machine lines.

4 SIMULATION-OPTIMIZATION ALGORITHM

A description of the simulation-optimization algorithm is provided: section 4.1 describes the decomposition approach adopted for the production line and section 4.2 details the algorithm.

4.1 Decomposition for Optimization

Adopting the approach of Weiss and Stolletz (2015), we decompose the line into several sub-systems assuming that the first station of each sub-system has unlimited supply and that the last station is never blocked. Define the sub-system size ℓ as the number of buffers included in a certain sub-system; therefore, $S-1$ optimization levels are created where, given size ℓ, the optimization level is composed by a set of $N_\ell = S-\ell$ sub-systems. Figure 1 represents system decomposition and hierarchy levels identified by ℓ. The sub-system $M_{\ell,j}|\ell = 1,\cdots,S-1;\ j = 1,\cdots,N_\ell$ identifies the portion of the line from station j to station $j+\ell$ and it includes the (ℓ)-tuple of buffer capacities from x_j to $x_{j+\ell-1}$. Therefore, the sub-system $M_{\ell,j}$ has ℓ decision variables in common with the complete system $M_{S-1,1}$. Since, the isolated throughput of each sub-system $M_{\ell,j}$ is higher than that of the original system, the solution of sub-system $M_{\ell,j}$ provides a lower bound to the buffer capacity of the sub-system. Once the algorithm arrives at the highest hierarchy $\ell = S-1$, the optimal solution is found.

4.2 Multi-fidelity Simulation-optimization Algorithm

The algorithm is represented in Figure 2 and it can be described as follows:

1. Initialize $\ell = 1$ and no bounds.
2. For $j = 1,\cdots,N_\ell$, solve BAP of each sub-system $M_{\ell,j}$ using a meta-model created by EKR method and obtain the local solution $\mathbf{x}^*_{\ell,j}$.
3. Add lower bounds $\sum_{i=j}^{j+\ell-1} x_i \leq \sum_{i=j}^{j+\ell-1} \mathbf{x}^*_{\ell,j}$.
4. If $\ell < S-1$, increase $\ell = \ell+1$ and go to Step 2. If $\ell = S-1$, the optimal solution is found $\mathbf{x}^* = \mathbf{x}^*_{S-1,1}$ and stop.

