Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

GREEN SIMULATION ASSISTED REINFORCEMENT LEARNING WITH MODEL RISK FOR BIOMANUFACTURING LEARNING AND CONTROL

Hua Zheng, Wei Xie

Department of Mechanical and Industrial Engineering
Northeastern University
Boston, MA 02115, USA

M. Ben Feng

Department of Statistics and Actuarial Science
University of Waterloo
Waterloo, Ontario, CANADA

ABSTRACT

Biopharmaceutical manufacturing faces critical challenges, including complexity, high variability, lengthy lead time, and limited historical data and knowledge of the underlying system stochastic process. To address these challenges, we propose a green simulation assisted model-based reinforcement learning to support process online learning and guide dynamic decision making. Basically, the process model risk is quantified by the posterior distribution. At any given policy, we predict the expected system response with prediction risk accounting for both inherent stochastic uncertainty and model risk. Then, we propose green simulation assisted reinforcement learning and derive the mixture proposal distribution of decision process and likelihood ratio based metamodel for the policy gradient, which can selectively reuse process trajectory outputs collected from previous experiments to increase the simulation data-efficiency, improve the policy gradient estimation accuracy, and speed up the search for the optimal policy. Our numerical study indicates that the proposed approach demonstrates the promising performance.

1 INTRODUCTION

To address critical needs in biomanufacturing automation, in this paper, we introduce a green simulation assisted Bayesian reinforcement learning to support bioprocess online learning and guide dynamic decision making. The biomanufacturing industry is growing rapidly and becoming one of the key drivers of personalized medicine. However, biopharmaceutical production faces critical challenges, including complexity, high variability, long lead time, and very limited process data. Biotherapeutics are manufactured in living cells whose biological processes are complex and have highly variable outputs (e.g., product critical quality attributes (CQAs)) whose values are determined by many factors (e.g., raw materials, media, critical process parameters (CPPs)). As new biotherapeutics (e.g., cell and gene therapies) become more and more "personalized," biomanufacturing requires more advanced manufacturing protocols. In addition, the analytical testing time required by biopharmaceuticals of complex molecular structure is lengthy, and the process observations are relatively limited.

Driven by these challenges, we consider the model-based reinforcement learning (MBRL) or Markov Decision Process (MDP) to fully leverage the existing bioprocess domain knowledge, utilize the limited process data, support online learning, and guide dynamic decision making. At each time step t, the system is in state s_t, and the decision maker takes the action a_t by following a policy $a_t = \pi_t(a_t|s_t)$. At the next time step $(t+1)$, the system evolves to new state s_{t+1} by following the state transition probabilistic model $P(s_{t+1}|s_t, a_t; \boldsymbol{\omega})$, and then we collect a reward $r_t(a_t, s_t)$. *Thus, the statistical properties and dynamic evolution of stochastic control process depend on decision policy π_t and state transition model $P(s_{t+1}|s_t, a_t; \boldsymbol{\omega})$.* In the biomanufacturing, the prior knowledge of state transition model is constructed based on the existing biological/physical/chemical mechanisms and dynamics. The unknown model parameters $\boldsymbol{\omega}$ (e.g., cell growth, protein production, and substrate consumption rates in cell culture; nucleation rate and heat transfer

coefficients in freeze drying) will be online learned and updated as the arrivals of new process data. *The optimal policy depends on the current knowledge of process model parameters.*

In this paper, we propose a green simulation assisted Bayesian reinforcement learning (GS-RL) to guide dynamic decision making. Given any policy, we predict the expected system response with prediction risk accounting for both process inherent stochastic uncertainty and model estimation uncertainty, call *model risk*. The model risk is quantified by the posterior distribution and it can efficiently leverage the existing bioprocess domain knowledge through the selection of prior and support the online learning. *Thus, the proposed Bayesian reinforcement learning can provide the robust dynamic decision guidance*, which can be applicable for cases with various amount of process historical data. In addition, motivated by the studies on green simulation (i.e., Feng and Staum (2017) and Dong et al. (2018)), we propose the stochastic control process likelihood ratio-based metamodel to improve the policy gradient estimation, which can fully leverage the historical trajectories generated with various state transition models and policies. *Therefore, the proposed green simulation assisted Bayesian reinforcement learning can: (1) incorporate the existing process domain knowledge; (2) facilitate the interprertable online learning; (3) guide complex bioprocess dynamic decision making; and (4) provide the reliable, flexible, robust, and coherent decision guidance.*

For the model-free inforcement learning, Mnih et al. (2015) introduce the experience replay (ER) to reuse the past experience, increase the data efficiency, and decrease the data correlation. It randomly samples and reuses the past trajectories. Built on ER, Schaul et al. (2016) further propose the prioritized experience replay (PER), which prioritizes the historical trajectories based on temporal-difference error.

The **main contribution** of our study is to propose a green simulation assisted Bayesian reinforcement learning (GS-RL). Even though both GS-RL and PER are motivated by "experience replay" and reuse the historical data, there is the fundamental difference between GS-RL and PER. In our approach, the posterior distribution of state transition model can provide the risk- and science-based knowledge of underlying bioprocess dynamic mechanisms, and facilitate the online learning. Then, the likelihood ratio of stochastic decision process is used to construct the metamodel of policy gradient in the complex decision process space, accounting for the selection and impact of both policy and state transition model. It allows us to reuse the trajectories from previous experiments, and the weight assigned to each trajectory depends on its importance measured by the spatial-temporal distance of decision processes. In addition, a mixture process proposal distribution used in the likelihood ratio can improve the estimation accuracy and stability of policy gradient and speed up the search for the optimal policy. Since the model risk is automatically updated during the learning, our approach can dynamically adjust the importance weights on the previous trajectories, which makes GS-RL flexible, efficient, and automatically deal with non-stationary bioprocess.

The organization of the paper is as follow. In Section 2, we provide the problem description. To facilitate the biomanufacturing process online learning and automation, we focus on the model-based reinforcement learning with the posterior distribution quantifying the model risk. Then, in Section 3, we propose the green simulation assisted policy gradient, which can fully leverage the process trajectories obtained from previous experiments and speed up the search for the optimal policy. After that, a biomanfuacturing example is used to study the performance of proposed approach and compare it with the state-of-art policy gradient approaches in Section 4. We conclude this paper in Section 5.

2 PROBLEM DESCRIPTION AND MODEL BASED REINFORCEMENT LEARNING

To facilitate the biomanufacturing automation, we consider the reinforcement learning for finite horizon problem. In Section 2.1, we suppose the underlying model of production process is known and review the model-based reinforcement learning. Since the process model is typically unknown and estimated by very limited process data in the biomanufacturing, in Section 2.2, the posterior distribution is used to quantify the model risk and the posterior predictive distribution, accounting for stochastic uncertainty and model risk, is used to generate the trajectories characterizing the overall prediction risk. Thus, in this paper, we focus on the model-based reinforcement learning with model risk so that we can efficiently leverage the existing process knowledge, support online learning, and guide process dynamic decision making.

2.1 Model-Based Reinforcement Learning for Dynamic Decision Making

We formalize model-based reinforcement learning or Markov decision process (MDP) over finite horizon H as $(\mathscr{S},\mathscr{A},P,r,s_1,H)$, where $\mathscr{S}$ is a set of states, s_1 is the starting state, $\mathscr{A}$ is the set of actions. The process proceeds in discrete time step $t=1,2,...,H$. In each t-th time step, the agent observes the current state $s_t\in\mathscr{S}$, takes an action $a_t\in\mathscr{A}$, and observes a feedback in form of a reward signal $r_{t+1}\in\mathbb{R}$. Moreover, let $\pi_{\boldsymbol{\theta}}:\mathscr{S}\rightarrow\mathscr{A}$ denote a policy specified by parameter vector $\boldsymbol{\theta}\in\mathbb{R}^d$. The policy is a function of current state, $a_t=\pi_{\boldsymbol{\theta}}(s_t)$, whose output is action for deterministic policy or its selection probabilities for random policy. For *non-stationary* finite horizon MDP, we can write $\boldsymbol{\pi}_{\boldsymbol{\theta}}=(\pi_{\boldsymbol{\theta}}^1,\ldots,\pi_{\boldsymbol{\theta}}^H)$.

