Constrained Q-Learning for Batch Process Optimization

Elton Pan
Centre for Process Systems Engineering
Department of Chemical Engineering
Imperial College London
London, SW7 2BU
elton.pan17@imperial.ac.uk

Panagiotis Petsagkourakis
Centre for Process Systems Engineering
Department of Chemical Engineering
University College London
London, WC1E 7JE
p.petsagkourakis@ucl.ac.uk

Max Mowbray
Department of Chemical Engineering
and Analytical Science
University of Manchester
Manchester, M1 3AL
max.mowbray@manchester.ac.uk

Dongda Zhang
Department of Chemical Engineering
and Analytical Science
University of Manchester
Manchester, M1 3AL
dongda.zhang@manchester.ac.uk

Antonio del Rio-Chanona
Centre for Process Systems Engineering
Department of Chemical Engineering
Imperial College London
London, SW7 2BU
a.del-rio-chanona@imperial.ac.uk

Abstract

Chemical process optimization and control are often mired by the need to satisfy constraints for safe operation. Reinforcement learning (RL) has been shown to be a powerful control approach that can handle nonlinear stochastic optimal control problems. However, despite the promise exhibited, RL has yet to see marked translation to industrial practice primarily due to its inability to satisfy state constraints. In this work we aim to address this challenge. We propose an "oracle"-assisted constrained Q-learning algorithm that guarantees the satisfaction of joint chance constraints with a high probability, which is crucial for safety critical tasks. To achieve this, constraint tightening (backoffs) are introduced, which can be adjusted using Broyden's method, hence making them self-tuned. This results in a general methodology that can be imbued into approximate dynamic programming-based algorithms to ensure constraint satisfaction with high probability. Finally, we present case studies that analyze the performance of the proposed approach and compare this algorithm with model predictive control (MPC). The superior performance of this algorithm, in terms of constraint handling, signifies a step toward the incorporation of RL into real world optimization and control of systems, where constraints are essential in ensuring safety.

1 Introduction

The online optimization and control of chemical and biochemical processes, provides significant improvements in operative sustainability. Currently, the optimization of nonlinear stochastic processes

Challenges of Real-World Reinforcement Learning Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

poses a challenge for conventional control schemes given the requirement of an accurate process model and method to simultaneously handle process stochasticity and satisfy state and safety constraints. Recent works have explored the application of model-free reinforcement learning (RL) methods for online dynamic optimization of batch processes within the chemical and biochemical industries [1, 2]. Many of these works demonstrate the capability of RL algorithms to learn a control law independently of a nominal process model, but negate proper satisfaction of state and safety constraints [3]. In this work, we use constrained Q learning, a model-free algorithm to meet the operational and safety requirements of constraint satisfaction with high probability.

Despite the interest of the academic community in the application of RL for data-driven control, there exists relative inertia in practical and industrial implementation. Specifically, in the chemical and biochemical process industries, the development of methods to guarantee safe process operation and constraint satisfaction would enhance prospective deployment of RL-based systems [4]. The literature documents a number of approaches to constraint satisfaction, which typically either add penalty to the original reward function for constraint violation [5, 6] or augment the original MDP to take the form of a constrained MDP (CMDP) [7, 8]. The former approach introduces a number of hyperparmeters, which are typically chosen on the basis of heuristics and have bearing on policy optimality. This is also discussed in [9, 10]. The latter approach is underpinned by the learning of surrogate cost functions for each individual constraint combined with appropriate adaptation of the policy [9, 8] or value learning rule [11]. Other works include a Lyapunov-based approach proposed in [12], where a Lyapunov function is found and the unconstrained policy is projected to a safety layer. All the approaches above ensure constraint satisfaction only *in expectation*, which is insufficient for control and optimization of (bio)chemical processes. As most engineering systems are safety critical, satisfaction of constraints with high probability is a necessity [13].

To our knowledge, no method has been proposed which achieves such constraint satisfaction for pure action-value based methods. In this work, we propose a Q-learning method, which guarantees constraint satisfaction with high probability. Here, we learn an unconstrained actor and surrogate constraint action-value functions. We then subsequently construct a constrained actor action-value function as a superimposition of the unconstrained actor with the surrogate constraints. The constrained actor is iteratively tuned, as learning proceeds, via localised backoffs [14] to penalize constraint violation. Conceptually, backoffs provide a policy variant shaping mechanism to ensure high probability satisfaction [15]. Tuning comprises a Monte Carlo method to estimate the probability of constraint violation under the policy combined with Broyden's root finding method. The optimal greedy constrained policy is optimized through an evolutionary strategy [16] given its nonconvex nature. The work is arranged as follows; the problem description is formalised in section 2, the methodology proposed in section 3 and demonstrated empirically in section 4 via two benchmark case studies.