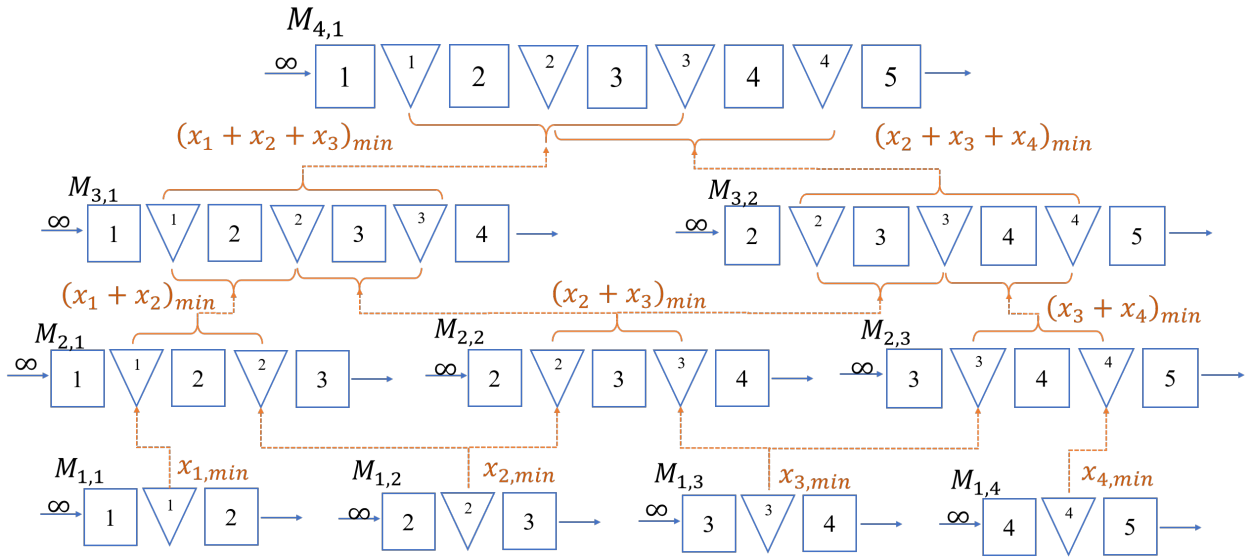

Figure 1: System decomposition in sub-system $M_{\ell,j}|\forall \ell, j$ with stations s (represented with squares) and buffer capacities x_s (represented with triangles). Blue (plain) arrows represent material flow from infinite source to infinite downstream buffers. Orange (dotted) arrows represent the application of lower bounds to superior systems. For example, sub-system $M_{1,1}$ provides a lower bound to buffer capacity x_1 to sub-systems $M_{\ell,1}|\ell = 2,3,4$, whilst sub-system $M_{2,1}$ provides a lower bound to buffer capacities $x_1 + x_2$ to sub-systems $M_{3,1}$ and $M_{4,1}$.

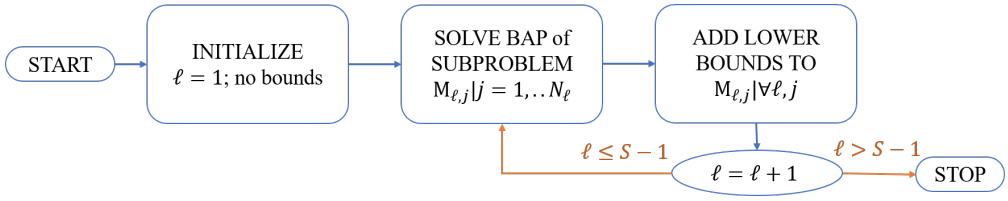

Figure 2: Main description of the algorithm.

Into details, the optimization of a certain sub-system $M_{\ell,j}$ is executed as in Figure 3 by four modules:

1. The *Sampling module* uses a space filling design (i.e., Latin hypercube design) to sample $n_{0,\ell}$ vectors of size ℓ where each element is non-negative and constrained by the maximum buffer capacity.
2. The *Simulation module* provides sub-system HF performance.
3. The *EKR Fitting module* creates a meta-model for sub-system $M_{\ell,j}$ using:
   - Simulation data of sub-system $M_{\ell,j}$ that are considered as HF data;
   - Meta-model data of sub-systems $M_{\ell-1,j}$ and $M_{\ell-1,j+1}$ (i.e., the sub-system of size $\ell - 1$ included in sub-system $M_{\ell,j}$) are considered as LF data. For $\ell = 1$, only simulation data are used.
4. The *Optimization module* solves the BAP using *"fseminf"* Matlab function providing $\mathbf{x}^*_{\ell,j}$. J independent optimizations are executed by changing the initial solution.

In order to reduce the meta-model error, the first three modules can be used iteratively to update the model with new design points. In principles, the update of the meta-model can be performed at each new sample. In the current version of the algorithm, the meta-model is updated one time after the initial creation.

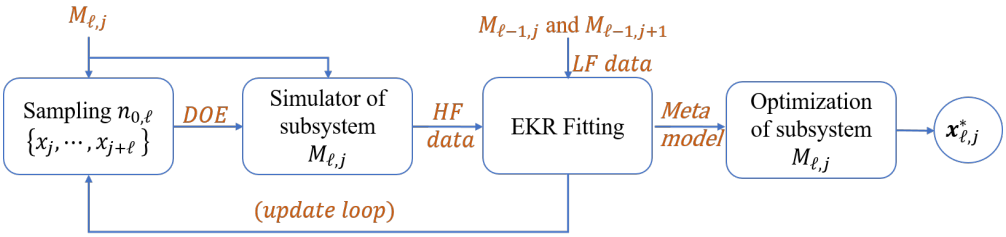

Figure 3: Detailed description of the sub-problem solving block of the algorithm.

Indeed, a certain portion α of the available budget (i.e., $\alpha \cdot n_{0,\ell}$) is used to create an initial meta-model of sub-system $M_{\ell,j}$ which provides an estimate of system performance. The variability of the estimate can be evaluated from the EKR method: the larger the standard deviation, the larger the bias. The Sampling module performs a second sampling of $(1-\alpha) \cdot n_{0,\ell}$ additional points with an Acceptance-and-Rejection Algorithm using the estimated standard deviation of the meta-model as probability density function. More points are sampled in the region where the estimate of the performance has larger variability. The EKR fitting module updates the meta-model including the additional simulated design points.