Let $P(s_{t+1}|s_t,a_t;\boldsymbol{\omega}^c)$ represent the state transition model characterizing the probability of transitioning to a particular state s_{t+1} from state s_t. Suppose the underlying process model can be characterized by parameters $\boldsymbol{\omega}^c$. Let $D_{P_{\boldsymbol{\omega}^c}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{\tau})$ denote the probability distribution of the trajectory

$$\boldsymbol{\tau}=\boldsymbol{\tau}_{[1:H-1]}\equiv(s_1,a_1,s_2,a_2,\ldots,s_{H-1},a_{H-1},s_H)$$

of state-action sequence over transition probabilities parameterized by transition model $P(s_{t+1}|s_t,a_t;\boldsymbol{\omega}^c)$ starting from state s_1 and following policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. *The bioprocess trajectory length H can be scenario-dependent.* For example, it can depend on the CQAs of raw materials and working cells. We write the distribution of decision process trajectory as

$$D_{P_{\boldsymbol{\omega}^c}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{\tau})\equiv p(s_1;\boldsymbol{\omega}^c)\prod_{t=1}^{H-1}\pi_{\boldsymbol{\theta}}^t(a_t|s_t)p(s_{t+1}|s_t,a_t;\boldsymbol{\omega}^c). \tag{1}$$

Let $R(\boldsymbol{\tau})$ denote the expected total reward for the trajectory (sample path) $\boldsymbol{\tau}$ starting from s_1, i.e., $R(\boldsymbol{\tau})\equiv\sum_{t=1}^{H-1}\gamma^{t-1}r_t(s_t,a_t)$, where γ is the discount factor and the reward $r_t(s_t,a_t)$ occurring in the t-th time step depends on the state s_t and action a_t. Therefore, given the process model specified by $\boldsymbol{\omega}^c$, we are interested in finding the optimal policy maximizing the expected total reward,

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\star}(\cdot|\boldsymbol{\omega}^c)=\arg\max_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\mu^c(\boldsymbol{\pi}_{\boldsymbol{\theta}})\equiv\arg\max_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\mathbb{E}_{\boldsymbol{\tau}\sim D_{P_{\boldsymbol{\omega}^c}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{\tau})}\left[\sum_{t=1}^{H-1}\gamma^{t-1}r_t\middle|\boldsymbol{\pi}_{\boldsymbol{\theta}},s_1\right]. \tag{2}$$

2.2 Model Risk Quantification and Bayesian Reinforcement Learning

However, the underlying process model is typically unknown and estimated by the limited historical real-world data. Here, we focus on Bayesian reinforcement learning (RL) with model risk quantified by the posterior distribution. We consider the *growing-batch RL* setting (Laroche and Tachet des Combes 2019). *The process consists in successive periods: In each p-th period, a batch of data is collected with a fixed policy from distributed complex bioprocess, it is used to update the knowledge of bioprocess state transition model, and then the policy will be updated for the next period.* At any p-th period, given all real-world historical data collected so far, denoted by $\mathscr{D}_p$, we construct the posterior distribution of state transition model quantifying the model risk, $p(\boldsymbol{\omega}|\mathscr{D}_p)\propto p(\mathscr{D}_p|\boldsymbol{\omega})p(\boldsymbol{\omega})$, where the prior $p(\boldsymbol{\omega})$ quantifies the existing knowledge on bioprocess dynamic mechanisms. Since the posterior of previous time period can be the prior for the next update, the posterior will be updated as new process data are collected. There are various *advantages of using the posterior distribution quantifying the model risk*, including: (1) it can incorporate the existing domain knowledge on bioprocess dynamic mechanisms; (2) it is valid even when the historical process data are very limited, which often happens in the biomanufacturing; and (3) it facilitates online learning and bioprocess knowledge automatic update.

At any p-th period, to provide the reliable guidance on the dynamic decision making, we need to consider both process inherent stochastic uncertainty and model risk. Let $\mu(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ denote the total expected reward accounting for both sources of uncertainty: $\mu(\boldsymbol{\pi}_{\boldsymbol{\theta}})\equiv\mathbb{E}_{\boldsymbol{\omega}\sim p(\boldsymbol{\omega}|\mathscr{D}_p)}\left[\mathbb{E}_{\boldsymbol{\tau}\sim D_{P_{\boldsymbol{\omega}}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{\tau})}\left[\sum_{t=1}^{H-1}\gamma^{t-1}r_t|\boldsymbol{\pi}_{\boldsymbol{\theta}},s_1,\boldsymbol{\omega}\right]\right]$, with the inner conditional expectation, $\widetilde{\mu}(\boldsymbol{\pi}_{\boldsymbol{\theta}};\boldsymbol{\omega})\equiv\mathbb{E}_{\boldsymbol{\tau}\sim D_{P_{\boldsymbol{\omega}}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{\tau})}\left[\sum_{t=1}^{H-1}\gamma^{t-1}r_t|\boldsymbol{\pi}_{\boldsymbol{\theta}},s_1,\boldsymbol{\omega}\right]$ accounting for stochastic uncertainty and the outer expectation accounting for model risk. Therefore, given the partial information

of bioprocess characterized by $p(\boldsymbol{\omega}|\mathscr{D}_p)$, we are interested in finding the optimal policy,

$$\boldsymbol{\pi}^{\star}_{\boldsymbol{\theta}}(\cdot|p(\boldsymbol{\omega}|\mathscr{D}_p)) = \arg\max_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} \mu(\boldsymbol{\pi}_{\boldsymbol{\theta}}) \equiv \arg\max_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} \mathbb{E}_{\boldsymbol{\omega}\sim p(\boldsymbol{\omega}|\mathscr{D}_p)} \left[\mathbb{E}_{\boldsymbol{\tau}\sim D^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{P_{\boldsymbol{\omega}}}(\boldsymbol{\tau})} \left[\sum_{t=1}^{H-1} \gamma^{t-1} r_t \middle| \boldsymbol{\pi}_{\boldsymbol{\theta}}, s_1, \boldsymbol{\omega} \right] \right]. \tag{3}$$

3 GREEN SIMULATION ASSISTED REINFORCEMENT LEARNING WITH MODEL RISK

In this section, we present the green simulation assisted Bayesian reinforcement learning, which can efficiently leverage the information from historical process trajectory data and accelerate the search for the optimal policy. In Section 3.1, at each p-th period and given real-world data $\mathscr{D}_p$, we derive the policy gradient solving the stochastic optimization problem (3) and develop the likelihood ratio based green simulation to improve the gradient estimation. Motivated by the metamodel study in Dong, Feng, and Nelson (2018), a decision process mixture proposal distribution and the likelihood ratio based metamodel for policy gradient are derived, which can reuse the process trajectories generated from previous experiments to improve the gradient estimation stability and speed up the search for the optimal policy. In Section 3.2, we provide the algorithm for proposed online green simulation assisted policy gradient with model risk.

3.1 Green Simulation Assisted Policy Gradient

At each p-th period and given real-world data $\mathscr{D}_p$, we develop the green simulation based likelihood ratio to efficiently use the existing process data and facilitate the policy gradient search. Conditional on the posterior distribution $p(\boldsymbol{\omega}|\mathscr{D}_p)$, the objective of reinforcement learning is to maximize the expected performance $\boldsymbol{\pi}^{\star}_{\boldsymbol{\theta}}(\cdot|p(\boldsymbol{\omega}|\mathscr{D}_p)) = \arg\max_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} \mu(\boldsymbol{\pi}_{\boldsymbol{\theta}})$. Based on eq. (3), we can rewrite the objective function,

$$\begin{aligned}
\mu(\boldsymbol{\pi}_{\boldsymbol{\theta}}) &= \mathbb{E}_{\boldsymbol{\omega}\sim p(\boldsymbol{\omega}|\mathscr{D}_p)} \left[\mathbb{E}_{\boldsymbol{\tau}\sim D^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{P_{\boldsymbol{\omega}}}(\boldsymbol{\tau})} \left[\sum_{t=1}^{H-1} \gamma^{t-1} r_t \middle| \boldsymbol{\pi}_{\boldsymbol{\theta}}, s_1, \boldsymbol{\omega} \right] \right] \\
&= \int\int \frac{p_{\boldsymbol{\omega}}(s_1)\prod_{t=1}^{H-1}\pi_{\boldsymbol{\theta}}(a_t|s_t)p_{\boldsymbol{\omega}}(s_{t+1}|s_t,a_t)}{p_{\bar{\boldsymbol{\omega}}}(s_1)\prod_{t=1}^{H-1}\pi_{\bar{\boldsymbol{\theta}}}(a_t|s_t)p_{\bar{\boldsymbol{\omega}}}(s_{t+1}|s_t,a_t)} p_{\bar{\boldsymbol{\omega}}}(s_1) \prod_{t=1}^{H-1}\pi_{\bar{\boldsymbol{\theta}}}(a_t|s_t)p_{\bar{\boldsymbol{\omega}}}(s_{t+1}|s_t,a_t) \sum_{t=1}^{H-1}\gamma^{t-1} r_t p(\boldsymbol{\omega}|\mathscr{D}_p) d\boldsymbol{\tau} d\boldsymbol{\omega} \\
&= \mathbb{E}_{\boldsymbol{\omega}\sim p(\boldsymbol{\omega}|\mathscr{D}_p)} \left[\mathbb{E}_{\boldsymbol{\tau}\sim D^{\boldsymbol{\pi}_{\bar{\boldsymbol{\theta}}}}_{P_{\bar{\boldsymbol{\omega}}}}(\boldsymbol{\tau})} \left[\frac{p_{\boldsymbol{\omega}}(s_1)\prod_{t=1}^{H-1}\pi_{\boldsymbol{\theta}}(a_t|s_t)p_{\boldsymbol{\omega}}(s_{t+1}|s_t,a_t)}{p_{\bar{\boldsymbol{\omega}}}(s_1)\prod_{t=1}^{H-1}\pi_{\bar{\boldsymbol{\theta}}}(a_t|s_t)p_{\bar{\boldsymbol{\omega}}}(s_{t+1}|s_t,a_t)} \sum_{t=1}^{H-1}\gamma^{t-1} r_t \middle| \boldsymbol{\pi}_{\boldsymbol{\theta}}, s_1, \boldsymbol{\omega} \right] \right]. \\
&= \mathbb{E}_{\boldsymbol{\omega}\sim p(\boldsymbol{\omega}|\mathscr{D}_p)} \left[\mathbb{E}_{\boldsymbol{\tau}\sim D^{\boldsymbol{\pi}_{\bar{\boldsymbol{\theta}}}}_{P_{\bar{\boldsymbol{\omega}}}}(\boldsymbol{\tau})} \left[\frac{D^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{P_{\boldsymbol{\omega}}}(\boldsymbol{\tau})}{D^{\boldsymbol{\pi}_{\bar{\boldsymbol{\theta}}}}_{P_{\bar{\boldsymbol{\omega}}}}(\boldsymbol{\tau})} \sum_{t=1}^{H-1}\gamma^{t-1} r_t \middle| \boldsymbol{\pi}_{\boldsymbol{\theta}}, s_1, \boldsymbol{\omega} \right] \right].
\end{aligned} \tag{4}$$