2 Reinforcement learning

2.1 RL in process engineering

Using RL directly on an industrial plant to construct an accurate controller would require prohibitive amounts of data. As such, process models must be used for the initial part of the training. The workflow shown in Fig. 1 starts with either a randomly initialized policy or a policy that is warm-started by an existing controller and apprenticeship learning [17]. Preliminary training is performed using closed-loop simulations from the offline process model. Here, the resulting control policy is a good approximation of the optimal policy, which is subsequently deployed in the real plant for further training online. Importantly, system stochasticity is accounted for and the controller will continue to adapt and learn to better control and optimize the process, hence addressing plant-model mismatch [18, 19].

2.2 Problem statement

We assume that the stochastic dynamic system in question follows a Markov process and transitions are given by

$$\mathbf{x}_{t+1} \sim p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{u}_t\right), \tag{1}$$

where $p(\mathbf{x}_{t+1})$ is the probability density function of future state $\mathbf{x}_{t+1}$ given a current state $\mathbf{x}_t \in \mathbb{R}^{n_x}$ and control $\mathbf{u}_t \in \mathbb{R}^{n_u}$ at discrete time t, and the initial state is given by $\mathbf{x}_0 \sim p_{\mathbf{x}_0}(\cdot)$. Without loss of

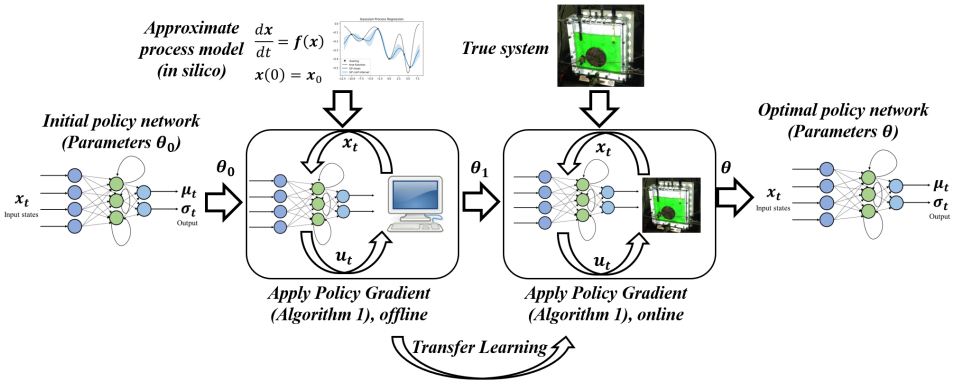

Figure 1: Schematic representation of RL for chemical process optimization

generality we can write Eq. (1) as:

$$\mathbf{x}_{t+1} = f\left(\mathbf{x}_t, \mathbf{u}_t, \mathbf{d}_t, \mathbf{p}\right), \tag{2}$$

where $\mathbf{p} \in \mathbb{R}^{n_p}$ are the uncertain parameters of the system and $\mathbf{d}_t \in \mathbb{R}^{n_d}$ are the stochastic disturbances. In this work, the goal is to maximize a predefined economic metric via an optimal policy subject to constraints. Consequently, this problem can be framed as an optimal control problem:

$$\pi(\cdot) := \begin{cases} \max_{\pi(\cdot)} \mathbb{E}\left\{J\left(\mathbf{x}_0, \ldots, \mathbf{x}_{t_f}, \mathbf{u}_0, \ldots, \mathbf{u}_{t_f}\right)\right\} \\ \text{s.t.} \\ \mathbf{x}_0 \sim p_{\mathbf{x}_0}\left(\mathbf{x}_0\right) \\ \mathbf{x}_{t+1} \sim p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{u}_t\right) \\ \mathbf{u}_t = \pi\left(\mathbf{x}_t\right) \\ \mathbf{u}_t \in \mathbb{U} \\ \mathbb{P}\left(\bigcap_{t=0}^{t_f}\left\{\mathbf{x}_t \in \mathbb{X}_t\right\}\right) = 1 - \omega \\ \forall t \in \{0, \ldots, t_f\} \end{cases} \tag{3}$$

where J is the objective function, $\mathbb{U}$ is the set of hard constraints for the controls and $\mathbb{X}_t$ denotes constraints for states that must be satisfied. In other words,

$$\mathbb{X}_t = \left\{\mathbf{x}_t \in \mathbb{R}^{n_x} \mid g_{j,t}\left(\mathbf{x}_t\right) \leq 0, j = 1, \ldots, n_g\right\}, \tag{4}$$

with n_g being the total number of constraints to be satisfied, and $g_{j,t}$ being the jth constraint to be satisfied at time t. Joint constraint satisfaction must occur at high probability of $1 - \omega$ where $\omega \in [0, 1]$. Herein, we present a Q-learning algorithm that allows to obtain the optimal policy which satisfies joint chance constraints.

