IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 5, 2019, accepted November 4, 2019, date of publication November 11, 2019, date of current version November 22, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2952651*

# Safe Q-Learning Method Based on Constrained Markov Decision Processes

**YANGYANG GE[1], FEI ZHU[1,2], XINGHONG LING[1], AND QUAN LIU[1]**

[1]School of Computer Science and Technology, Soochow University, Suzhou 215006, China
[2]Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006, China

Corresponding author: Fei Zhu (zhufei@suda.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61303108 and Grant 61772355, in part by the Natural Science Foundation of Jiangsu Higher Education Institutions of China under Grant 17KJA520004, in part by the Suzhou Key Industries Technological Innovation-Prospective Applied Research Project under Grant SYG201804, in part by the Program of the Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, under Grant KJS1524, and in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

**ABSTRACT** The application of reinforcement learning in industrial fields makes the safety problem of the agent a research hotspot. Traditional methods mainly alter the objective function and the exploration process of the agent to address the safety problem. Those methods, however, can hardly prevent the agent from falling into dangerous states because most of the methods ignore the damage caused by unsafe states. As a result, most solutions are not satisfactory. In order to solve the aforementioned problem, we come forward with a safe Q-learning method that is based on constrained Markov decision processes, adding safety constraints as prerequisites to the model, which improves standard Q-learning algorithm so that the proposed algorithm seeks for the optimal solution ensuring that the safety premise is satisfied. During the process of finding the solution in form of the optimal state-action value, the feasible space of the agent is limited to the safe space that guarantees the safety via the feasible space being filtered by constraints added to the action space. Because the traditional solution methods are not applicable to the safe Q-learning model as they tend to obtain local optimal solution, we take advantage of the Lagrange multiplier method to solve the optimal action that can be performed in the current state based on the premise of linearizing constraint functions, which not only improves the efficiency and accuracy of the algorithm, but also guarantees to obtain the global optimal solution. The experiments verify the effectiveness of the algorithm.

**INDEX TERMS** Constrained Markov decision processes, safe reinforcement learning, Q-learning, constraint, Lagrange multiplier.

## I. INTRODUCTION

Reinforcement learning (RL) aims to solve the problem with maximizing its long-term reward returned by executing actions. Given a specific state-action pair, the agent of reinforcement learning is able to learn the optimal policy by interacting with the dynamic environment through trial and error [1]. Although reinforcement learning algorithms have long been studied, most of them were initially designed for toy environments [2], [3]. Recently some reinforcement learning algorithms were gradually applied to the industrial fields, e.g. automatic driving [4], where the safety of the agent is particularly important [5]. As a result, the goal of the agent is not only to maximize the long-term reward, but also guarantee that the agent always stays away from unsafe states, which means the agent should seek for optimal solution under safety constraints to avoid unnecessary losses. However, traditional reinforcement learning methods are incompatible with the above cases unless dangerous operations are completely eliminated and the safety of the agent is ensured at the initial time of deployment.

At present, there are two main methods to solve safety problems for the agent: changing the objective function and improving the exploration process [6]. The method of changing the objective function obtains the safety policy for the agent by changing the optimization criterion. In some cases, this method may reduce the probability of the agent entering dangerous state, but it does not fundamentally solve the safety problem of the agent, and the agent may still enter dangerous state causing damage to itself and increasing the cost of

The associate editor coordinating the review of this manuscript and approving it for publication was Xiping Hu.

seeking the optimal policy [7]. The method of improving exploration process is to obtain the knowledge of tasks by random exploration of state space and action space. Only when enough information is gathered from the environment can the algorithm's performance be improved [8]. Driessens et al. proposed that the agent can convert the exploring process into Boltzmann exploring or completely greedy exploring according to the predicted value of the initial training stage [9]. In this way, the agent is exposed to the most relevant areas of the state space and action space from the initial stage of the learning process, thus reducing the time required for random exploration to discover these areas. However, this method needs more additional information. Moreover, it doesn't fundamentally solve the safety problems of the agent, and the agent may still enter a dangerous state because it is difficult for the agent to obtain all undesired states in a risky environment.

In recent years, researchers have applied constrained reinforcement learning methods to system safety and other related issues [10]. Meanwhile, the reinforcement learning methods based on the constrained model is only a preliminary attempt and the solution is not satisfactory. There is no systematic description of solving the problem, especially description of effectively solving large-scale continuous space tasks.

In reinforcement learning, although dynamic programming method is able to solve problems ensuring the safety of the agent to some extent, it needs a perfect environment model, namely, a thorough understanding of the environment in advance [11]. As in most cases of control tasks, prior knowledge is unavailable, making dynamic programming incapable of dealing with model-free control problems. Safe Q-learning, however, can solve model-free problem for Q-learning is an efficient model-free reinforcement algorithm.

Aiming at the safety problem of the agent in reinforcement learning, we propose a method of safe Q-learning (SQL) based on constrained Markov decision processes (CMDPS) [12]. Our safe Q-learning algorithm adds multi-dimensional constraint function to the original objective function so that it divides the state space into feasible state space that satisfies the constraint condition, as well as infeasible state space that doesn't meet the constraint condition. Similarly, the action space is also divided into feasible action space and infeasible action space. Under constraint functions, the agent only performs feasible actions and enters into feasible states so as to avoid unnecessary damage caused by incorrectly performing the infeasible action and entering the infeasible state.

This paper is organized as follows. In Section 2, we introduce the related concepts and studies. In Section 3, we formalize our model and transform the model into a convex model by linearizing constraint functions, where we exploit the Lagrange multiplier method to obtain the optimal action that the agent can take in the current state. In Section 4, we compare safe Q-learning with standard reinforcement learning algorithms. In Section 5, a summary of the safe Q-learning and a discussion of future work are concluded.