The likelihood ratio $D^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{P_{\boldsymbol{\omega}}}(\boldsymbol{\tau})/D^{\boldsymbol{\pi}_{\bar{\boldsymbol{\theta}}}}_{P_{\bar{\boldsymbol{\omega}}}}(\boldsymbol{\tau})$ in eq. (4) can adjust the existing trajectories generated by policy $\boldsymbol{\pi}_{\bar{\boldsymbol{\theta}}}$ and transition model $p(s_{t+1}|s_t,a_t;\bar{\boldsymbol{\omega}})$ to predict the mean response at the new policy $\mu(\boldsymbol{\pi}_{\boldsymbol{\theta}})$.

Let k denote the *accumulated* number of iterations for the optimal search occurring in the previous p periods. For notation simplification, suppose there is a fixed number of iterations in each period (say K). At k-th iteration, we only generate one posterior sample $\boldsymbol{\omega}_k \sim p(\boldsymbol{\omega}|\mathscr{D}_p)$ to estimate the outer expectation in eq. (4). For the candidate policy $\boldsymbol{\pi}_{\boldsymbol{\theta}_k}$, the likelihood ratio based green simulation is used to estimate the mean response $\mu(\boldsymbol{\pi}_{\boldsymbol{\theta}_k})$. It can reuse the process trajectories obtained from previous simulation experiments generated by using the policies and state transition models $(\boldsymbol{\pi}_{\boldsymbol{\theta}_i}, \boldsymbol{\omega}_i)$ with $i = 1,2,\ldots,k$. They are obtained in previous p periods with different posterior distributions, i.e., $p(\boldsymbol{\omega}|\mathscr{D}_\ell)$ with $\ell = 1,2,\ldots,p$. Then, since each proposal distribution is based on a single decision process distribution $D^{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}}_{P_{\boldsymbol{\omega}_i}}(\boldsymbol{\tau})$ specified by $(\boldsymbol{\pi}_{\boldsymbol{\theta}_i}, \boldsymbol{\omega}_i)$, we create the green simulation *individual likelihood ratio (ILR) estimator* of $\mu(\boldsymbol{\pi}_{\boldsymbol{\theta}_k})$,

$$\widehat{\mu}^{ILR}_{k,\mathbf{n}} \equiv \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\frac{D^{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}_{P_{\boldsymbol{\omega}_k}}(\boldsymbol{\tau}^{(i,j)})}{D^{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}}_{P_{\boldsymbol{\omega}_i}}(\boldsymbol{\tau}^{(i,j)})} \sum_{t=1}^{H_{ij}-1}\gamma^{t-1} r_t(a_t^{(i,j)}, s_t^{(i,j)}) \right], \ \boldsymbol{\tau}^{(i,j)} \overset{\text{i.i.d}}{\sim} D^{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}}_{P_{\boldsymbol{\omega}_i}}$$

where $\boldsymbol{\tau}^{(i,j)}$ is the j-th sample path generated by using $(\boldsymbol{\pi}_{\boldsymbol{\theta}_i}, \boldsymbol{\omega}_i)$ and $\mathbf{n} = (n_1, n_2, \ldots, n_k)$ is the combination of replications allocated at each $(\boldsymbol{\pi}_{\boldsymbol{\theta}_i}, \boldsymbol{\omega}_i)$ for $i = 1, 2, \ldots, k$. Since the process trajectory length is scenario-dependent, we replace the horizon H with H_{ij} to indicate its trajectory dependence.

This expected total reward estimator $\widehat{\mu}^{ILR}_{k,\mathbf{n}}$ can be used in the policy gradient to search for the optimal policy. Under some regularity conditions, we provide the derivation for the policy gradient estimator.

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}} \widetilde{\mu}(\boldsymbol{\pi}_{\boldsymbol{\theta}}; \boldsymbol{\omega}) &= \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\tau} \sim D^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{P_{\boldsymbol{\omega}}}} \left[\sum_{t=1}^{H-1} \gamma^{t-1} r_t | \boldsymbol{\pi}_{\boldsymbol{\theta}}, s_1, \boldsymbol{\omega} \right] = \int \nabla_{\boldsymbol{\theta}} D^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{P_{\boldsymbol{\omega}}}(\boldsymbol{\tau}) \left[\sum_{t=1}^{H-1} \gamma^{t-1} r_t(s_t, a_t) \right] d\boldsymbol{\tau} \\
&= \int D^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{P_{\boldsymbol{\omega}}}(\boldsymbol{\tau}) \nabla_{\boldsymbol{\theta}} \log(D^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{P_{\boldsymbol{\omega}}}(\boldsymbol{\tau})) \left[\sum_{t=1}^{H-1} \gamma^{t-1} r_t(s_t, a_t) \right] d\boldsymbol{\tau} \\
&= \int D^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{P_{\boldsymbol{\omega}}}(\boldsymbol{\tau}) \sum_{t=1}^{H-1} [\nabla_{\boldsymbol{\theta}} \log(\pi_{\boldsymbol{\theta}}(a_t|s_t)) + \nabla_{\boldsymbol{\theta}} \log(p(s_{t+1}|s_t, a_t))] \left[\sum_{t=1}^{H-1} \gamma^{t-1} r_t(s_t, a_t) \right] d\boldsymbol{\tau} \\
&= \int D^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{P_{\boldsymbol{\omega}}}(\boldsymbol{\tau}) \sum_{t=1}^{H-1} [\nabla_{\boldsymbol{\theta}} \log(\pi_{\boldsymbol{\theta}}(a_t|s_t))] \left[\sum_{t=1}^{H-1} \gamma^{t-1} r_t(s_t, a_t) \right] d\boldsymbol{\tau} \\
&= \mathbb{E}_{\boldsymbol{\tau} \sim D^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{P_{\boldsymbol{\omega}}}} \left[\sum_{t=1}^{H-1} \nabla_{\boldsymbol{\theta}} \log(\pi_{\boldsymbol{\theta}}(a_t|s_t)) \left[\sum_{t'=1}^{t-1} \gamma^{t'-1} r_{t'}(s_{t'}, a_{t'}) + \sum_{t'=t}^{H-1} \gamma^{t'-1} r_{t'}(s_{t'}, a_{t'}) \right] \middle| \boldsymbol{\pi}_{\boldsymbol{\theta}}, s_1, \boldsymbol{\omega} \right] \\
&= \sum_{t=1}^{H-1} \mathbb{E}_{\boldsymbol{\tau}_{[1:t-1]}} \left[\mathbb{E}_{\boldsymbol{\tau}_{[t:H-1]}} \left[\nabla_{\boldsymbol{\theta}} \log(\pi_{\boldsymbol{\theta}}(a_t|s_t)) \sum_{t'=1}^{t-1} \gamma^{t'-1} r_{t'}(s_{t'}, a_{t'}) \middle| \boldsymbol{\tau}_{[1:t-1]} \right] \middle| \boldsymbol{\pi}_{\boldsymbol{\theta}}, s_1, \boldsymbol{\omega} \right] \\
&\quad + \mathbb{E}_{\boldsymbol{\tau} \sim D^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{P_{\boldsymbol{\omega}}}} \left[\sum_{t=1}^{H-1} \nabla_{\boldsymbol{\theta}} \log(\pi_{\boldsymbol{\theta}}(a_t|s_t)) \sum_{t'=t}^{H-1} \gamma^{t'-1} r_{t'}(s_{t'}, a_{t'}) \middle| \boldsymbol{\pi}_{\boldsymbol{\theta}}, s_1, \boldsymbol{\omega} \right] \\
&= \mathbb{E}_{\boldsymbol{\tau} \sim D^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{P_{\boldsymbol{\omega}}}} \left[\sum_{t=1}^{H-1} \nabla_{\boldsymbol{\theta}} \log(\pi_{\boldsymbol{\theta}}(a_t|s_t)) \sum_{t'=t}^{H-1} \gamma^{t'-1} r_{t'}(s_{t'}, a_{t'}) \middle| \boldsymbol{\pi}_{\boldsymbol{\theta}}, s_1, \boldsymbol{\omega} \right] \qquad (5) \\
&= \mathbb{E}_{\boldsymbol{\tau} \sim D^{\boldsymbol{\pi}_{\bar{\boldsymbol{\theta}}}}_{P_{\bar{\boldsymbol{\omega}}}}} \left[\frac{D^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}_{P_{\boldsymbol{\omega}}}(\boldsymbol{\tau})}{D^{\boldsymbol{\pi}_{\bar{\boldsymbol{\theta}}}}_{P_{\bar{\boldsymbol{\omega}}}}(\boldsymbol{\tau})} \sum_{t=1}^{H-1} \nabla_{\boldsymbol{\theta}} \log(\pi_{\boldsymbol{\theta}}(a_t|s_t)) \sum_{t'=t}^{H-1} \gamma^{t'-1} r'_t(s_{t'}, a_{t'}) \middle| \boldsymbol{\pi}_{\boldsymbol{\theta}}, s_1, \boldsymbol{\omega} \right] \qquad (6)
\end{aligned}
$$