We assume that the simulated sample path is large enough to assure convergence of simulated performance so that we use only one replication for each solution in the Simulation module. Also, common random numbers are used among sub-systems. Since optimization results rely on the meta-model, the local solution found might not be the real optimum. Therefore, generated bounds are relaxed of percentage β. A list of algorithm parameters is in Table 1.

Table 1: List of algorithm parameters.

Parameter	Description
K_{tot}	Total simulation budget
$n_{0,\ell} \vert \ell = 1, \cdots, S-1$	Simulation budget allocated to each model j of size ℓ
α	Portion of the simulation budget allocated to initial uniform sampling
J	Number of independent optimizations using *"fseminf"* Matlab function
β	Relaxation of lower bounds

5 NUMERICAL RESULTS

Several experiments are executed as described in section 5.1 and the numerical results are presented in sections 5.2, 5.3 and 5.4.

5.1 Description of Experiments

The algorithm and the simulator are implemented in Matlab R2017a. The numerical study is performed on an Intel Core i7-6500U with 2.50GHz and 16GB of RAM and it required at most 2.5 min to solve one replication.

We refer to the numerical study of Matta (2008). The system under study is a 5-station transfer line with bottleneck at the end and buffer capacity constrained to $B_s = 20, \forall s = 1, \cdots, S-1$. The processing times are exponentially distributed with a base processing rate of 7 and a bottleneck processing rate of 6. Buffer allocation is minimized by assuring a target throughput of 5.776. For this case, the algorithm proposed by Weiss and Stolletz (2015) converges to the optimal buffer capacity of 38 as the simulation length increases up to $W = 5,000,000$. The simulation length $W = 250,000$ is appropriate to provide reliable estimate of the throughput for the analyzed cases and the warm-up length is set to $W_0 = 2,000$ parts.

We consider increasing simulation budget (i.e., K_{tot} = 125; 250; 500; 1,000 and 2,000) whose half is dedicated to the initial meta-model creation and half to the meta-model update, i.e., $\alpha = 0.5$. Also, we

consider four different distributions of the budget along optimization levels: (A1) all budget is dedicated to HF simulations (i.e., [0; 0; 0; 100]%), (A2) balanced budget (i.e., [25; 25; 25; 25]%), (A3) increasing budget (i.e., [15; 20; 30; 35]%), and (A4) decreasing budget (i.e., [35; 30; 20; 15]%). Whenever the budget dedicated to a meta-model exceeds the number of alternative solutions for that sub-system, the budget is reallocated to other levels. More details are provided in Table 2. An additional setting (A5) is considered where the budget is balanced but the meta-model is created using the standard Kernel Regression or KR (Wand and Jones 1995), i.e, the data from lower levels are not re-used for meta-model building but decomposition is used to bound the domain. The lower bounds found are relaxed by $\beta = 10\%$ to cope with meta-model error.

Table 3 collects the optimization results as average onto 5 sample-paths. A total of 25 algorithm settings are compared and results are obtained by executing 20 replications for each sample-path. For each path, the exact solution is found with enumeration.

Table 2: Simulation budget $n_{0,\ell}$ allocated according to hierarchy ℓ: $[n_{0,1}; n_{0,2}; n_{0,3}; n_{0,4}]$.

A1 HF	A2 and A5 Balanced	A3 Increasing	A4 Decreasing
[0; 0; 0; 125]	[8; 11; 16; 32]	[5; 9; 19; 44]	[11; 13; 13; 19]
[0; 0; 0; 250]	[16; 21; 32; 63]	[10; 17 ;38 ; 88]	[21; 26; 26; 39]
[0; 0; 0; 500]	[21; 47; 70; 139]	[21; 64; 64; 96]	[21; 34; 75; 175]
[0; 0; 0; 1,000]	[21; 102; 153; 306]	[21; 141; 141; 212]	[21; 72; 162; 378]
[0; 0; 0; 2,000]	[21; 213; 320; 639]	[21; 295; 295; 443]	[21; 151; 339; 789]