3 Methodology

3.1 Oracle-assisted Constrained Q-learning

Q-learning, when unconstrained, may offer little practical utility in process optimization due to unbounded exploration by the RL agent. For instance, an unconstrained policy may often result in a thermal runaway leading to a safety hazard in the process. As such, herein constraints $g_{j,t}$ are incorporated through the use of an oracle $\hat{g}_{j,t}$ which is formulated as

$$\hat{g}_{j,t} = \max(g_{j,t'}), t' \geq t \tag{5}$$

with $g_{j,t}$ being the jth constraint to be satisfied at time t, and the oracle $\hat{g}_{j,t}$ is determined by the maximum level of violation to occur in all current and future time steps t' in the process realization.

The intuition behind this framework is as follows: Imagine a car (agent) accelerating towards the wall with the goal of minimizing the time it takes to reach some distance from the wall (objective) without actually crashing into the wall (constraint). Accelerating the car without foresight causes it to

go so fast that it cannot brake and stop in time, causing it to crash into the wall (constraint violated). As such, there is a need for foresight to ensure constraint satisfaction.

Effectively, the framework shown in Eq. (5) is aking to an oracle (or fortune-teller peeking into a crystal ball) advising the agent on the *worst* (or maximum) violation that a specific action can cause in the future given the current state. These values are easily obtained using Monte-Carlo simulations of the system. Analogous to a how a Q-function that gives the sum of all future rewards, the oracle provides the worst violation in all future states if a certain action is taken by the agent, hence imbuing in the agent a sense of foresight to avoid future constraint violation.

Similar to the Q-function, constraint values are represented by neural networks $G_{j,\theta}$ with state and action as input features. However, the subtle difference between the two is that the state representation of the input for $G_{j,\theta}$ involves time-to-termination $t_f - t$ instead of time t.

3.2 Constraint Tightening

To satisfy the constraints with high probability, it is required that the constraints are tightened with backoffs [20, 21] $b_{j,t}$ as:

$$\overline{\mathbb{X}}_t = \{\mathbf{x}_t \in \mathbb{R}^{n_x} \mid g_{j,t}(\mathbf{x}_t) + b_{j,t} \leq 0, j = 1, \ldots, n_g\} \tag{6}$$

where $b_{j,t}$ are the backoffs which tighten the former feasible set $\mathbb{X}_t$ stated in Eq. (4). The result of this would be the reduction of the perceived feasible space by the agent, which consequently allows for the satisfaction of constraints. Notice that the value of the backoffs necessarily imply a trade-off: large backoff values ensure constraint satisfaction, but renders the policy over-conservative hence sacrificing performance. Conversely, smaller backoff values afford solutions with higher rewards, but may not guarantee constraint satisfaction. Therefore, the values of $b_{j,t}$ are the minimum value needed to guarantee satisfaction of constraints.

To determine the desired backoffs, the cumulative distribution function (CDF) F of the oracle $\hat{g}_{j,t}$ is approximated using sample approximation (SSA) with S Monte Carlo (MC) simulations to give its empirical cumulative distribution function (ECDF) $\hat{F}_S$ where

$$\hat{F}_S(0) \approx F(0) = \mathbb{P}(\hat{g}_{j,t} \leq 0) \tag{7}$$

hence $\hat{F}_S(0)$ is the approximate probability for a trajectory to satisfy a constraint.

Subsequently, we adjust backoffs $b_{j,t}$ using Broyden's method in order to satisfy Eq. (8) to obtain the desired backoffs [22].

$$\hat{F}_S(0) \approx \mathbb{P}(\hat{g}_{j,t} \leq 0) = 1 - \omega \tag{8}$$

where ω is a tunable parameter depending on the case study, such that constraint satisfaction occurs with high probability $1 - \omega$ as shown in Eq. (3). Alternatively, the lower bound of the ECDF can be forced to be $1 - \omega$, and guarantee with confidence $1 - \epsilon$ that $\mathbb{P}(\hat{g}_{j,t} \leq 0) \geq 1 - \omega$. More technical details can be found in [13].

4 Case studies

4.1 Case study 1

This case study pertains to the photoproduction of phycocyanin synthesized by cyanobacterium *Arthrospira platensis*. Phycocyanin is a high-value bioproduct, and serves its biological role by increasing the photosynthetic efficiency of cyanobacteria and red algae. In addition, it is used as a natural colorant to substitute toxic synthetic pigments in cosmetic and food manufacturing. Moreover, it possesses antioxidant, and anti-inflammatory properties.

The dynamic system comprises a system of ODEs from [20] that describe the evolution of concentration (c) of biomass (x), nitrate (N) and product (q) under parametric uncertainty. The model is based on Monod kinetics, which describes the growth of microorganism in nutrient-sufficient cultures, where intracellular nutrient concentration is kept constant because of rapid replenishment. Here, a fixed volume fed-batch is assumed. The controls are light intensity ($u_1 = I$) and inflow rate ($u_2 = F_N$).