where eq. (6) holds due to similar derivation as eq. (4). Eq. (5) holds because

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{\tau}_{[1:t-1]}} \left[\mathbb{E}_{\boldsymbol{\tau}_{[t:H-1]}} \left[\nabla_{\boldsymbol{\theta}} \log(\pi_{\boldsymbol{\theta}}(a_t|s_t)) \sum_{t'=1}^{t-1} \gamma^{t'-1} r_{t'}(s_{t'}, a_{t'}) \middle| \boldsymbol{\tau}_{[1:t-1]} \right] \middle| \boldsymbol{\pi}_{\boldsymbol{\theta}}, s_1, \boldsymbol{\omega} \right] \\
&= \mathbb{E}_{\boldsymbol{\tau}_{[1:t-1]}} \left[\sum_{t'=1}^{t-1} \gamma^{t'-1} r_{t'}(s_{t'}, a_{t'}) \mathbb{E}_{\boldsymbol{\tau}_{[t:H-1]}} \left[\nabla_{\boldsymbol{\theta}} \log(\pi_{\boldsymbol{\theta}}(a_t|s_t)) | \boldsymbol{\tau}_{[1:t-1]} \right] \middle| \boldsymbol{\pi}_{\boldsymbol{\theta}}, s_1, \boldsymbol{\omega} \right] \qquad (7)
\end{aligned}
$$

where

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{\tau}_{[t:H-1]}} \left[\nabla_{\boldsymbol{\theta}} \log(\pi_{\boldsymbol{\theta}}(a_t|s_t)) | \boldsymbol{\tau}_{[1:t-1]} \right] \\
&= \prod_{t'=t+1}^{H-1} \int \pi_{\boldsymbol{\theta}}(a_{t'}|s_{t'}) p(s_{t'+1}|s_{t'}, a_{t'}) da_{t'} ds_{t'+1} \int \pi_{\boldsymbol{\theta}}(a_t|s_t) p(s_{t+1}|s_t, a_t) \nabla_{\boldsymbol{\theta}} \log(p(s_{t+1}|s_t, a_t) \pi_{\boldsymbol{\theta}}(a_t|s_t)) da_t ds_{t+1} \\
&= \int p(s_{t+1}, a_t|s_t) \nabla_{\theta} \log p(s_{t+1}, a_t|s_t) da_t ds_{t+1}, \text{ since } p(s_{t+1}, a_t|s_t) = \pi_{\boldsymbol{\theta}}(a_t|s_t) p(s_{t+1}|s_t, a_t) \\
&= \nabla_{\boldsymbol{\theta}} \int p(s_{t+1}, a_t|s_t) da_t ds_{t+1} = \nabla_{\boldsymbol{\theta}} 1 = 0.
\end{aligned}
$$

By plugging in eq.(6), the policy gradient becomes,

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mu(\boldsymbol{\pi}_{\boldsymbol{\theta}}) &= \nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{\omega}}\left[\mathbb{E}_{\boldsymbol{\tau}\sim D_{P_{\boldsymbol{\omega}}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{\tau})}\left[\sum_{t=1}^{H-1}\gamma^{t-1}r_t\middle|\boldsymbol{\pi}_{\boldsymbol{\theta}},s_1,\boldsymbol{\omega}\right]\right] = \nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{\omega}}[\widetilde{\mu}(\boldsymbol{\pi}_{\boldsymbol{\theta}};\boldsymbol{\omega})] = \mathbb{E}_{\boldsymbol{\omega}}[\nabla_{\boldsymbol{\theta}}\widetilde{\mu}(\boldsymbol{\pi}_{\boldsymbol{\theta}};\boldsymbol{\omega})] \\ &= \mathbb{E}_{\boldsymbol{\omega}}\left[\mathbb{E}_{\boldsymbol{\tau}\sim D_{P_{\boldsymbol{\omega}}}^{\boldsymbol{\pi}_{\bar{\boldsymbol{\theta}}}}}\left[\sum_{t=1}^{H-1}\nabla_{\boldsymbol{\theta}}\log(\pi_{\boldsymbol{\theta}}(a_t|s_t))\frac{D_{P_{\boldsymbol{\omega}}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{\tau})}{D_{P_{\boldsymbol{\omega}}}^{\boldsymbol{\pi}_{\bar{\boldsymbol{\theta}}}}(\boldsymbol{\tau})}\sum_{t'=t}^{H-1}\gamma'^{-1}r'_t(s_{t'},a_{t'})\right]\right]. \end{aligned} \tag{8}$$

Then, we obtain the *individual likelihood ratio based policy gradient estimator*,

$$\widehat{\nabla_{\boldsymbol{\theta}}\mu}_{k,\mathbf{n}}^{ILR} = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\sum_{t=1}^{H-1}\nabla_{\boldsymbol{\theta}}\log(\pi_{\boldsymbol{\theta}_k}(a_t^{(i,j)}|s_t^{(i,j)}))\frac{D_{P_{\boldsymbol{\omega}_k}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}(\boldsymbol{\tau}^{(i,j)})}{D_{P_{\boldsymbol{\omega}_i}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}}(\boldsymbol{\tau}^{(i,j)})}\sum_{t'=t}^{H-1}\gamma'^{-1}r'_t(a_{t'}^{(i,j)},s_{t'}^{(i,j)})\right]. \tag{9}$$

The importance weight or likelihood ratio $D_{P_{\boldsymbol{\omega}_k}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}(\boldsymbol{\tau})/D_{P_{\boldsymbol{\omega}_i}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}}(\boldsymbol{\tau})$ is larger for the trajectories $\boldsymbol{\tau}$ that are more likely to be generated by the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}_k}$ and transition probabilities $P_{\boldsymbol{\omega}_k}$. During the model learning process, the current policy candidate $\boldsymbol{\pi}_{\boldsymbol{\theta}_k}$ can be quite different from the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}_i}$ for $i=1,2,\ldots,k-1$ that generated the existing trajectories. Although this importance weight is unbiased, its variance could grow exponentially as the horizon H increases, which restricts their applications.