5.2 Accuracy of the Final Meta-model

The ability of EKR method in fitting the HF function of system performance $TH(\mathbf{x})$ is clear from the Mean Absolute Relative Error (MARE) of the final meta-model created (i.e., $M_{4,1}$) which is rapidly decreasing as the simulation budget increases (Table 3). The MARE is calculated over 5,000 checkpoints independently from the optimization algorithm. The use of EKR (i.e., settings A2, A3 and A4) compared to classical KR (i.e., settings A1 and A5) is significantly advantageous in terms of the prediction capability of the meta-model falling below $MARE = 2\%$ with 250 simulations. Also, simulation is executed at different fidelity levels thus it is faster than that at HF level. The budget allocation at different levels (A2, A3, A4) shows a trade-off between having good low-fidelity models having more HF simulation data. Its influence seems not significant in this case. However, more analysis should be performed in future works to find more efficient budget allocations. For example, the efficiency of the algorithm can be further improved by assigning more simulation effort to low hierarchy models where simulation is fast without reducing the accuracy of the final model. In more details, Figure 4 represents the boxplot of errors for a single path. Setting A5 has the highest MARE because the simulation budget dedicated to the meta-model creation is limited to $n_{0,4}$ without the use of any LF model.

5.3 Accuracy of the Solution

Results in Table 3 show that the use of meta-models is helpful in the search above all when the simulation budget is limited. The absolute distances of the found optima from the sample-exact optimum are significantly different in settings A2, A3, and A4 compared to that of setting A1 and A5. When only KR is used (A1 and A5) the algorithm does not provide good results. The setting A5 is worse than A1 because the advantage of considering the lower bounds is covered by the increased error of final meta-model due to reduced simulation effort at HF. Although it does not appear significant for this case, there exists a trade-off between dedicating more effort to high hierarchy simulations and to low hierarchy simulations. In the first case (A3 setting), simulation is dedicated where sub-system performance is closer to the final ones. In the second

Table 3: Numerical results by algorithm setting (average on 5 sample-paths).

Algorithm Setting		**MARE**	**Abs. distance**	**Abs. deviation from**
Budget allocation	K_{tot}		**from optimum**	$TH_{target} = 5.776$
A1 HF	125	3.35% ± 0.08%	4.86 ± 0.30	8.09% ± 1.60%
	250	2.70% ± 0.05%	4.12 ±0.48	4.38% ± 3.25%
	500	2.14% ± 0.02%	3.00 ± 0.23	2.71%± 1.25%
	1000	1.69% ± 0.03%	2.45 ± 0.34	0.56% ± 0.05%
	2000	1.30% ± 0.01%	1.89 ± 0.07	0.45% ± 0.03%
A5 KR (BAL)	125	4.66% ± 0.15%	7.28 ± 1.06	1.32% ± 0.15%
	250	4.15% ± 0.27%	7.58 ± 1.33	1.71% ± 0.39%
	500	3.27% ± 0.08%	6.17 ± 1.11	1.18% ± 0.25%
	1000	2.62% ± 0.28%	4.35 ± 1.37	0.86% ± 0.17%
	2000	1.99% ± 0.02%	2.99 ± 0.22	0.66% ± 0.03%
A2 BAL	125	3.30% ± 0.12%	3.54 ± 0.39	1.41% ± 0.42%
	250	1.86% ± 0.07%	1.70 ± 0.22	0.63% ±0.23%
	500	1.00% ± 0.05%	0.92 ± 0.23	0.25% ± 0.03%
	1000	0.60% ± 0.01%	1.21 ± 0.36	0.26% ± 0.06%
	2000	0.37% ± 0.01%	0.60 ± 0.12	0.15% ± 0.01%
A3 INCR	125	3.48%± 0.20%	4.23 ± 1.27	1.34% ± 0.35%
	250	2.15% ± 0.08%	2.34 ± 0.61	0.72% ± 0.28%
	500	1.16% ± 0.06%	1.21 ± 0.32	0.38% ± 0.03%
	1000	0.72% ± 0.35%	1.04 ± 0.16	0.23% ± 0.03%
	2000	0.37% ± 0.02%	0.58 ± 0.33	0.14% ± 0.06%
A4 DECR	125	3.80%± 0.45%	3.49 ± 0.64	1.59% ± 0.41%
	250	1.91% ± 0.07%	2.03 ± 0.39	0.74% ± 0.31%
	500	1.07% ± 0.04%	1.33 ± 0.42	0.43% ± 0.19%
	1000	0.65% ± 0.02%	1.13± 0.38	0.27% ± 0.05%
	2000	0.40% ± 0.01%	0.90 ± 0.15	0.20% ± 0.03%

case (A4 setting), more simulations are dedicated to build good LF meta-models. A proper trade-off of these settings might achieve better performance in wider problems.