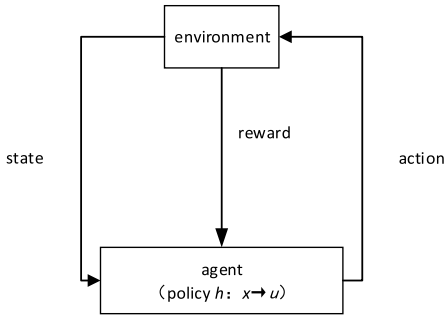


**FIGURE 1.** The illustration of reinforcement learning.

## II. PRELIMINARIES

### A. REINFORCEMENT LEARNING

In the reinforcement learning task, the agent perceives the environment with the purpose of obtaining maximum long-term cumulative return, a valuable reward signal [13]. Markov decision process (MDP) is the framework for solving reinforcement learning problems, described by quaternion $(X, U, P, R)$ [14], where $X$ is the state space set, $U$ is the action space set, $P$ is the state transition probability and $R$ is the reward function, represented as $r_{t+1} = R(x_t, u_t, x_{t+1})$ showing the immediate reward $r_{t+1}$ the agent receives after taking action $u_t$ in state $x_t$ and moving to next state $x_{t+1}$. The state transition probability [15] is as:

$$P^{u}_{xx'} = P\left[X_{t+1} = x' | X_t = x, U_t = u\right] \tag{1}$$

At time $t$, the agent is in the state $x_t$, and the action $u_t$ is selected according to the policy $h$, represented as $u_t \sim h(x_t)$. The environment returns the agent a feedback according to the action performed so that the agent is able to enter the next state and get the reward. The model of reinforcement learning is shown in Figure 1.

The state-action value function $Q^h(x, u)$ is used to evaluate the quality of policies in reinforcement learning, it refers to the sum of cumulative rewards that an agent obtains when performing action $u_t$ which is taken under the guidance of policy $h$ in the current state $x_t$. The state-action value function [16] is as:

$$Q^h(x, u) = E\left[R_t | x_t = x, u_t = u, h\right] \tag{2}$$

where $E$ is the expectation and $R_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$ is the sum of the accumulated discounted rewards of the agent from the time $t$ to the time $T$.

With the increasing of iteration steps, the state-action value converges to the optimal value. The optimal state-action value [17] is as:

$$Q^*(x, u) = \max_h E\left[R_t | x_t = x, u_t = u, h\right] \tag{3}$$

Q-learning algorithm [18] is a famous and widely used off-policy algorithm that introduced the concept of temporal

difference error (TD-error) [19], by which Q-value function is updated iteratively and finally converges to the optimal state-action value. The TD-error and update process of Q-value are as:

$$\delta_t = r_{t+1} + \gamma \max_{u'} Q_t\left(x_{t+1}, u'\right) - Q_t\left(x_t, u_t\right) \quad (4)$$

$$Q_{t+1}\left(x_t, u_t\right) = Q_t\left(x_t, u_t\right) + \alpha_t \delta_t \quad (5)$$

where $t$ represents time step, $\alpha_t$ is the learning rate, $\delta_t$ is the TD-error, and $u'$ is the action that is to be performed in the next state $x_{t+1}$.

The optimal control policy is obtained by iterating continuously using Behrman equation, and the state-action value finally converges [2], [20]. Q-learning algorithm is one of the most widely used algorithms with fast convergence speed and can obtain the maximum reward while executing the action [21].

### B. CONSTRAINED MARKOV DECISION PROCESSES

Constrained Markov decision processes can be modeled by a five-tuple $(X, U, P, R, C)$, where $X$ is a set of state space containing a limited number of states, $U$ is a set of action space consisting of a finite series of actions, $P$ is the transition probability from one state to another, $R$ is the reward function about instant reward and $C$ is the set of constraint functions. The purpose of constrained Markov decision processes is to maximize the reward function under prerequisites of satisfying constraints. The set of constraint functions is specifically expressed as:

$$C = \{c_i : X \times U \to \mathbb{R} | i = 1 \cdots k\} \quad (6)$$

where $k$ is a constant.

Constrained Markov decision process is based on the constrained Markov chain that refers to a constrained process where the state space is constrained, and the state space is a set of space with finite elements. In discrete space, the mathematical model of constrained Markov chains is as Eq.7 [22]:

$$\begin{aligned} &\limsup_{k\to\infty} \frac{1}{k} \sum_{i=0}^{k-1} E[f(X_i, U_i)] \\ &s.t.\ \limsup_{k\to\infty} \frac{1}{k} \sum_{i=0}^{k-1} E[g(X_i, U_i)] \le \beta \end{aligned} \quad (7)$$

where $\beta > 0$ is a scalar, the objective function $f$ and the constraint function $g$ are defined as: $X \times U \to \mathbb{R}$. The upper limit operation is taken for the objective function and the constraint function in the model, because the maximum value of the objective function is required, and the left side of the constraint function is not greater than the right side, otherwise the lower limit operation needs to be taken down.

In previous work, some researchers combined constrained Markov decision processes with reinforcement learning to solve limitation and safety problem of the agent in practical applications. Bušić et al. proposed action-constraints Markov decision processes with Kullback-Leibler (KL) cost and its main idea is to solve a complete parameter cluster of Markov decision processes [23]. Reinforcement learning algorithm based on constrained Markov decision process can solve the limitation and safety problem of the agent in the process of exploration. Borkar et al.'s work of actor-critic algorithm based on constrained Markov decision processes [24] provided theoretical support for reinforcement learning based on constrained Markov decision processes; however, they didn't carry out experimental verification. Achiam et al. put forward constrained policy optimization (CPO); however, their work only approximately satisfied constraints [25]. Wen et al. developed a constrained cross-entropy-based method to solve the safety issues for agents; however, it ignored the feasibility of initial policies with both Markovian and non-Markovian objective functions and constraint functions [26].

In general, the safe Q-learning belongs to constrained RL methods that have models with limited resources or minimum cost, multi-objective models, and limited speed for agents and so on.