Since the likelihood ratio with single proposal distribution can lead to high estimation variance, inspired by the BLR-M metamodel proposed in Dong et al. (2018), we develop the bioprocess Mixture proposal distribution and Likelihood Ratio based policy gradient estimation (MLR), which allows us to selectively reuse the previous experiment trajectories and reduce the gradient estimation variance. Specifically, at the k-th iteration of search for optimal policy, we generate a posterior sample of process model parameters, $\boldsymbol{\omega}_k \sim p(\boldsymbol{\omega}|\mathscr{D}_p)$. During the optimal policy search, if there are new process data coming, the posterior will automatically update. The policy candidate $\boldsymbol{\pi}_{\boldsymbol{\theta}_k}$ and transition probability model $P(s_{t+1}|s_t,a_t;\boldsymbol{\omega}_k)$ uniquely define the trajectory distribution $D_{P_{\boldsymbol{\omega}_k}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}(\boldsymbol{\tau})$. *Based on the historical trajectories generated during the previous p periods, we create a mixture proposal distribution* $\sum_{i=1}^{k}\alpha_i^k D_{P_{\boldsymbol{\omega}_i}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}}(\boldsymbol{\tau})$*, and then use it to construct the likelihood ratio,*

$$f_k(\boldsymbol{\tau}|\bar{\boldsymbol{\theta}},\bar{\boldsymbol{\omega}}) \equiv \frac{D_{P_{\boldsymbol{\omega}_k}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}(\boldsymbol{\tau})}{\sum_{i=1}^{k}\alpha_i^k D_{P_{\boldsymbol{\omega}_i}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}}(\boldsymbol{\tau})} \tag{10}$$

where $\bar{\boldsymbol{\theta}}=(\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_k)$, $\bar{\boldsymbol{\omega}}=(\boldsymbol{\omega}_1,\ldots,\boldsymbol{\omega}_k)$, $\alpha_i^k=\frac{n_i}{\sum_{i=1}^{k}n_i}$, and n_i is the number of trajectories generated during the previous i-th iteration with $(\boldsymbol{\theta}_i,\boldsymbol{\omega}_i)$ for $i=1,\ldots,k$. By replacing the likelihood ratio $D_{P_{\boldsymbol{\omega}_k}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}(\boldsymbol{\tau})/D_{P_{\boldsymbol{\omega}_i}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}}(\boldsymbol{\tau})$ in eq. (9) with $f_k(\boldsymbol{\tau}|\bar{\boldsymbol{\theta}},\bar{\boldsymbol{\omega}})$, the green simulation based policy gradient estimator becomes,

$$\widehat{\nabla_{\boldsymbol{\theta}}\mu}_{k,\mathbf{n}}^{MLR} = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\sum_{t=1}^{H_{ij}-1}\nabla_{\boldsymbol{\theta}}\log(\pi_{\boldsymbol{\theta}_k}(a_t^{(i,j)}|s_t^{(i,j)}))f_k(\boldsymbol{\tau}^{(i,j)}|\bar{\boldsymbol{\theta}},\bar{\boldsymbol{\omega}})\sum_{t'=t}^{H_{ij}-1}\gamma'^{-1}r_{t'}(a_{t'}^{(i,j)},s_{t'}^{(i,j)})\right] \tag{11}$$

where $\boldsymbol{\tau}^{(i,j)} \overset{\text{i.i.d}}{\sim} D_{P_{\boldsymbol{\omega}_i}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}}(\boldsymbol{\tau})$ with $j=1,2,\ldots,n_i$ represent the trajectories generated in the previous i-th iteration. *Notice that the mixture proposal distribution based likelihoood ratio* $f_k(\boldsymbol{\tau}|\bar{\boldsymbol{\theta}},\bar{\boldsymbol{\omega}})$ *is bounded by* $1/\alpha_i^k$*. In this way, the mixture likelihood ratio puts higher weight on the existing trajectories that are more likely to be generated by* $D_{P_{\boldsymbol{\omega}_k}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}(\boldsymbol{\tau})$ *in the k-th iteration without assigning extremely large weights on the others.*

Since the parameterization plays an important role in the optimal policy gradient approach, we briefly discuss several possible policy functions. The policy function in reinforcement learning can be either stochastic or deterministic; see Silver et al. (2014) and Sutton and Barto (2018). The policy for discrete actions could be defined as the softmax function, $\pi_{\boldsymbol{\theta}}(a|s)=\frac{e^{\boldsymbol{\theta}^T\phi(s,a)}}{\sum_{a'\in\mathscr{A}}e^{\boldsymbol{\theta}^T\phi(s,a')}}$, where $\phi(s,a)\in\mathbb{R}^d$ is feature vector of state-action pair (s,a). The gradient of the policy function is $\nabla_{\boldsymbol{\theta}}\log(\pi_{\boldsymbol{\theta}}(a|s))=\phi(s,a)-\sum_{a'\in\mathscr{A}}\phi(s,a')\pi_{\boldsymbol{\theta}}(s,a')$. For continuous action spaces, we can apply Gaussian policy; say for example $\pi_{\boldsymbol{\theta}}(a|s)=\mathscr{N}(\boldsymbol{\theta}^T\phi(s),\sigma^2)$

for some constant σ, where $\phi(s)$ is feature representation of s. The gradient of the policy function is $\nabla_{\boldsymbol{\theta}} \log(\pi_{\boldsymbol{\theta}}(a|s)) = \nabla_{\boldsymbol{\theta}} \frac{-(a-\boldsymbol{\theta}^T\phi(s))^2}{2\sigma^2} = \frac{\boldsymbol{\theta}^T\phi(s)-a}{\sigma^2}\phi(s)$. In general, as long as the predictive models have a gradient descent learning algorithm, they can be applied in our approach, such as deep neural network, generalized linear regression, SVM, etc. In the empirical study, we considered a two-layer MLP model as our policy function.

3.2 Optimal Policy Search Algorithm

Algorithm 1 provides the procedure for the green simulation assisted policy gradient approach to support online learning and guide dynamic decision making.

Algorithm 1: Online Green Simulation Assisted Policy Gradient Policy with Model Risk

Input: the number of periods P for real-world dynamic data collection; the number of iterations K for optimal policy search in each period; differentiable policy $\pi_{\boldsymbol{\theta}}(a|s)$, $\forall a \in \mathscr{A}, s \in \mathscr{S}, \boldsymbol{\theta} \in \mathbb{R}^d$; and initial real-world data $\mathscr{D}_1$. Initialize the set of sample trajectories $\mathscr{E}_1$, the set of transition model parameters $\boldsymbol{\Omega}_1$, and the set of policy parameters $\boldsymbol{\Theta}_1$ to be empty set.

for $p = 1, 2, \ldots, P$ *(at each new real-world data collection point)* **do**

  for $k = (p-1)K+1, (p-1)K+2, \ldots, pK$ **do**

    1. Generate posterior samples $\boldsymbol{\omega}_k \sim p(\boldsymbol{\omega}|\mathscr{D}_p)$ and build the transition model with new parameter $\boldsymbol{\omega}_k$, i.e., $p(s_{t+1}|s_t, a_t, \boldsymbol{\omega}_k)$ for $t = 1, 2, \ldots, H-1$;

    2. Generate n_k trajectories by using the current policy $\pi_{\boldsymbol{\theta}_k}$ and model parameter $\boldsymbol{\omega}_k$;

    for $j = 1, 2, \ldots, n_k$ **do**

      (a) Generate j-th episode $\boldsymbol{\tau}^{(k,j)} = (s_1^{(k,j)}, a_1^{(k,j)}, s_2^{(k,j)}, a_2^{(k,j)}, \ldots, s_{H-1}^{(k,j)}, a_{H-1}^{(k,j)}, s_H^{(k,j)})$ of state-action sequence starting from initial state $s_1^{(k,j)} \sim p(s_1|\boldsymbol{\omega}_k)$, interacting with transition model $s_{t+1}^{(k,j)} \sim p(s_{t+1}|s_t^{(k,j)}, a_t^{(k,j)}; \boldsymbol{\omega}_k)$ and following policy $a_t^{(k,j)} \sim \pi_{\boldsymbol{\theta}_k}(a_t|s_t^{(k,j)})$ for stochastic policy or $a_t^{(k,j)} = \pi_{\boldsymbol{\theta}_k}(s_t^{(k,j)})$ for deterministic policy;

    end

    3. Reuse the trajectories generated in current and all previous iterations to improve the gradient estimation;

    for $i = 1, 2, \ldots, k$ *and* $j = 1, 2, \ldots, n_k$ **do**

      Construct the mixture proposal distribution based likelihood ratio, $f_k(\boldsymbol{\tau}^{(i,j)}|\bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\omega}})$, by using eq. (10).

    end

    4. Calculate the gradient $\widehat{\nabla_{\boldsymbol{\theta}} \mu}_{k,\mathbf{n}}^{MLR}$ based on eq. (11) and update the policy: $\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \eta_k \cdot \widehat{\nabla \mu}_{k,\mathbf{n}}^{MLR}$;

    5. Record new generated trajectories $\mathscr{E}_{k+1} = \mathscr{E}_k \cup \{\boldsymbol{\tau}^{(k,j)} | j = 1, 2, \ldots, n_k\}$, transition model parameters $\boldsymbol{\Omega}_{k+1} = \boldsymbol{\Omega}_k \cup \{\boldsymbol{\omega}_k\}$ and policy parameters $\boldsymbol{\Theta}_{k+1} = \boldsymbol{\Theta}_k \cup \{\boldsymbol{\theta}_k\}$;

  end

  6. Collect new process real-world data $\mathscr{L}_p$ by following the estimated optimal policy $\widehat{\pi}^{\star}_{\boldsymbol{\theta}_k}(a|s)$ from Step (4). Then, update the historical data set $\mathscr{D}_{p+1} = \mathscr{D}_p \cup \mathscr{L}_p$ and the posterior distribution $p(\boldsymbol{\omega}|\mathscr{D}_{p+1})$.