This case study and parameter values are adopted from [20]. Uncertainty in the system is two-fold: First, the initial concentration adopts a Gaussian distribution, where $[c_{x,0}, c_{N,0}] \sim \mathcal{N}([1.0, 150.0], \text{diag}(10^{-3}, 22.5))$ and $c_q(0) = 0$. Second, parametric uncertainty is assumed to be: $\frac{k_s}{(\mu mol/m^2/s)} \sim \mathcal{N}(178.9, \sigma^2_{k_s})$, $\frac{k_i}{(mg/L)} \sim \mathcal{N}(447.1, \sigma^2_{k_i})$, $\frac{k_N}{(\mu mol/m^2/s)} \sim \mathcal{N}(393.1, \sigma^2_{k_N})$ where the variance $\sigma_i^2 = 10\%$ of its corresponding mean value. This type of uncertainty is common in engineering settings, as the parameters are experimentally determined, and therefore subject to confidence intervals after being extracted using regression techniques. The objective function is to maximize the product concentration (c_q) at the end of the batch, hence the reward is defined as:

$$R_{t_f} = c_{q,t_f} \tag{9}$$

where t_f is the terminal time step. The two path constraints are as follows: Nitrate concentration (c_N) is to remain below 800 mg/L, and the ratio of bioproduct concentration (c_q) to biomass concentration (c_x) cannot exceed 11.0 mg/g for high density biomass cultivation. These constraints can be formulated as:

$$\begin{aligned} g_{1,t} &= c_N - 800 \leq 0 \quad \forall t \in \{0, \ldots, t_f\} \\ g_{2,t} &= c_q - 0.011 c_x \leq 0 \quad \forall t \in \{0, \ldots, t_f\} \end{aligned} \tag{10}$$

The control inputs are subject to hard constraints to be in the interval $0 \leq F_N \leq 40$ and $120 \leq I \leq 400$. The time horizon was set to 12 with an overall batch time of 240 h, and hence giving a sampling time of 20 h. The Q-network Q_θ consists 2 fully connected hidden layers, each consisting of 200 neurons with a leaky rectified linear unit (LeakyReLU) as activation function. The parameters used for training the agent are: $\epsilon = 0.99$, $b_{1,t} = -500$, $b_{2,t} = -0.05$, $s_\mathcal{D} = 3000$, $s_\mathcal{G} = 30000$, $M = 2000$, $N = 100$, $G = 100$, $H_1 = 500$, $H_2 = 1000$, $D_1 = 0.99$ and $D_2 = 0.995$.

Algorithm 1 Oracle-assisted constrained Q-learning

1. Initialize replay buffer $\mathcal{D}$ of size $s_\mathcal{D}$ and constraint buffers $\mathcal{G}_j$ of size $s_\mathcal{G}$, $j = 1, \ldots, n_g$
2. Initialize Q-network Q_θ and constraint networks $G_{j,\theta}$ with random weights, $j = 1, \ldots, n_g$
3. Initialize ϵ and backoffs $b_{j,t}$

for *training iteration = 1, …, M* **do**
  for *episode = 1, …, N* **do**
    Initialize state $\mathbf{x}_0 \sim p_{\mathbf{x}_0}(\mathbf{x}_0)$ and episode $\mathcal{E}$
    for *t = 0, …, t_f* **do**
      1. With probability ϵ select random control $\mathbf{u}_t$
      otherwise select $\mathbf{u}_t = \max_\mathbf{u} Q_\theta(\mathbf{x}_t, \mathbf{u}_t) \mid G_{j,\theta}(\mathbf{x}_t, \mathbf{u}_t) + b_{j,t} \leq 0, j = 1, \ldots, n_g$ (Sub-problem[a])
      2. Execute control $\mathbf{u}_t$ and observe reward R_t and new state $\mathbf{x}_{t+1}$
      3. Store transition ($\mathbf{x}_t$, $\mathbf{u}_t$, R_t, $\mathbf{x}_{t+1}$) in $\mathcal{E}$
    end
    1. Extract Q-values from $\mathcal{E}$ and store datapoint ($\mathbf{x}_t$, $\mathbf{u}_t$, Q_t) in $\mathcal{D}$
    2. Extract oracle-constraint values from $\mathcal{E}$ using: $\hat{g}_{j,t} = \max(g_{j,t'}), t' \geq t, j = 1, \ldots, n_g$
    3. Store datapoint ($\mathbf{x}_t$, $\mathbf{u}_t$, $\hat{g}_{j,t}$) in $\mathcal{G}_j, j = 1, \ldots, n_g$
  end
  1. Sample random minibatch of datapoints of size G ($\mathbf{x}_t$, $\mathbf{u}_t$, Q_t) from $\mathcal{D}$
  2. Sample random minibatch of datapoints of size H_j ($\mathbf{x}_t$, $\mathbf{u}_t$, $\hat{g}_{j,t}$) from $\mathcal{G}_j$
  3. Perform gradient descent on Q_θ and $G_{j,\theta}$ using Adam optimizer[b] with step size of 10^{-3}
  4. Decay ϵ using $\epsilon = D_1 \epsilon$
  5. Decay backoffs using $b_{j,t} = D_2 b_{j,t}$
end

Output: Optimal Q-network Q^*_θ and constraint networks $G_{j,\theta}, j = 1, \ldots, n_g$

[a] Sub-problem: An evolutionary algorithm is used to optimize the constrained Q-function using fitness function $f(\mathbf{u}) = Q_\theta(\mathbf{u}) + \sum_j C_j \min(0, -(G_{j,\theta}(\mathbf{u}) + b_{j,t}))$ where $g_{j,t}$ is the jth constraint violation at time t, and $b_{j,t}$ is the corresponding backoff. C_j are large values to ensure large negative fitness values for controls that lead to constraint violation.