### C. LAGRANGE MULTIPLIER

Reinforcement learning methods are to make the agent performs the optimal action by maximizing the objective function, which is oversimplified and neglects the agent's safety and other limitations. Adding constraint functions to the objective function can ensure the safety of the agent in the process of exploration, nevertheless original reinforcement learning methods are no longer suitable.

Optimization problem with equality constraints can be solved by using Lagrange multiplier and the one with inequality constraints can be solved by exploiting Lagrange multiplier and Karush-Kuhn-Tucker (KTT) conditions which are necessary and sufficient condition when the model is convex and determine whether the solution obtained by Lagrange multiplier method is optimal [27]. The general form of constrained optimization model is represented by Eq.8, the objective function and the constraint function are differentiable in Eq.8 [28].

$$\begin{aligned} &\max f(y) \\ &s.t.\ c_i(y) = C_i, \quad i = 1, 2, \cdots k' \\ &\quad c_i'(y) \le C_i', \quad i = k'+1, \cdots, k \\ &\quad y \in \mathbb{R}^n \end{aligned} \quad (8)$$

where formulas $c_i(y) = C_i$ and $c_i'(y) \le C_i'$ are the abstract representation of constraint functions, $\gamma$ is the discount factor, $0 < \gamma < 1$, $k$ is a constant and represents the number of constraint functions, $k'$ is a constant and represents the number of equality constraint functions, and the number of inequality constrained functions is $k - k'$, $y$ is the independent variable which is an n-dimensional vector. The above model is abstract, and the concrete form is given by the specific task. If the model requires to minimize the objective function, the process of minimizing the objective function can be changed to the process of maximizing the objective function by adding a negative sign to the objective function.

The optimal solution satisfies $\lambda_i' = 0$ or $c_i'(y) - C_i' = 0$, $i = k' + 1, \ldots, k$ [29]. This has already been similarly applied to support vector machine(SVM) [30]. Thus, when variable $y$ satisfies strict inequality constraints, that is, when the constraint functions are strict inequality constraint, it is an inactive constraint, and only when the constraint function is an equality constraint can it be turned active. Therefore, the optimization problem with inequality constraints is transformed into the optimization problem with equality constraints and can be solved by Lagrange multiplier method [31]. As a result, the problem' solving process is simplified.

According to Lagrange multiplier method, the model is converted into the following form:

$$\max L(y, \lambda_i) = f(y) - \lambda_i(c_i(y) - C_i) - \lambda_i'(c_i'(y) - C_i') \quad (9)$$

where the current variable $y$ that satisfies $\nabla_y L(y, \boldsymbol{\lambda}_i) = 0$ and $\nabla_{\boldsymbol{\lambda}_i} L(y, \boldsymbol{\lambda}_i) = 0$ $\{i = 1, \ldots, k\}$ are local optimal solution which is the maximum point. We use gradient descent method to solve the local optimal solution. Solutions of $\lambda_i$ $\{i = 1, \ldots, k\}$ can be obtained by solving formulas $\nabla_{\boldsymbol{\lambda}_i} L(y, \boldsymbol{\lambda}_i) = 0$ $\{i = 1, \ldots, k\}$ and $\nabla_y L(y, \boldsymbol{\lambda}_i) = 0$ [32]. If the model is convex, the local optimal solution equals the global optimal solution. $\lambda_i \{i = 1, \ldots, k\}$ is the Lagrange multiplier, which implies that the objective function changes in accordance with constraint function. Because the optimal solution satisfies the constraint $c_i'(y) - C_i' = 0$, $\lambda_i'$ won't affect the solution of the optimization problem.

### D. SAFE REINFORCEMENT LEARNING

Reinforcement learning with safety issues is referred as safe reinforcement learning (SRL), which can be defined as maximizing the expected return value of related issues in the process of learning policy, and at the same time guarantees reasonable system performance and satisfies safety constraints in the whole process of the learning or exploring [33]. Existing safe reinforcement learning methods mainly include changing the objective function and improving the exploration process [34].

The safe reinforcement learning method based on changing the objective function is to obtain the optimal policy which can produce the maximum return in the worst case by maximizing the optimization criterion. This optimization criterion can mitigate the impact of variability caused by a given policy, which may lead to risk or adverse conditions. There are three main methods to ensure the safety of the agent in the system by changing the optimization criteria: Worst Case Criterion [35], Risk-Sensitive Criterion [36] and Constrained Criterion [37]. Worst Case Criterion maximizes the worst-case return to obtain the optimal policy and Risk-Sensitive Criterion method balances the reward and the risk to obtain the optimal policy. However, neither of the two approaches fundamentally solves the safety problem of the agent as the agent still has a certain probability of entering a dangerous state. Moreover, they expend more in exploration. Constrained Criterion method can better solve the safety problem of the agent, but the existing method is only experimental. Moldovan et al. proposed safe exploration in Markov decision processes [38] and Kadota et al. put forward discounted Markov decision processes with utility constraint functions [39]. Chow et al. introduced the classical Lyapunov Function in the control theory into reinforcement learning [40], but experiments only compared the two algorithms: Safe Policy Improvement (SDPI) and Safe Q-learning (SDQN).

Although all the above methods guarantee the safety of the agent by providing some constraints, they didn't consider the problem of global optimal solution and only ensured that the agent obtain a safe sub-optimal solution after exploration. The method of discounted Markov decision processes with utility constraints function only proposed the theoretical conclusion, without experimental verification.

There are two main ways to modify the exploration process to avoid that the agent enters into dangerous situations. The first method is to improve the algorithm by integrating external information; the second method is the exploration based on risk. Clouse et al. put forward the method of accepting external suggestions when the agent decides it needs guidance and the agent explores the optimal policy through external guidance [41]. Tang et al. regarded that the integrated external information was not used to guide the agent to explore, but to deduce a safety policy offline [42], [43]. Since the method requires extensive external information, it requires extra cost to integrate external information, in addition to exploration. However, the agent still has a certain probability to enter the dangerous state under this method.