end

At any p-th period, given the real-world data $\mathscr{D}_p$ collected so far, the model risk is quantified by the posterior distribution $p(\boldsymbol{\omega}|\mathscr{D}_p)$, and then we apply the green simulation assisted policy gradient to search for the optimal policy in Steps (1)–(4). Specifically, in each k-th iteration, we first generate the posterior

sample for state transition probability model in Step (1), $\boldsymbol{\omega}_k \sim p(\boldsymbol{\omega}|\mathscr{D}_p)$, and then generate n_k trajectories by using the current policy $\pi_{\boldsymbol{\theta}_k}$ and model parameter $\boldsymbol{\omega}_k$ in Step (2). Then, in Steps (3) and (4), we reuse all historical trajectories and apply the green simulation-based policy gradient to speed up the search for the optimal policy. After that, as new real-world data coming, we update the posterior of transition model in Step (6), and then repeat the above procedure. In the empirical study, we use a fixed learning rate $\eta_k = 0.01$. *Notice that the proposed mixture likelihood ratio based policy gradient can be easily extended to broader reinforcement learning settings, such as online, offline, and model-free cases.*

4 EMPIRICAL STUDY

In this section, we study the performance of MLR using a biomanufacturing example. The upstream simulation model was built based on a first-principle model proposed by Jahic et al. (2002) and the downstream chromatography purification process follows Martagan et al. (2018). The empirical study results show that MLR outperforms the state-of-the-art policy search and baseline model-based stochastic gradient algorithms without BLR-M metamodel.

4.1 A Biomanufacturing Example

In this paper, we consider the batch-based biomanfucturing and use the stochastic simulation model built based on our previous study (Wang et al. 2019) to characterize the dynamic evolution of biomanufacturing process. A reinforcement learning model with continuous state and discrete action space is then constructed to search for the optimal decisions on chromatography pooling window, which was studied by Martagan et al. (2018). Instead of assuming that each chromatography step removes the uniformly distributed random proportion of protein and impurity (Martagan et al. 2018), we let the random removal fraction following Beta distribution with more realistic and flexible shape.

This biomanufacturing process consists of: (1) upstream fermentation where cells produce the target protein; and (2) downstream purification to remove the impurities through multiple chromatography steps. The primary output of fermentation is a mixture including the target protein and significant amount of unwanted impurity derived from the host cells or fermentation medium. After fermentation, each batch needs to be purified using chromatography to meet the specified quality requirements, i.e., purity concentration reaching to certain threshold level p_d. Since the chromatography typically contributes the main cost for downstream purification, in this paper, we focus on optimizing the integrated protein purification decisions related to chromatography operations or pooling window selection. To guide the downstream purification dynamic decision making, we formulate the reinforcement learning for biomanufacturing process as follows.
Decision Epoch: Following Martagan et al. (2018), we consider three-step chromatography. During chromatography we observe measurements and make decisions at each decision epoch $\mathscr{T} = \{t : 1,2,3\}$.
State Space: The state $\mathbf{s}_t$ at any decision time t is denoted by the protein-impurity-step tuple $\mathbf{s}_t \triangleq (p_t, i_t, t)$ on the finite space $\mathbf{P} \times \mathbf{I} \times \mathscr{T}$, where $\mathbf{P} \equiv [0, \bar{P}]$ and $\mathbf{I} \equiv [0, \bar{I}]$. The state space $\mathbf{P}$ is bounded by a predefined constant threshold $\bar{P}$ due to limitation in cell viability, growth rate and antibody production rate, etc. The state space $\mathbf{I}$ is bounded by a predefined constant threshold $\bar{I}$ following FDA process quality standards.
Action Space: Let a_t denote the selection of pooling window given the state $\boldsymbol{s}_t = (p_t, i_t, t)$ at the time $t \in \mathscr{T}$ following a policy $\pi_{\boldsymbol{\theta}}(\boldsymbol{s}_t)$. To simplify the problem, we consider 10 candidate pooling windows per chromatography step here.
Reward: At the end of downstream process, we record the reward,

$$r(p_t, i_t, t = 3) = \begin{cases} -c_f, & \text{if } r_t < r_d, \\ r(p_d), & \text{if } r_t \geq r_d, p_t \geq p_d, \\ r(p_t) - c_l(p_d - p_t), & \text{if } r_t \geq r_d, p_t \leq p_d. \end{cases}$$

We set the failure cost $c_f = \$48$, the protein shortage cost $c_l = \$6$ per milligram (mg), the product price \$5 per mg, $r(p_t) = \$5 \times p_t$, the amount of purity percentage requirement $p_d = 8$ mg, and the purity requirement

$r_d \geq 85\%$. The operational cost for each chromatography column is \$8 for $t \in 1,2,3$ and $r(p_t, i_t, t) = -\$8$ for $t \in \{1,2\}$.

Initial State: The random protein and impurity inputs for downstream chromatography are generated with the cell culture first-principle model, which is based on the differential equations with random noise. Here, we consider a fed batch bioreactor dynamic model proposed by Jahic et al. (2002),

$$\frac{dX}{dt} = (-\frac{F}{V} + \mu)X,\ \frac{dS}{dt} = \frac{F}{V}(S_i - S) - q_s X,\ P = \nu_1 X \text{ and } I = \nu_2 X \tag{12}$$

where $\nu_1 \sim \mathcal{N}(0.11, 0.01^2)$ and $\nu_2 \sim \mathcal{N}(0.11, 0.01^2)$ denote the constant specific mAb protein production and impurity rates, X denotes the biomass concentration from dry weight (gL^{-1}), $V = 1000$ is medium volume (L), $S_i \sim \mathcal{N}(780, 40)$ denotes inlet substrate concentration (gL^{-1}), S is substrate concentration (gL^{-1}), $q_{s,max} = 0.57$ is specific maximum rate of substrate consumption $(g\,g^{-1}h^{-1})$, $q_s = q_{s,max}\frac{S}{S+0.1}$ is the specific rate of substrate consumption $(g\,g^{-1}h^{-1})$, $\mu = (q_s - q_m) \cdot Y_{em}$ is the specific growth rate (h^{-1}) and $Y_{em} = 0.3$ is biomass yield coefficient exclusive maintenance and $q_m = 0.013$ is maintenance coefficient $(g\,g^{-1}h^{-1})$. The initial biomass and substrate is set to be $(0\,gL^{-1}, 40gL^{-1})$. We set the total time of production fermentation to be 50 days, and obtain $p^{(u)}$ mg of target protein and $i^{(u)}$ mg of impurity by applying the PDEs in (12). After the harvest, we further add the noise, following the normal distribution $\mathcal{N}(0, 5^2)$, to account for the overall impact from other factors introduced during the cell production process. Then, the protein p_1 and impurity i_1 inputs for downstream purification become $p_1 \sim \mathcal{N}(p^{(u)}, 5^2)$ and $i_1 \sim \mathcal{N}(i^{(u)}, 5^2)$. Therefore, in the empirical study, this PDE first-principle model based simulation is used to generate the random initial state or input $\mathbf{s}_1 = (p_1, i_1, 1)$ for downstream chromatography purification.

State Transitions: In each step of chromatography, the random proportions of protein and impurity will be removed, which depend on the selection of pooling window a_t. In specific, given a pooling window, each chromatography step removes random proportions of protein and impurity,

$$i_{t+1} = (\Psi_t | a_t) i_t \text{ and } p_{t+1} = (H_t | a_t) p_t,$$

where the fraction $\Psi_t | a_t \sim \text{Beta}(\psi_t^l | a_t, \psi_t^u | a_t)$ and $H_t | a_t \sim \text{Beta}(\eta_t^l | a_t, \eta_t^u | a_t)$ for all $a_t \in \mathscr{A}$ and $t \in \mathscr{T}$. We use the posterior distribution for model parameters $\psi_t^l | a_t, \psi_t^u | a_t$ $\eta_t^l | a_t$ and $\eta_t^u | a_t$ to quantify the model risk. Here we use a uniform prior $\text{Unif}(0, 300)$ for all parameters and generate the posterior samples based on MCMC using "PyMC3".