[b] Any other full optimization step can be used here.

After completion of training using Algorithm 1, the backoffs are adjusted to satisfy Eq. (8), with backoffs at all time-steps t being constant. For simplicity, these backoffs are adjusted to ensure satisfaction of individual constraints, but it is worth noting that methods to satisfy joint chance

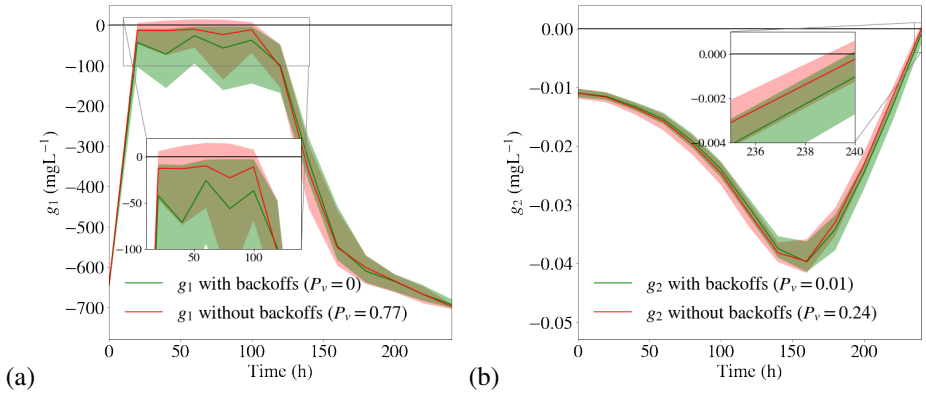

Figure 2: Case Study 1: Constraints $g_{1,t}$ (a) and $g_{2,t}$ (b) when backoffs are applied (green), and when they are absent (red) with probabilities of violation P_v within the parentheses. Inset: Zoomed-in region where violation of constraints occur. Shaded areas represent the 99th to 1st percentiles.

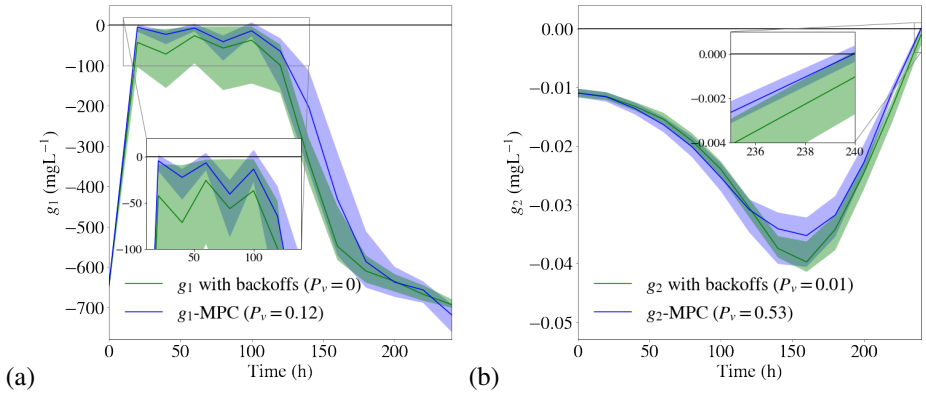

Figure 3: Case Study 1: Constraints $g_{1,t}$ (a) and $g_{2,t}$ (b) when backoffs are applied (green), and for MPC (blue) with probabilities of violation P_v within the parentheses. Inset: Zoomed-in region where violation of constraints occur. Shaded areas represent the 99th to 1st percentiles.

Table 1: Case Study 1: Comparison of probabilities of constraint violation P_v and objective values of different algorithms

Algorithm	Violation probability P_v	Objective (c_{q,t_f})
Oracle Q-learning with backoffs	**0.01**	0.166
Oracle Q-learning without backoffs	0.82	0.169
MPC	0.53	0.168

constraints can also be implemented as shown in [13] and [20]. The constraint satisfaction is shown in Fig. 2, where the shaded areas represent the 99th to 1st percentiles. Here, we elucidate the importance of applying backoffs to the policy: As shown in Fig. 2 (a), even though it may seem at face value that $g_{1,t}$ values for both methods are similar, the zoomed-in region (in the inset) clearly shows that oracle Q-learning without backoffs (red) results in a high probability of constraint violation ($P_v = 0.77$). The violation probabilities P_v in Fig. 2 and 3 correspond to the fraction of 400 MC trajectories that violate a certain constraint. Gratifying, when backoffs are applied (green) in Fig. 2 (a), all constraints are satisfied ($P_v = 0$).