## III. SAFE Q-LEARNING BASED ON CONSTRAINED MODEL

Methods based on constrained Markov decision processes have shown good results in improving safety of the agent, but the solution is not always effective and especially cannot be applied to the problem with large-scale space or continuous solution. Aiming at solving this problem, an algorithm named safe Q-learning based on constrained Markov decision processes is proposed and we adopt Lagrange multiplier method to solve this model. By linearizing the constraint function using Taylor expansion [44], the problem is transformed into a convex problem, so that the algorithm is able to find the global optimal solution rather than the sub-optimal solution.

For tasks with small discrete state space, the problem can be directly resolved through the constraint to determine whether a state is safe and which action the agent should currently take. If the state space and action space is very large and continuous, the problem can be solved by using the Lagrange multiplier method.

### A. CONSTRAINED SAFE Q-LEARNING MODEL

Reinforcement learning that is based on constrained Markov decision processes, referred to as constrained reinforcement learning (CRL), is described as a quintuple $(X, U, P, R, C)$ [12], where $X$ is the state space, $U$ is the action space, $P$ is the transition probability, $R$ is the immediate reward function, the set of constraint functions $C = \{c_i : X \times$

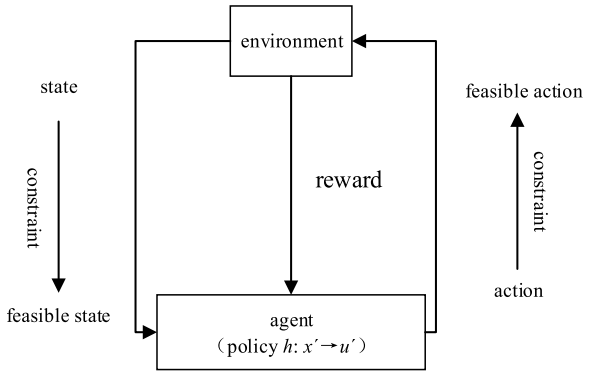


**FIGURE 2.** The illustration of constrained reinforcement learning.

$U \to \mathbb{R}|i = 1, \cdots, k$ represents that the action space is constrained, and the set of constraint functions whose state space is constrained is represented as $\overline{C} = \{\overline{c}_i : X \to \mathbb{R}|i = 1, \cdots, k$. The purpose of constrained reinforcement learning is to maximize the reward function. The constrained reinforcement learning model is shown in Figure 2.

The safe Q-learning model that adds constraints to the action space on the basis of Q-learning is as follows:

$$\begin{aligned} &Q_{t+1}(x_t, u_t) \\ &= Q_t(x_t, u_t) + \alpha_t \left( r_{t+1} + \gamma \max_{u'} Q_t\left(x_{t+1}, u'\right) - Q_t(x_t, u_t) \right) \\ &\quad s.t.\ c_i(x_t, u_t) = C_i, \quad i = 1, 2, \cdots k', \\ &\quad\ \ c_i'(x_t, u_t) \le C_i', \quad i = k'+1, \cdots, k \end{aligned} \tag{10}$$

where formulas $c_i(x_t, u_t) = C_i$ and $c_i'(x_t, u_t) \le C_i'$ are the abstract representation of constraint functions, indicator sets {1,2, …, k'} and {k'+1, k'+2, …, k} represent the equality constraint and the inequality constraint respectively, $x_t$ represents the state of the agent at time $t$, and $u_t$ represents the action selected by the agent under state $x_t$ and time $t$.

The state space can be divided into a feasible state space set and an infeasible one by adding constraints. The action space is also divided into a feasible action space set and an infeasible one, which ensures safety from the beginning of scheduling by determining whether the state is safe through the constraint condition, and only in the safe state is agent allowed to proceed. Therefore, the feasible region of the safe Q-learning model is represented by Eqs.11 and 12.

$$\begin{aligned} \overline{X} = \Big\{ & x | \overline{c_i}(x_t) = C_i, i = 1, \cdots, k'; \overline{c_i'}(x_t) \le C_i', \\ & i = k'+1, \cdots, k \Big\} \end{aligned} \tag{11}$$

$$\begin{aligned} \overline{U} = \big\{ & u | c_i(x_t, u_t) = C_i, i = 1, \cdots, k'; c_i'(x_t, u_t) \le C_i', \\ & i = k'+1, \cdots, k \big\} \end{aligned} \tag{12}$$

where $\overline{U}$ is a feasible action space and $\overline{X}$ is a feasible state space, $x$ represents any safety state and $u$ represents any safety action. The inequality constraint $c_i(x_t, u_t) \le C_i$ on the action and state space is a general form. If the inequality constraint form $c_i(x_t, u_t) \ge C_i$ is satisfied for some states, it is only necessary to multiply both sides of the inequality by -1 to convert the inequality constraint into a general form. Indicator sets {1,2, …, k'} and {k'+1, k'+2, …, k} are represented by the expression 13.

$$\begin{aligned} \xi &= \{1, 2, \cdots, k'\}, \\ I &= \{k'+1, \cdots, k\}, \\ I' &= \{i | c_i(x, u) = C_i, i \in I\} \end{aligned} \tag{13}$$

where $\xi$ represents the indicator set of equality constraint functions, $I$ represents the indicator set of non-strict inequality constraint functions, and $I'$ represents the indicator set of non-strict inequality constraint functions ($c(x) \ge C$ or $c(x) \le C$) that satisfy the equality constraint by taking an action $u$ under a certain state $x$. For non-strict inequality constraint, if indicator $i_0 \in I'$ is present, the $i_0$th constraint is an inactive constraint at the state $x$, and the non-strict inequality constraint function can be converted into equality constraint function.