Moreover, as the simulation budget increases, the variability of the solution decreases as shown in Figure 5. Problem decomposition is shown to be more efficient although the variability of results is still significant; indeed, the variability represented in Figure 5 is entirely associated to the variability of the algorithm in the creation of the meta-models.

5.4 Local Solutions and Bounds

Let us focus on setting A2 with a balanced distribution of the budget and $K_{tot} = 500$, which is one of the most performing setting. Nevertheless, the following discussion can be applied to any of the tested settings. Table 4 and Table 5 collect results as averages of results obtained by optimizing 5 sample-paths (20 replications).

A total of 10 meta-models are created along the optimization process and Table 4 contains the MARE of each of the meta-models $M_{\ell,j}$ in estimating the performance of the final system (5,000 checkpoints). The process of creating meta-models in sequence is reducing the error of the estimate as the hierarchy increases. Moreover, some meta-models of low hierarchy are very accurate in estimating the performance of the final system because they contain a parameter that is highly affecting the system performance. In this case, the bottleneck is located at station $s = 5$ and the parameter mostly affecting the behavior of the system is x_4.

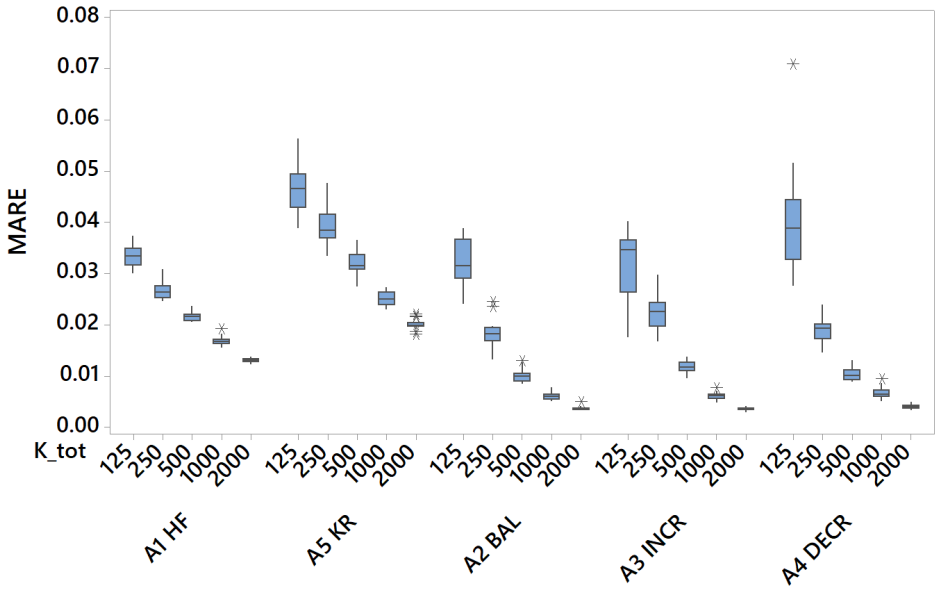

Figure 4: Boxplot of MARE (Mean Absolute Relative Error) of meta-model $M_{4,1}$ according to algorithm setting (single path, 20 replications).