In the same vein, in Fig. 2 (b), applying backoffs resulted in a drastic reduction of constraint violation from $P_v = 0.24$ to 0.01. This is expected since the backoffs are adjusted using the 99th percentile of $g_{j,t}$ values as shown in Eq. (8) where ω is set to 0.01. The objective value, represented by the final concentration of product c_q, are 0.166 and 0.169 for oracle Q-learning with and without backoffs, respectively. Consequently, this indicates that a small compromise in objective value can result in

high probability of constraint satisfaction, where violation probability is reduced from 0.82 to 0.01 (in boldface) upon applying backoffs as shown in Table 1.

In addition, the performance of the oracle Q-learning algorithm with backoffs has been compared with that of MPC, which is one of the main process control techniques used in chemical process optimization and hence serves as an important benchmark. Although MPC achieves a slightly higher objective value (Table 1), it fares poorly in terms of constraint satisfaction as shown in blue Fig. 3 (a) and (b) where probabilities of violation are 12 and 53 % for g_1 and g_2, respectively. This is unsurprising, since MPC is only able to satisfy constraints in *expectation*, which means that in a stochastic system, loosely speaking, violation occurs 50 % of the time. On the other hand, oracle Q-learning with backoffs violated a constraint only 1 % of the time (boldface in Table 1). Therefore, it is clear that this algorithm offers a more effective means of handling constraints compared to MPC.

4.2 Case study 2

The second case study involves a challenging semi-batch reactor adopted from [23], with the following chemical reactions in the reactor catalyzed by H_2SO_4:

$$2\mathrm{A} \xrightarrow[(1)]{k_{1\mathrm{A}}} \mathrm{B} \xrightarrow[(2)]{k_{2\mathrm{B}}} 3\mathrm{C} \tag{11}$$

Here, the reactions are first-order. Reactions (1) and (2) are exothermic and endothermic, respectively. The temperature is controlled by a cooling jacket. The controls are the flowrate of reactant A entering the reactor and the temperature of the cooling jacket T_0. Therefore, the state is represented by the concentrations of A, B, and C in mol/L (c_A, c_B, c_C), reactor temperature in K (T), and the reactor volume in L (Vol).

The objective function is to maximize the amount of product ($c_C \cdot Vol$) at the end of the batch. Two path constraints exist. Firstly, the reactor temperature needs to be below 420 K due to safety reasons and secondly, the reactor volume is required to be below the maximum reactor capacity of 800 L and therefore:

$$\begin{aligned} g_{1,t} &= T - 420 \leq 0 \quad \forall t \in \{0, \ldots, t_f\} \\ g_{2,t} &= Vol - 800 \leq 0 \quad \forall t \in \{0, \ldots, t_f\} \end{aligned} \tag{12}$$

The ODEs describing the evolution of the system can be found in [23]. The time horizon is fixed to 10 with an overall batch time of 4 h, therefore the sampling time is 0.4 h. Parametric uncertainty is set as: $\theta_1 \sim \mathcal{N}(4, 0.1)$, $A_2 \sim \mathcal{N}(0.08, 1.6 \times 10^{-4})$, $\theta_4 \sim \mathcal{N}(100, 5)$. The initial concentrations of A, B and C are set to zero. The initial reactor temperature and volume are 290 K and 100 L, respectively.

In this case study, due to its more challenging nature in terms of constraint satisfaction compared to the first case study, the backoffs have been adjusted to satisfy Eq. (8) using the 90th percentile $(g_{j,t})$ with $\omega = 0.1$ in Eq. (3). We observe that backoffs again proved to be necessary to ensure high probability of constraint satisfaction. From the inset of Fig. 4 (a), we can see that without backoffs the policy violates g_1 41% of the time, and this probability is reduced to 9% when backoffs are applied. The same applies for g_2 in Fig. 4 (b) where P_v is completely eliminated from 3 to 0% using backoffs.

Table 2: Case Study 2: Comparison of probabilities of constraint violation P_v and objective values of different algorithms

Algorithm	Violation probability (P_v)	Objective $(c_{C,t_f} \cdot Vol_{t_f})$
Oracle Q-learning with backoffs	**0.09**	532
Oracle Q-learning without backoffs	0.44	680
MPC	0.66	714

To compare the performance of MPC with oracle Q-learning with backoffs in the context of this case study, we consider two cases: First, in a deterministic system, MPC is found to be more efficient and gives solutions of much higher objective values. However, chemical systems are rarely deterministic in nature, hence limiting the applicability of MPC. Second, in the stochastic system, MPC often struggles in terms of constraint handling. This can be clearly seen in Fig. 5 (a), where the MPC trajectories only satisfy g_1 in expectation (blue line), hence resulting in high levels of violations