The set of effective constraint indicators at state $x$ is $\xi \cup I'$, and above-mentioned inequality constraints can be transformed into equality constraints, so the safe Q-learning model is modified to Eq.14:

$$\begin{aligned} &Q_{t+1}(x_t, u_t) \\ &= Q_t(x_t, u_t) + \alpha_t \left( r_{t+1} + \gamma \max_{u'} Q_t\left(x_{t+1}, u'\right) - Q_t(x_t, u_t) \right) \\ &\quad s.t.\ c_i(x_t, u_t) = C_i, \quad i \in \xi \cup I' \end{aligned} \tag{14}$$

The above-mentioned safe Q-learning model only contains equality constraints, because only the equality constraint is an active constraint, and the strict inequality constraint removed is an inactive constraint, which reduces the complexity and the difficulty of solving the model.

### B. SAFE MODEL WITH ACTION CONSTRAINTS

The safe Q-learning model adds multi-dimensional constraints to the Q-learning model, so the traditional solution method is no longer able to deal with the problem. In order to solve safe Q-learning efficiently and accurately, we propose a method to solve the optimal action that can be performed in the current state by Lagrange multiplier method on the basis of Q-learning, which requires that the objective function and the constraint function are first-order continuous and differentiable. The first-order continuous differentiability of the objective function can be satisfied in the case of continuous time $t$, but the constraint function does not always guarantee the first-order continuous differentiability during the process of construction. The differentiability of the constraint function can be realized by linearizing the constraint function. Since the next state of the agent is determined by the current state and the current action, the following formulas can be obtained:

$$x_{t+1} \sim f(x_t, u_t) \tag{15}$$

$$\overline{c_i}(x_{t+1}) \doteq c_i(x_t, u_t) \tag{16}$$

where Eq.16 indicates that the state of the agent at next time is determined by the current state of the agent and the action currently executed, and it represents a mapping relationship from $x_t$ and $u_t$ to $x_{t+1}$, and formula 16 indicates that the safety of the current state and the current action is determined by the safety of the state of the agent at next time.

In the process of solving the model, in order to obtain a global optimal solution, the objective function and the constraint function are required to be convex functions. According to the safe Q-learning model, the objective function is a convex function, but the constraint function may be not. In this paper, for the constraint function we use a linear approximation. Since the linear function must be a convex function, the constrained function obtained is a convex function. At this point, the solution of the constrained reinforcement learning model must be able to obtain the global optimal solution. A linear approximation of the constraint function is:

$$c_i(x_t, u_t) \approx \overline{c_i}(x_t) + d\left(x_t; \omega_i\right)^T u_t \tag{17}$$

where $x_t$ is used as input, the output that has the same dimension with $u_t$ is a vector, and $d(x_t; \omega_i)$ can be obtained by solving the following function:

$$\arg\min_{\omega_i} \sum_{(x_t, u_t, x_{t+1}) \in D} \left(\overline{c_i}(x_{t+1}) - \left(\overline{c_i}(x_t) + d\left(x_t; \omega_i\right)^T u_t\right)\right)^2, \quad D = \{(x_t, u_t, x_{t+1})\} \tag{18}$$

where $D$ is a set whose elements are a triad $(x_t, u_t, x_t')$ meaning that agent takes action $u_t$ and transfer from state $x_t$ to state $x_t'$ and the optimal solution of the objective function is found in set $D$. Therefore, the safe Q-learning model obtained by linear approximation of the constraint function is:

$$\begin{aligned} &Q_{t+1}(x_t, u_t) \\ &= Q_t(x_t, u_t) + \alpha_t\left(r_{t+1} + \gamma \max_{u'} Q_t\left(x_{t+1}, u'\right) - Q_t(x_t, u_t)\right) \\ &\quad s.t.\ c_i(x_t, u_t) \approx \overline{c_i}(x_t) + d\left(x_t; \omega_i\right)^T u_t = C_i, \quad i \in \xi \cup I' \end{aligned} \tag{19}$$

Solving the above model according to the Lagrange multiplier method is to solve the following formula:

$$u^* = \arg\max_{u_t} \left\{ Q_{t+1}(x_t, u_t) - \sum_{i \in \xi \cup I'} \lambda_i \left(\overline{c_i}(x_t) + d\left(x_t; \omega_i\right)^T u_t - C_i\right) \right\} \tag{20}$$

where $u*$ is the optimal action that the agent can take at the current state, $\lambda_i\{i \in \xi \cup I\}$ is lagrangian multiplier.

In order to avoid the maximum long-term cumulative return value of the model falling into the local optimal solution, we use linearization to convert Q-learning model's constraint function into convex function. Therefore, the optimal solution obtained by the Lagrange multiplier method that is the global optimal solution. The following is a proof that the global optimal solution can be obtained by using the Lagrange multiplier method to solve the safe Q-learning model Eq.19.

*Proposition 1*: It is assumed that safe Q-learning model Eq.19 has a feasible solution $\{u*, \{\lambda_i^*\}_{i=1}^*\}$ at the state $x*$, where $\lambda_i^*$ is the best Lagrange multiplier associated with the *ith* constraint.

*Proof*: Let $u*$ be the local optimal solution of problem Eq.18, and $Q_{t+1}(x_t, u_t)$ and $\overline{c}_i(x_t) + d(x_t; \omega_i)^T u_t \{i \in \xi \cup I\}$ are the first order continuous differentiable in the neighborhood of $u*$. If the constraint specification condition is established as

$$SFD\left(u^*, U\right) = LFD\left(u^*, U\right) \tag{21}$$

where SFD [45] is the abbreviation of sequence feasible direction and LFD [45] is the abbreviation of linear feasible direction, the existence of $\lambda_i^*\{i \in \xi \cup I\}$ makes that formulas (22)-(25) are established:

$$\nabla Q_{t+1}\left(x^*, u^*\right) = \sum_{i \in \xi \cup I'} \lambda_i^* \nabla\left(\overline{c_i}(x^*) + d\left(x^*; \omega_i\right)^T u^* - C_i\right) \tag{22}$$

$$\overline{c_i}(x^*) + d\left(x^*; \omega_i\right)^T u^* - C_i = 0, \quad i \in \xi \tag{23}$$

$$\lambda_i^* \geq 0, \quad i \in I' \tag{24}$$

$$\lambda_i^*\left(\overline{c_i}(x^*) + d\left(x^*; \omega_i\right)^T u^* - C_i\right) = 0, \quad i \in I' \tag{25}$$

As $u*$ is a local optimal and feasible solution, Eq.23 is active. Let $d \in SFD(u^*, U)$, since $u*$ is a local optimal solution, $d^T \nabla Q_{t+1}(x^*, u^*) \leq 0, \forall d \in SFD(u^*, U)$ can be obtained from the geometric optimality condition.