Policy: We use a 2-layer perceptron (MLP) of $D = 16$ dimensional first layer and 10 dimensional output layer with softmax activation function to parameterize our policy; see Section 11 in Hastie, Tibshirani, and Friedman (2001) for more discussion. For 10 pooling window outputs, there are 10 units T_ℓ with $\ell = 1, \ldots, 10$ at the second stage, with the ℓ-th unit modeling the probability of selecting action a_ℓ with $\ell = 1, \ldots, 10$. There are 10 pooling window candidate actions a_ℓ, $\ell = 1, 2, \ldots, 10$, each being coded as 0-1 variable. The derived feature Z_d depends on the linear combination of the input states $\mathbf{s}$, and then the output T_ℓ is modeled as a function of linear combinations of the Z_d,

$$\begin{aligned} Z_d &= \text{Sigmoid}(w_{0d} + \mathbf{w}_d^T \mathbf{s}), d = 1, \ldots, D, \\ T_\ell &= \beta_{0\ell} + \boldsymbol{\beta}_\ell^T \mathbf{Z}, \ell = 1, \ldots, 10 \\ \text{Prob}(a_\ell | \mathbf{s}) &\equiv \text{MLP}_\ell(\mathbf{s}) = g_\ell(\mathbf{T}), \ell = 1, \ldots, 10 \end{aligned} \tag{13}$$

where $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_D)$, $\mathbf{T} = (T_1, \ldots, T_{10})$, $\mathbf{w} = (w_{0d}, \mathbf{w}_d^T)$, $\boldsymbol{\beta} = (\beta_{0\ell}, \boldsymbol{\beta}_\ell^T)$. We can obtain the policy parameters $\boldsymbol{\theta} = (\mathbf{w}, \boldsymbol{\beta})$. The activation function is set to be sigmoid function, i.e., $\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$. The output function $g_\ell(T)$ allows a final transformation of the vector of outputs $\mathbf{T}$, which is set to be softmax functioon $g_\ell(\mathbf{T}) = \frac{e^{T_\ell}}{\sum_{\ell=1}^{10} e^{T_\ell}}$.

4.2 Study the Performance of Green Simulation Assisted Policy Gradient

In this section, we compare the performance of proposed green simulation assisted policy gradient with RL (MLR), individual likelihood ratio based policy gradient (ILR) and classical policy gradient (PG).

- Likelihood ratio based policy gradient with mixture proposal distribution (MLR): To reduce the computation complexity, instead of reusing all previous iterations, we introduce a rolling window parameter k_r to control how many historical trajectories we use,

$$\widehat{\nabla_{\boldsymbol{\theta}}\mu}_{k,\mathbf{n}}^{MLR} = \frac{1}{k_r}\sum_{i=k-k_r+1}^{k}\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\sum_{t=1}^{H_{ij}-1}\nabla_{\boldsymbol{\theta}}\log(\pi_{\boldsymbol{\theta}_k}(a_t^{(i,j)}|\mathbf{s}_t^{(i,j)}))f_k(\boldsymbol{\tau}^{(i,j)}|\bar{\boldsymbol{\theta}},\bar{\boldsymbol{\omega}})\sum_{t'=t}^{H_{ij}-1}\gamma^{t'-1}r_{t'}(a_{t'}^{(i,j)},\mathbf{s}_{t'}^{(i,j)})\right].$$

  In the empirical study, we use the most recent $k_r = 10$ iterations.
- Likelihood ratio based policy gradient with true transition model known (TLR),

$$\begin{aligned}\widehat{\nabla_{\boldsymbol{\theta}}\mu}_{k,\mathbf{n}}^{TLR} &= \frac{1}{k_r}\sum_{i=k-k_r+1}^{k}\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\sum_{t=1}^{H_{ij}-1}\nabla_{\boldsymbol{\theta}}\log(\pi_{\boldsymbol{\theta}_k}(a_t^{(i,j)}|\mathbf{s}_t^{(i,j)}))f_k(\boldsymbol{\tau}^{(i,j)}|\bar{\boldsymbol{\theta}},\boldsymbol{\omega}^c)\sum_{t'=t}^{H_{ij}-1}\gamma^{t'-1}r_{t'}(a_{t'}^{(i,j)},\mathbf{s}_{t'}^{(i,j)})\right]\\ &= \frac{1}{k_r}\sum_{i=k-k_r+1}^{k}\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\sum_{t=1}^{H_{ij}-1}\nabla_{\boldsymbol{\theta}}\log(\pi_{\boldsymbol{\theta}_k}(a_t^{(i,j)}|\mathbf{s}_t^{(i,j)}))\frac{\prod_{t=1}^{H-1}\pi_{\boldsymbol{\theta}_k}(a_t|\mathbf{s}_t)}{\sum_{i=1}^{k}\prod_{t=1}^{H-1}\pi_{\boldsymbol{\theta}_i}(a_t|\mathbf{s}_t)}\sum_{t'=t}^{H_{ij}-1}\gamma^{t'-1}r_{t'}(a_{t'}^{(i,j)},\mathbf{s}_{t'}^{(i,j)})\right],\end{aligned}$$

  where the last step holds because $\frac{p_{\boldsymbol{\omega}^c}(s_1)\prod_{t=1}^{H-1}\pi_{\boldsymbol{\theta}_k}(a_t|\mathbf{s}_t)p_{\boldsymbol{\omega}^c}(\mathbf{s}_{t+1}|\mathbf{s}_t,a_t)}{\sum_{i=1}^{k}p_{\boldsymbol{\omega}^c}(s_1)\prod_{t=1}^{H-1}\pi_{\boldsymbol{\theta}_i}(a_t|\mathbf{s}_t)p_{\boldsymbol{\omega}^c}(\mathbf{s}_{t+1}|\mathbf{s}_t,a_t)} = \frac{\prod_{t=1}^{H-1}\pi_{\boldsymbol{\theta}_k}(a_t|\mathbf{s}_t)}{\sum_{i=1}^{k}\prod_{t=1}^{H-1}\pi_{\boldsymbol{\theta}_i}(a_t|\mathbf{s}_t)}$.
- Individual likelihood ratio based policy gradient (ILR): It is obtained based on Equation (9),

$$\widehat{\nabla_{\boldsymbol{\theta}}\mu}_{k,\mathbf{n}}^{ILR} = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\sum_{t=1}^{H-1}\nabla_{\boldsymbol{\theta}}\log(\pi_{\boldsymbol{\theta}_k}(a_t|\mathbf{s}_t))\frac{D_{P_{\boldsymbol{\omega}_k}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}(\boldsymbol{\tau}^{(i,j)})}{D_{P_{\boldsymbol{\omega}_i}}^{\boldsymbol{\pi}_{\boldsymbol{\theta}_i}}(\boldsymbol{\tau}^{(i,j)})}\sum_{t'=t}^{H-1}\gamma^{t'-1}r_t'(a_{t'}^{(i,j)},\mathbf{s}_{t'}^{(i,j)})\right].$$

- Empirical policy gradient (PG): It uses the point estimator of state transition model parameter as the true one,

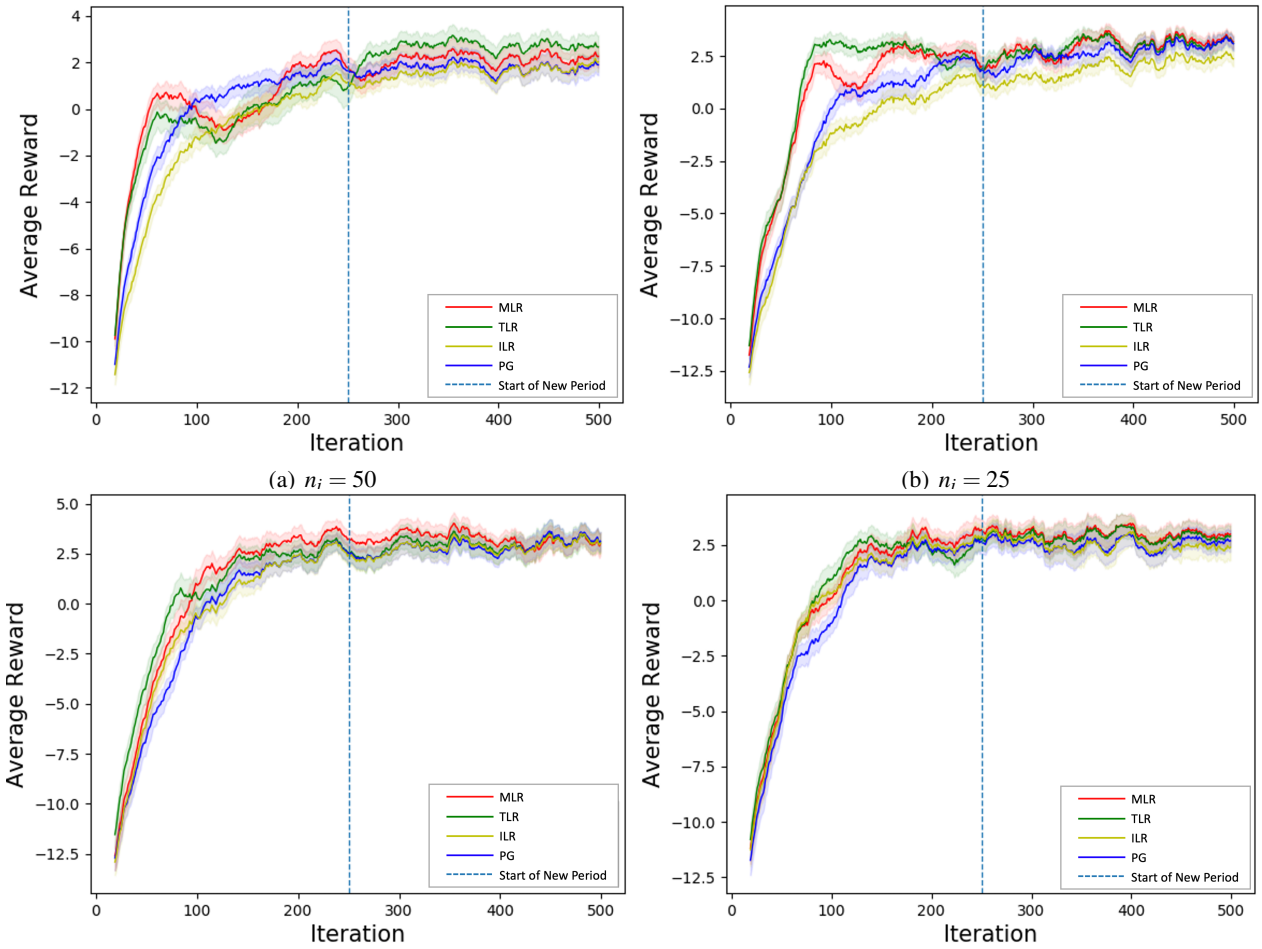
$$\widehat{\nabla_{\boldsymbol{\theta}}\mu}^{PG} = \frac{1}{n_i}\sum_{j=1}^{n_i}\left[\sum_{t=1}^{H-1}\nabla_{\boldsymbol{\theta}}\log(\pi_{\boldsymbol{\theta}}(a_t|\mathbf{s}_t))\sum_{t'=t}^{H-1}\gamma^{t'-1}r_t'(a_{t'}^{(i,j)},\mathbf{s}_{t'}^{(i,j)})\right].$$