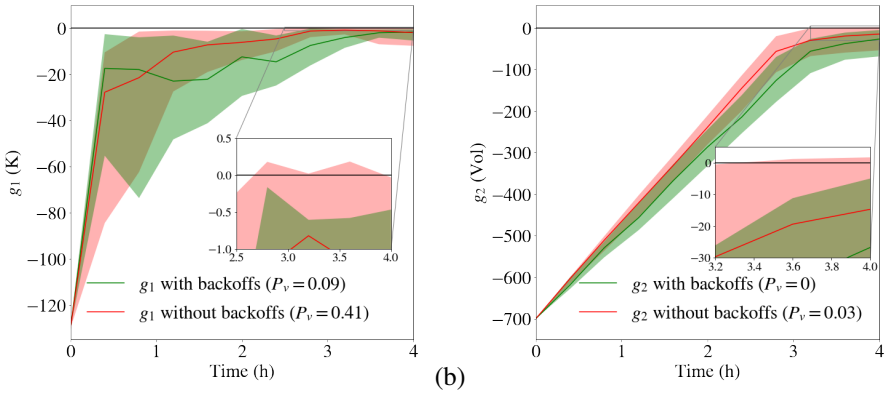

Figure 4: Case Study 2: Constraints $g_{1,t}$ (a) and $g_{2,t}$ (b) when backoffs are applied (green), and when they are absent (red) with probabilities of violation P_v within the parentheses. Inset: Zoomed-in region where violation of constraints occur. Shaded areas represent the 95th-5th percentiles for (a) and 99th-1st percentiles for (b).

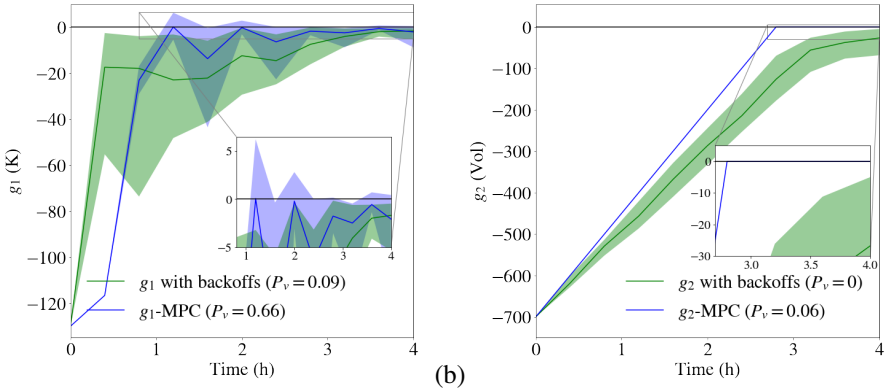

Figure 5: Case Study 2: Constraints $g_{1,t}$ (a) and $g_{2,t}$ (b) when backoffs are applied (green), and for MPC (blue) with probabilities of violation P_v within the parentheses. Inset: Zoomed-in region where violation of constraints occur. Shaded areas represent the 95th-5th percentiles for (a) and 99th-1st percentiles for (b).

(66%). Intriguingly, for g_2 the MPC trajectory in Fig. 5 displayed little variation, resulting in only small probability of violation (6%).

In terms of objective values, unlike the first case study, oracle Q-learning with backoffs saw a significant decrease in objective value in Table 2 after applying backoffs. This is expected because we further restrict the feasible space of the controller leading to a more conservative solution, hence exhibiting a trade-off between constraint satisfaction and objective value.

This trade-off is justified as the MPC solution results in 66% probability of constraint violation. In the context of a chemical plant, the MPC solution is unfeasible due to the high risk, for example, of a plant meltdown. The adoption of RL in such industries necessitates that these probabilities are minimized as safety is of utmost importance in chemical engineering.

Gratifyingly, it can been seen that the probability of constraint violation has been significantly improved from 66% (for MPC) to 9% (boldface in Table 2). Clearly, oracle Q-learning offers an effective means of not only satisfying constraints in expectation (green lines in Fig. 4), but more importantly with high probability (all green shaded areas below zero).

However, it is worth noting that this algorithm is based on Q-learning, which is expected to take longer time to train, particularly because it requires backoffs to be tuned. This is a direct consequence of shifting the computation time from online to offline. Indeed, such a tradeoff can be justified as this

guarantees robust constraint satisfaction *online* with fast computation time, which is crucial in many safety critical engineering applications.