By the constraint specification condition (21), the equations

$$\begin{aligned} d^T \nabla\left(\overline{c_i}(x^*) + d\left(x^*; \omega_i\right)^T u^* - C_i\right) &= 0, \quad i \in \xi \\ d^T \nabla\left(\overline{c_i}(x^*) + d\left(x^*; \omega_i\right)^T u^* - C_i\right) &\leq 0, \quad i \in I' \\ d^T \nabla Q_{t+1}\left(x^*, u^*\right) &> 0 \end{aligned} \tag{26}$$

have no solution.

Using the Farkas Lemma [46]:

$$\begin{aligned} \nabla Q_{t+1}\left(x^*, u^*\right) = &\sum_{i \in \xi} \lambda_i^*\left(\overline{c_i}(x^*) + d\left(x^*; \omega_i\right)^T u^* - C_i\right) \\ &+ \sum_{i \in I'} \lambda_i^*\left(\overline{c_i}(x^*) + d\left(x^*; \omega_i\right)^T u^* - C_i\right) \end{aligned}$$

Finally, obviously there is $\lambda_i^* \geq 0$, and formula (27).

$$\lambda_i^*\left(\overline{c_i}(x^*) + d\left(x^*; \omega_i\right)^T u^* - C_i\right) = 0, \quad \forall i \in I' \tag{27}$$

Because safe Q-learning model is a convex model, the local optimal solution is the global optimal solution. □

Proposition 1 is the first-order optimality condition theorem, which theoretically guarantees that Lagrange multiplier method can solve the safe Q-learning model's optimal action at each step and obtain the global optimal solution. The algorithm of the safe Q-learning is convergent with the above proposition and the convergence of Q-learning.

### C. ALGORITHM DESCRIPTION

We propose a safe Q-learning algorithm based on the constrained Markov decision processes, which limits the action executed at each step to the set of safe action by adding multi-dimensional constraints. By limiting the feasible state of the agent to the set of safe state, the safety of the agent can be guaranteed at the early stage of exploration. Safe Q-learning algorithm is shown in Algorithm 1.

**Algorithm 1** Safe Q-Learning (SQL)

**Input**: state set $X$, action set $U$, and reward function
**Output**: safe state-action pair sequence
1: initialize: state-action value function $Q(x,u), \forall x \in X, u \in U$, Lagrange multiplier $\lambda_i, i = 1, \cdots, k$, $k \in N^+$, parameter $\omega_i$, step size $\alpha \in (0,1]$ and $D = \{(x,u,x')\}, \forall x, x' \in X, u \in U$
2: **Repeat**
3: initialize initial state
4: **Repeat**
5: get $d(x;\omega_i)$ by solving the formula $\omega_i^* \leftarrow \arg\min_{\omega_i} \sum_{(x,u,x')\in D} \left(\overline{c_i}(x') - \left(\overline{c_i}(x) + d(x;\omega_i)^T u\right)\right)^2$
6: the constraint function is approximated linearly: $c_i \leftarrow \overline{c_i}(x) + d(x,\omega_i^*)^T u$
7: get the action by The Lagrange multiplier method $u^* \leftarrow \arg\max_u \{Q(x,u) - \sum_{i\in\xi\cup I'} \lambda_i (c_i - C_i)\}$
8: take action $u^*$, observe $r, x'$
9: $Q(x,u^*) \leftarrow Q(x,u^*) + \alpha\left[R + \gamma \max_u Q(x',u) - Q(x,u^*)\right]$
10: **Until** termination
11: performs the action $u^*$, then moves to the next state: $x \leftarrow x'$
12: **Until** $x$ is terminal

For steps from 5 to 9 in algorithm1, the Lagrange multiplier method is used to solve the constraint problem to obtain the safety action under the condition that the long-term cumulative return value is the largest, and the safety of the agent is ensured while satisfying the global optimality. In step 11, the agent performs the optimal safety action and moves to next state.

## IV. EXPERIMENT AND ANALYSIS

Safe Q-learning method based on the constrained Markov decision processes can be used for the exploration of the agent in a limited situation to obtain the maximum long-term cumulative return value so as to improve the safety of the agent. Comparison algorithms in this paper are classical reinforcement learning algorithms Sarsa [2], Sarsa(λ) [2] and Q-learning [18]. All comparison algorithms and the safe Q-learning algorithms are used to solve model-free reinforcement learning problems.

In order to ensure the safety of the agent during the process of exploration, we add constraint to the objective function represented by Eq.28.

$$d(x_{t+1}, \chi) > 0 \tag{28}$$

where, $\chi$ is a set consisting of all the dangerous states, and $x_{t+1}$ is the next state. Eq.28 shows the distance between $x_{t+1}$ and $\chi$ in the form of Manhattan distance [47] between the current state and dangerous states which is positive.

In reinforcement learning, the agent isn't aware of the experimental environment, and when the agent performs an action, it may move in different directions, reach different states and get different rewards. The agent maximizes the long-term cumulative return value to obtain the optimal action and the state reached at next moment, and finally gets the optimal path.