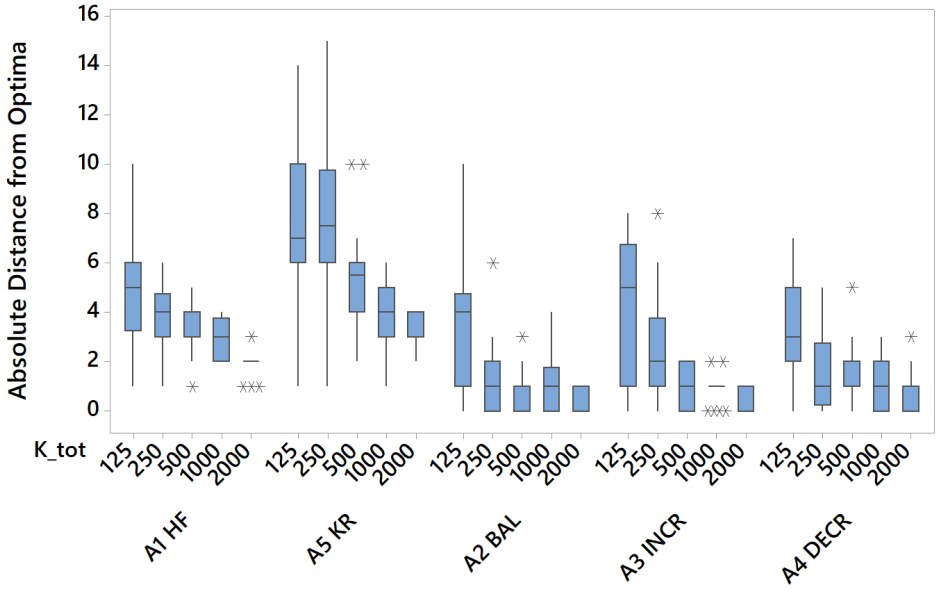

Figure 5: Boxplot of the gap between the solution found and the real optimum (i.e., $B^*_{tot}|_{sim} = 39$) according to algorithm setting (single path, 20 replications).

In terms of budget allocation, the accuracy of local meta-models might drive a more efficient allocation of

simulation effort. Table 4 is also a confirmation of our previous statement about the meta-model hierarchy that cannot be a-prior defined.

Table 5 shows the local solutions obtained by the optimization of sub-systems $M_{\ell,j}$. For the lowest hierarchy $\ell = 1$, all possible solutions $x_s \in [0, B_s]$ are evaluated by simulating sub-systems $M_{1,j} | j = 1,2,3,4$ and the sub-problem optima $\{x_j\}_1^*, \forall j = 1,2,3,4$ are sample-exact. As the hierarchy ℓ increases, the current version of the algorithm does not prove the convergence to solution; therefore, variability is observed depending on local prediction error of the meta-model used. For the analyzed cases, the range width of local solutions never exceeds $[-2.95; +2.3]$ with respect to the average in Table 5.

Table 4: MARE of meta-models $M_{\ell,j}$ with algorithm setting A2 and $K_{tot} = 500$ (average on 5 sample-paths).

	$j=1$	$j=2$	$j=3$	$j=4$
$\ell=1$	17.13% ± 0.16%	17.27% ± 0.20%	17.13% ± 0.10%	5.68% ± 0.10%
$\ell=2$	10.95% ± 0.19%	10.91% ± 0.11%	3.26% ± 0.09%	
$\ell=3$	7.17% ± 0.16%	1.83% ± 0.06%		
$\ell=4$	1.00% ± 0.05%			

Table 5: Local solution of each meta-model $M_{\ell,j}$ with algorithm setting A2 and $K_{tot} = 500$ (average on 5 sample-paths). The solution found for meta-models of size $\ell = 1$ is exact because all candidate solutions are simulated.

	$j=1$	$j=2$	$j=3$	$j=4$
$\{x_j\}_1^*$	3 ± 0	3 ± 0	3 ± 0	9 ± 0
$\{x_j + x_{j+1}\}_2^*$	10.73 ± 0.11	10.78 ± 0.22	19.18 ± 0.26	
$\{x_j + x_{j+1} + x_{j+2}\}_3^*$	19.89 ± 0.22	28.31 ± 0.47		
$\{x_j + x_{j+1} + x_{j+2} + x_{j+3}\}_4^*$	38.18 ± 0.56			

6 CONCLUSIONS AND FUTURE DEVELOPMENTS

The numerical study presented in this work enables the identification of strengths and weaknesses of the proposed approach. The re-use of simulation data in the creation of meta-models of different hierarchy is positively affecting the performance of the decomposed simulation-optimization approach. The current version of the algorithm relies only on the estimate provided by the meta-models for the optimization. Iterative methods will be included in the algorithm in order to provide sample-exact solutions, for instance, expected improvement based methods.