5 Conclusions

In this paper we propose a new RL methodology for finding a controller policy that can satisfy constraints with high probability in stochastic and complex process systems. The proposed algorithm - oracle-assisted constrained Q-learning - uses constraint tightening by applying backoffs to the original feasible set. Backoffs restrict the perceived feasible space by the controller, hence allowing guarantees on the satisfaction of chance constraints. Here, we find the smallest backoffs (least conservative) that still guarantee the desired probability of satisfaction by solving a root-finding problem using Broyden's method. Results show that our proposed methodology compares favorably to model predictive control (MPC), a benchmark control technique commonly used in the industry, in terms of constraint handling. This is expected since MPC guarantees constraint satisfaction only in *expectation* (loosely speaking constraints are satisfied only 50% of the time), while our algorithm ensures constraint satisfaction with probabilities as high as 99% as shown in the case studies. Being able to solve constraint policy optimization problems with high probability constraint satisfaction has been one of the main hurdles of the widespread use of RL in engineering applications. The promising performance of this algorithm is an encouraging step towards applying RL to the real world, where constraints on policies are absolutely critical due to safety reasons.

References

[1] Vikas Singh and Hariprasad Kodamana. "Reinforcement learning based control of batch polymerisation processes". In: *IFAC-PapersOnLine* 53.1 (2020), pp. 667–672.

[2] Panagiotis Petsagkourakis et al. "Reinforcement learning for batch bioprocess optimization". In: *Computers & Chemical Engineering* 133 (2020), p. 106649.

[3] Neythen J Treloar et al. "Deep reinforcement learning for the control of microbial co-cultures in bioreactors". In: *PLOS Computational Biology* 16.4 (2020), e1007783.

[4] Joohyun Shin et al. "Reinforcement Learning–Overview of recent progress and implications for process control". In: *Computers & Chemical Engineering* 127 (2019), pp. 282–294.

[5] Jong Min Lee and Jay H Lee. "Approximate dynamic programming-based approaches for input–output data-driven control of nonlinear processes". In: *Automatica* 41.7 (2005), pp. 1281–1288.

[6] Chen Tessler, Daniel J Mankowitz, and Shie Mannor. "Reward constrained policy optimization". In: *arXiv preprint arXiv:1805.11074* (2018).

[7] Eitan Altman. *Constrained Markov decision processes*. Vol. 7. CRC Press, 1999.

[8] Yongshuai Liu, Jiaxin Ding, and Xin Liu. *IPO: Interior-point Policy Optimization under Constraints*. 2019. arXiv: 1910.09615. URL: http://arxiv.org/abs/1910.09615.

[9] Joshua Achiam et al. "Constrained policy optimization". In: *arXiv preprint arXiv:1705.10528* (2017).

[10] Logan Engstrom et al. "Implementation Matters in Deep Policy Gradients: A Case Study on PPO and TRPO". In: *arXiv preprint arXiv:2005.12729* (2020).

[11] Yangyang Ge et al. "Safe Q-Learning Method Based on Constrained Markov Decision Processes". In: *IEEE Access* 7 (2019), pp. 165007–165017.

[12] Yinlam Chow et al. *Lyapunov-based Safe Policy Optimization for Continuous Control*. 2019. arXiv: 1901.10031. URL: http://arxiv.org/abs/1901.10031.

[13] Panagiotis Petsagkourakis et al. "Chance Constrained Policy Optimization for Process Control and Optimization". In: *arXiv preprint arXiv:2008.00030* (2020).

[14] Joel A Paulson and Ali Mesbah. "Nonlinear model predictive control with explicit backoffs for stochastic systems under arbitrary uncertainty". In: *IFAC-PapersOnLine* 51.20 (2018), pp. 523–534.

[15] Andrew Y Ng, Daishi Harada, and Stuart Russell. "Policy invariance under reward transformations: Theory and application to reward shaping". In: *ICML*. Vol. 99. 1999, pp. 278–287.

[16] Adam Slowik and Halina Kwasnicka. "Evolutionary algorithms and their applications to engineering problems". In: *Neural Computing and Applications* (2020), pp. 1–17.

[17] Pieter Abbeel and Andrew Y Ng. "Apprenticeship learning via inverse reinforcement learning". In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 1.

[18] Steven Spielberg et al. "Toward self-driving processes: A deep reinforcement learning approach to control". In: *AIChE Journal* 65.10 (2019), e16689.

[19] Zhi Wang, Han-Xiong Li, and Chunlin Chen. "Incremental Reinforcement Learning in Continuous Spaces via Policy Relaxation and Importance Weighting". In: *IEEE Transactions on Neural Networks and Learning Systems* (2019).

[20] Eric Bradford et al. "Stochastic data-driven model predictive control using Gaussian processes". In: *Computers & Chemical Engineering* 139 (2020), p. 106844.

[21] Mina Rafiei and Luis A Ricardez-Sandoval. "Stochastic back-off approach for integration of design and control under uncertainty". In: *Industrial & Engineering Chemistry Research* 57.12 (2018), pp. 4351–4365.

[22] Carl T Kelley. *Iterative methods for linear and nonlinear equations*. SIAM, 1995.

[23] Eric Bradford and Lars Imsland. "Economic stochastic model predictive control using the unscented kalman filter". In: *IFAC-PapersOnLine* 51.18 (2018), pp. 417–422.