Reinforcement learning algorithm uses long-term cumulative reward as an evaluation metric because reinforcement learning algorithms seek for an optimal solution by maximizing long-term reward. Therefore, larger long-term reward value denotes better performance. In the experiment we utilized long-term reward as evaluation. Moreover, we used the standard deviation of reward to exam the stability of the algorithms.

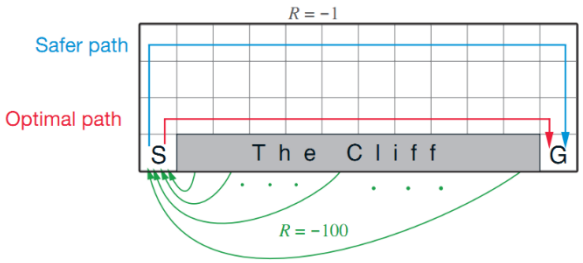


**FIGURE 3.** The schematic diagram of Cliff walking.

### A. CLIFF WALKING EXPERIMENT

In the cliff walking experiment, the agent needs to find a shortest path from start point S to terminal point G without falling into the cliff, as shown in Figure 3.

The purpose is to learn a safe shortest path. The gray part is dangerous state. If the agent reaches cliff, the agent gets a reward of -100 and ends the exploration, then the agent needs to return to the starting position to explore again. In addition to the dangerous state and the terminal state, in order to prevent the agent from strolling freely in the grid, the agent will get a reward of -1 every time it enters a new state, so as to prevent the agent from walking freely in the grid, ensuring that the agent can find the shortest path into the termination state in the shortest period of time. In Figure 3, the path indicated by the red line is the optimal path. The agent can go from the starting point to the end point in the shortest time period under the premise of ensuring safety. The path indicated by the blue line is only the safety path, that is, the further from the cliff the agent experiences in the process from start to finish, the safer the agent is, but this is often not the optimal path.

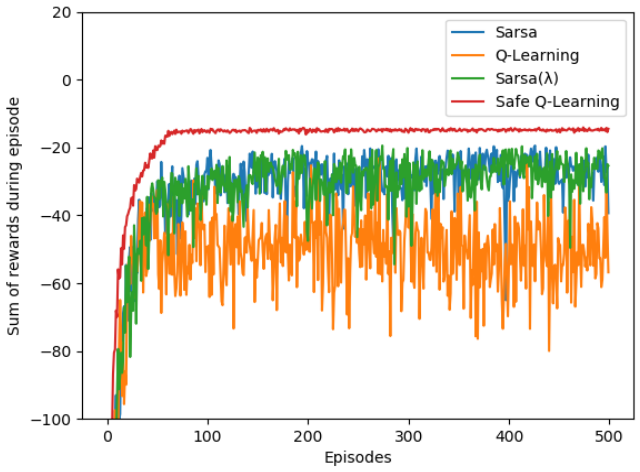


**FIGURE 4.** The long-term cumulative return value of each episodes of four algorithms in the cliff walking .

The baselines in the experiment setting is given, the step size $\alpha$ is uniformly set to 0.5, and the discount factor $\gamma$ is set to 1. The $\varepsilon$-greedy method is used for the agent to explore the random action in the training phase, in order to avoid the agent falling into the local optimum in the learning process. The policy parameter $\varepsilon$ set to 0.1. The experiment runs 50 times independently, and the number of episodes for each independent operation was 500.

We compared the learning performance of the Sarsa, Sarsa(λ), Q-learning and safe Q-learning in the Cliff Walking. Figure 4 shows the long-term cumulative return value of each episodes of four algorithms.

As shown in Figure 4, the long-term cumulative return value obtained using the safe Q-learning algorithm is significantly higher than those of Sarsa, Sarsa(λ), and Q-learning. As Q-learning is a typical off-policy algorithm, the behavior policy that is used for generating behavior is different from the evaluation policy that is used for updating value function. While as Sarsa and Sarsa(λ) are on-policy algorithms, their behavior policy and evaluation policy are the same. As a result, the agent of Sarsa and Sarsa(λ) can't avoid entering the dangerous state during the exploration process. Every time the agent enters a dangerous state, the agent gets a reward of -100 and returns to the initial state, so the long-term cumulative return values of each episode calculated by Q-learning, Sarsa and Sarsa (λ) algorithm are not stable. The agent is easy to enter dangerous state and has great volatility. The long-term accumulation of each episode calculated by Q-learning algorithm is obtained, which is the most unstable and the learning effect is the worst. On the contrary, the safe Q-learning algorithm can get a better long-term cumulative return value per episode, because the safety of the agent is ensured by adding constraints, and the agent does not enter the dangerous state and causes unnecessary loss. Therefore, the algorithm converges faster and the learning effect is better.

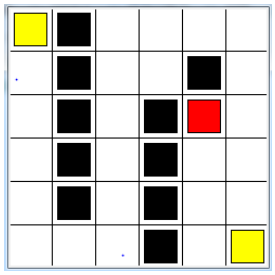

**FIGURE 5.** Schematic diagram of 6 × 6 maze with traps.

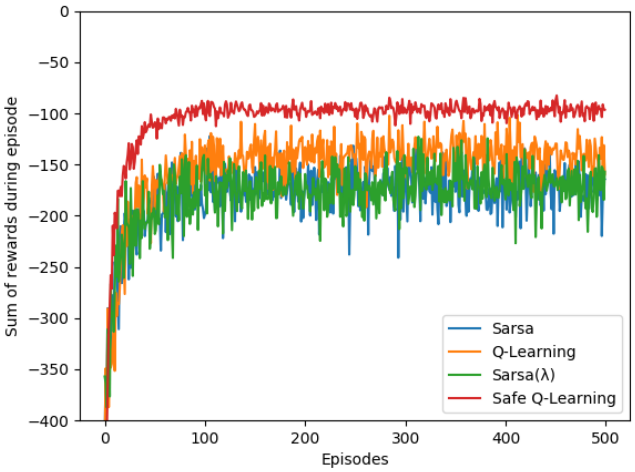


**FIGURE 6.** The long-term cumulative return value of each episodes of four algorithms in the 6 × 6 labyrinth experiment with traps.