The effect of different budget allocations (A2, A3, and A4 settings) seems to not significantly affect the algorithm efficiency. However, a sensitivity analysis on algorithm parameters is required when dealing with larger scale problems. In principles, an optimal budget allocation method among sub-problems should be investigated in order to efficiently use the available budget.

The goodness of low hierarchy meta-models is key to guide the search toward the optima through the creation of effective bounds. Global sampling is executed to create an initial meta-model and then a localized sampling is used for update. To perform an appropriate sampling for the creation of meta-models is crucial to achieve small errors in the estimates. Currently, bounds are used only in the optimization phase, whereas they might be also used during the sampling phase.

Furthermore, it might happen that a meta-model of low hierarchy is already very accurate in estimating the system performance because it includes a parameter with high impact on the objective function, e.g., the bottleneck station in BAP. These models can be used directly at higher levels. Also, more than two LF models can be combined during the creation of a meta-model. For example, analytical methods can be included as LF models together with the lower level meta-model in EKR method.

REFERENCES

Alfieri, A., and A. Matta. 2012. "Mathematical Programming Formulations for Approximate Simulation of Multistage Production Systems". *European Journal of Operational Research* 219(3):773–783.

Ankenman, B., B. L. Nelson, and J. Staum. 2010. "Stochastic Kriging for Simulation Metamodeling". *Operations Research* 58(2):362–370.

Box, G. E., and K. B. Wilson. 1992. "On the Experimental Attainment of Optimum Conditions". In *Breakthroughs in Statistics*, 270–310. New York: Springer New York.

Diamantidis, A., and C. Papadopoulos. 2004. "A Dynamic Programming Algorithm for the Buffer Allocation Problem in Homogeneous Asymptotically Reliable Serial Production Lines". *Mathematical Problems in Engineering* 2004(3):209–223.

Dolgui, A., A. V. Eremeev, and V. S. Sigaev. 2007. "HBBA: Hybrid Algorithm for Buffer Allocation in Tandem Production Lines". *Journal of Intelligent Manufacturing* 18(3):411–420.

Helber, S., K. Schimmelpfeng, R. Stolletz, and S. Lagershausen. 2011. "Using Linear Programming to Analyze and Optimize Stochastic Flow Lines". *Annals of Operations Research* 182(1):193–211.

Hillier, M. S. 2000. "Characterizing the Optimal Allocation of Storage Space in Production Line Systems with Variable Processing Times". *IIE Transactions* 32(1):1–8.

Huang, D., T. T. Allen, W. I. Notz, and N. Zeng. 2006. "Global Optimization of Stochastic Black-Box Systems Via Sequential Kriging Meta-Models". *Journal of Global Optimization* 34(3):441–466.

Huang, M.-G., P.-L. Chang, and Y.-C. Chou. 2002. "Buffer Allocation in Flow-Shop-Type Production Systems with General Arrival and Service Patterns". *Computers & Operations Research* 29(2):103–121.

Jafari, M. A., and J. G. Shanthikumar. 1989. "Determination of Optimal Buffer Storage Capacities and Optimal Allocation in Multistage Automatic Transfer Lines". *IIE Transactions* 21(2):130–135.

Jones, D. R., M. Schonlau, and W. J. Welch. 1998. "Efficient Global Optimization of Expensive Black-Box Functions". *Journal of Global Optimization* 13(4):455–492.

Koenigsberg, E. 1959. "Production Lines and Internal Storage: A Review". *Management Science* 5(4):410–433.

Kose, S. Y., and O. Kilincci. 2015. "Hybrid Approach for Buffer Allocation in Open Serial Production Lines". *Computers & Operations Research* 60:67–78.

Lin, Z., A. Matta, N. Li, and J. G. Shanthikumar. 2016. "Extended Kernel Regression: A Multi-Resolution Method to Combine Simulation Experiments with Analytical Methods". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder et al., 590–601. Piscataway, New Jersey: IEEE.

Lin, Z., A. Matta, and J. G. Shanthikumar. 2018. "Combining Simulation Experiments and Analytical Models with Different Area-Based Accuracy for Performance Evaluation of Manufacturing Systems". *IISE Transactions* doi:10.1080/24725854.2018.1490046.