Notice that in MLR, ILR, PG approaches, the underlying state transition model is unknown and estimated by finite real-world data. In TLR, we assume the model is known.

Here we set the amount of real-world data $m = 20$ for chromatography operation. Fig. 1 shows the convergence performance of MLR, TLR, ILR, and PG. The results are based on $M = 5$ macro replications. The x-axis represents the iteration index k, and the vertical dash line indicates the time when the new real-world process data are collected. Let $r_h(k)$ denote the average reward of the policy obtained from the k-th iteration in the h-th macro replications, which is estimated by running $r_{test} = 200$ trajectories with the true state transition model. The y-axis reports $\bar{r}(k) = \frac{1}{M}\sum_{h=1}^{M} r_h(k)$. We also plot the 95% confidence band for each approach, $[\bar{r}(k) - 1.96\times \mathrm{SE}(\bar{r}(k)), \bar{r}(k) + 1.96\times \mathrm{SE}(\bar{r}(k))]$, where $\mathrm{SE}(\bar{r}(k)) = \frac{1}{\sqrt{M(M-1)}}\sqrt{\sum_{h=1}^{M}(r_h(k)-\bar{r}(k))^2}$. Fig. 1 shows that MLR (red line) converges faster than PG and ILR. To better compare the performance of candidate algorithms, we apply the common random numbers (CRNs) for each macro replication.

From Fig 1, we can see the algorithms have already converged after 400 iterations. We compare the performance of policies obtained from MLR, PG and ILR based on the results from the last 100 iterations. We record the sample mean $\mu_a = \frac{1}{100}\sum_{k=401}^{500}\bar{r}(k)$ and standard error $\mathrm{SE} = \frac{1}{10}\sqrt{\frac{1}{99}\sum_{k=401}^{500}(\bar{r}(k)-\mu_a)^2}$ in Table 1. The results show that MLR tends to have better performance than both PG and ILR approaches.

When $n_i = 25$ based on $M = 5$ macro replications, the average runtime for MLR is 53.0 mins (12.6 mins for updating posterior distribution and 40.4 mins for policy search). The average runtime for PG is 34.3 mins (12.9 mins for updating posterior distribution and 21.4 mins for policy search). The average runtime for ILR is 50.1 mins (12.2 mins for updating posterior distribution and 37.9 mins for policy search).



(a) $n_i = 50$ (b) $n_i = 25$

(c) $n_i = 10$ (d) $n_i = 5$

Figure 1: Convergence results of MLR, TLR, ILR and PG.

Table 1: Average reward estimated based on last 100 iterations for MLR, TLR, ILR and PG.

	$n_i = 50$		$n_i = 25$		$n_i = 10$		$n_i = 5$	
	Mean	SE	Mean	SE	Mean	SE	Mean	SE
MLR	2.23	0.10	3.25	0.09	3.07	0.09	2.92	0.11
TLR	2.75	0.10	3.14	0.09	3.08	0.09	2.83	0.11
PG	1.80	0.09	3.04	0.10	3.10	0.10	2.53	0.11
ILR	1.83	0.10	2.36	0.10	3.01	0.10	2.39	0.13

5 CONCLUSIONS

We propose a green simulation assisted policy gradient algorithm. It can reduce the policy gradient estimation variance through selectively reusing the experiment data and automatically allocating more weight to those historical trajectories that are more likely generated by the stochastic decision process of interest. In addition, since we quantify the state transition probabilistic model risk with the posterior distribution, our model-based reinforcement learning can simultanesouly support online learning and guide dynamic decision making. Thus, the proposed approach is robust to model risk, and it can be applicable to various cases with different amounts of real-world data and process dynamic knowledge. In this paper, the empirical study of biomanufacturing example is used to illustrate that our approach can perform better than the state-of-art reinforcement learning and policy gradient approaches.

REFERENCES

Dong, J., M. B. Feng, and B. L. Nelson. 2018, Dec. "Unbiased Metamodeling via Likelihood Ratios". In *2018 Winter Simulation Conference (WSC)*, 1778–1789.

Feng, M., and J. Staum. 2017, October. "Green Simulation: Reusing the Output of Repeated Experiments". *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 27(4):23:1–23:28.

Hastie, T., R. Tibshirani, and J. Friedman. 2001. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Jahic, M., J. Rotticci-Mulder, M. Martinelle, K. Hult, and S.-O. Enfors. 2002. "Modeling of growth and energy metabolism of Pichia pastoris producing a fusion protein". *Bioprocess and Biosystems Engineering* 24(6):385–393.

Laroche, R., and R. Tachet des Combes. 2019, July. "Multi-batch Reinforcement Learning". In *The 4th Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM)*.

Martagan, T., A. Krishnamurthy, P. A. Leland, and C. T. Maravelias. 2018, January. "Performance Guarantees and Optimal Purification Decisions for Engineered Proteins". *Operations Research* 66(1):18–41.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. 2015, February. "Human-level control through deep reinforcement learning". *Nature* 518(7540):529–533.

Schaul, T., J. Quan, I. Antonoglou, and D. Silver. 2016. "Prioritized Experience Replay". *CoRR* abs/1511.05952.

Silver, D., G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. 2014, June. "Deterministic Policy Gradient Algorithms". In *Proceedings of the 31st International Conference on Machine Learning, ICML*. Beijing, China.

Sutton, R. S., and A. G. Barto. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book.

Wang, B., W. Xie, T. Martagan, A. Akcay, and C. G. Corlu. 2019. "Stochastic Simulation Model Development for Biopharmaceutical Production Process Risk Analysis and Stability Control". In *Proceedings of the 2019 Winter Simulation Conference*: IEEE, Inc.

AUTHOR BIOGRAPHIES

HUA ZHENG is Ph.D. student of the Department of Mechanical and Industrial Engineering (MIE) at Northeastern University. His research interests includes machine learning, data analytics, computer simulation and stochastic optimization. His email address is zheng.hua1@husky.neu.edu.

WEI XIE is an assistant professor in MIE at Northeastern University. She received her M.S. and Ph.D. in Industrial Engineering and Management Sciences (IEMS) at Northwestern University. Her research interests include interpretable Artificial Intelligence (AI), computer simulation, data analytics, stochastic optimization, and blockchain development for cyber-physical system risk management, learning, and automation. Her email address is w.xie@northeastern.edu. Her website is http://www1.coe.neu.edu/∼wxie/

BEN MINGBIN FENG is an assistant professor in actuarial science at the University of Waterloo. He earned his Ph.D. in IEMS at Northwestern University. His research interests include stochastic simulation design and analysis, optimization via simulation, nonlinear optimization, and financial and actuarial applications of simulation and optimization methodologies. His e-mail address is ben.feng@uwaterloo.ca. His website is http://www.math.uwaterloo.ca/∼mbfeng/.