Matta, A. 2008. "Simulation Optimization with Mathematical Programming Representation of Discrete Event Systems". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason et al., 1393–1400. Piscataway, New Jersey: IEEE.

Matta, A., N. Li, Z. Lin, J. Shanthikumar et al. 2015. "Operational Learning of Approximate Analytical Methods for Performance Evaluation of Manufacturing Systems". In *10th Conference on Stochastic Models of Manufacturing and Service Operations SMMSO 2013*, 137–144. University of Thessaly Press.

Matta, A., M. Pezzoni, and Q. Semeraro. 2012. "A Kriging-Based Algorithm to Optimize Production Systems Approximated by Analytical Models". *Journal of Intelligent Manufacturing* 23(3):587–597.

Mockus, J. 1989. *Bayesian Approach to Global Optimization*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Mockus, J., V. Tiesis, and A. Zilinskas. 1978. "The Application Bayesian Methods for Seeking the Extremum". *Towards Global Optimisation* 2:117–129.

Myers, R. H., and D. C. Montgomery. 1995. *Response Surface Methodology: Process and Product in Optimization Using Designed Experiments*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Osorio, C., and M. Bierlaire. 2013. "A Simulation-Based Optimization Framework for Urban Transportation Problems". *Operations Research* 61(6):1333–1345.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn. 1989. "Design and Analysis of Computer Experiments". *Statistical Science* 409–423.

Shi, C., and S. B. Gershwin. 2016. "A Segmentation Approach for Solving Buffer Allocation Problems in Large Production Systems". *International Journal of Production Research* 54(20):6121–6141.

Shi, L., and S. Men. 2003. "Optimal Buffer Allocation in Production Lines". *IIE Transactions* 35(1):1–10.

Soyster, A. L., J. Schmidt, and M. Rohrer. 1979. "Allocation of Buffer Capacities for a Class of Fixed Cycle Production Lines". *AIIE Transactions* 11(2):140–146.

Srinivas, N., A. Krause, S. M. Kakade, and M. W. Seeger. 2012. "Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting". *IEEE Transactions on Information Theory* 58(5):3250–3265.

Stolletz, R., and S. Weiss. 2013. "Buffer Allocation Using Exact Linear Programming Formulations and Sampling Approaches". *IFAC Proceedings Volumes* 46(9):1435–1440.

Wand, M. P., and M. C. Jones. 1995. *Kernel Smoothing*. Monographs on Statistics & Applied Probability. New York: Chapman and Hall/CRC.

Weiss, S., and R. Stolletz. 2015. "Buffer Allocation in Stochastic Flow Lines Via Sample-Based Optimization with Initial Bounds". *OR Spectrum* 37(4):869–902.

Weiss, S., A. Matta, and R. Stolletz. 2018a. "Optimization of Buffer Allocations in Flow Lines with Limited Supply". *IISE Transactions* 50(3):191–202.

Weiss, S., J. A. Schwarz, and R. Stolletz. 2018b. "The Buffer Allocation Problem in Production Lines: Formulations, Solution Methods, and Instances". *IISE Transactions* doi:10.1080/24725854.2018.1442031.

Xu, J. 2012. "Efficient Discrete Optimization Via Simulation Using Stochastic Kriging". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque et al., 466–477. Piscataway, New Jersey: IEEE.

AUTHOR BIOGRAPHIES

NICLA FRIGERIO is Post-Doc Research Fellow in the Mechanical Engineering Department of Politecnico di Milano, Italy. She holds a Ph.D. in mechanical engineering from Politecnico di Milano. Her research focuses on production system modeling and stochastic control with particular attention to control problem for improving energy efficiency in manufacturing. Her email address is nicla.frigerio@polimi.it.

ZIWEI LIN is Ph.D. candidate in the Department of Industrial Engineering and Management of Shanghai Jiao Tong University, China. Her thesis focuses on performance evaluation and optimization of manufacturing systems based on multi-fidelity models. Her email address is linziwei@sjtu.edu.cn.

ANDREA MATTA is Professor at Politecnico di Milano, where he currently teaches integrated manufacturing systems and manufacturing. His research area includes analysis and design of manufacturing and health care systems. He is Editor-in-Chief of Flexible Services and Manufacturing Journal. His email address is andrea.matta@polimi.it.