### B. MAZE EXPERIMENTS WITH TRAPS

The second comparative experiment is a maze world with traps. The agent is to find out a safe shortest path from the start point to the terminal point without entering the trap. In the maze experiment, there is a $6 \times 6$ grid, in which the black grid represents the trap, the yellow grid in the upper left corner is the starting point, the yellow grid in the lower right corner is the terminal point, and the red grid refers to the grid position where the agent currently is. When the agent enters the black grid, a dangerous state, the agent will get a reward of -10. When the agent reaches the lower right corner, the destination block, the agent will get a reward of 10. In order to urge the agent to find a shortest safe path from the starting point to get to the destination as quickly as possible, the agent will get a reward of -1 every time it reaches a new state, as shown in Figure 5.

The baselines in the experiment setting is given, the step size $\alpha$ is set to 0.1, the discount factor $\gamma$ is uniformly set to 0.9, the parameter $\varepsilon$ of the $\varepsilon$-greedy policy is set to 0.8, and the number of the episode is set to 500.

Figure 6 shows the long-term cumulative return value of each episode of the four reinforcement learning algorithms of Sarsa, Sarsa(λ), Q-learning, and safe Q-learning in the maze experiment.

According to Figure 6, the long-term cumulative return value of per episode calculated by the safe Q-learning

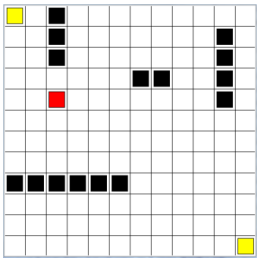

**FIGURE 7.** Schematic diagram of a 12 × 12 maze with traps.

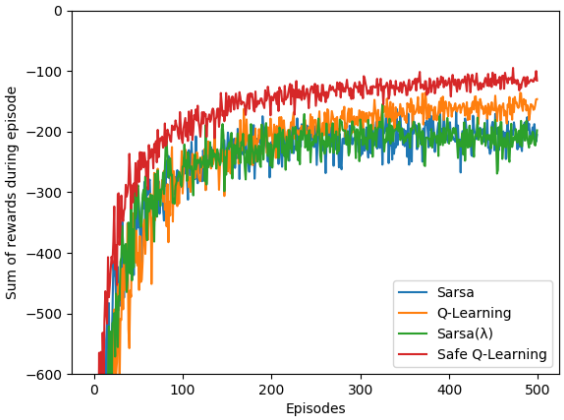


**FIGURE 8.** The long-term cumulative return value of each episodes of four algorithms in 12 × 12 maze experiment with traps.

algorithm is significantly higher than those calculated by the Sarsa, Sarsa(λ), and Q-learning algorithms, because under the safe Q-learning algorithm, the agent does not enter the trap during the learning process, while the Sarsa, Sarsa(λ) and Q-learning algorithms continue to learn to avoid dangerous situation by entering the trap, so the calculated long-term cumulative return value of per episode is low and unstable. The safe Q-learning algorithm ensures the safety of the agent during the exploration process and avoids the loss of the agent as a result of entering a dangerous state.

### C. LARGE-SCALE MAZE EXPERIMENTS WITH TRAPS

In order to verify that the safe Q-learning algorithm still has better performance when the state space and action space have a larger scale, we expand the 6 × 6 maze world with traps to a 12 × 12 maze world with traps, increasing the number of traps and the random distribution of traps, as shown in Figure 7.

As shown in Figure 8, the safe Q-learning algorithm still has the best performance, and the long-term cumulative return value converges faster and more stable. In the reinforcement learning algorithm, the choice of the action of the agent is random, and the agent cannot be completely prevented from entering the dangerous state. The safe Q-learning algorithm can ensure that the agent does not enter the dangerous state during the process of exploration, thus ensuring the safety of the agent and avoid damage to the agent and unnecessary losses.

**TABLE 1.** The standard deviation of different methods in different games.

| game | Sarsa | Sarsa(λ) | Q-learning | safe Q-learning |
|---|---|---|---|---|
| cliff walking | 4.9 | 4.4 | 10.4 | **0.31** |
| 6×6 maze | 18.1 | 17.8 | 15.5 | **4.8** |
| 12×12 maze | 25.4 | 25.8 | 35.0 | **23.1** |

### D. ANALYSIS OF ALGORITHM STABILITY

To evaluate stableness of safe Q-learning, we calculate the standard deviation of safe Q-learning, Q-learning, Sarsa (λ) and Sarsa using formula (29). According to the table 1, safe Q-learning has the lowest standard deviation and the best stability.

$$std = \sqrt{\left((r_1 - \bar{r})^2 + (r_2 - \bar{r})^2 + \cdots + (r_n - \bar{r})^2\right)/n} \qquad (29)$$

where $\bar{r}$ is the average of the sum of rewards $r_i \{i = 1 \cdots n\}$.

The experiment calculated the standard deviation for each method when models converge and the formula of standard deviation is as illustrated in the table below.

## V. CONCLUSION

The application of reinforcement learning in the industrial field makes the safety of the agent increasingly important. For the safety of the agent, we propose a safe Q-learning algorithm based on the constrained Markov decision processes. The feasible action space and state space of the agent are limited to the safe action space and the safe state space, thereby ensuring the safety of the agent.

The safe Q-learning algorithm proposed in this paper can guarantee the safety of the agent in the exploration process, which can be possibly applied not only in video game and toy to improve players' performance, but also in industry area such as automatic drive, where cars are prevented from moving to dangerous areas and hitting moving objects on the path; the method can also be applied in robot platform, enabling the robot to avoid obstacles on the road to destination point and program an energy-saving path. In our subsequent experiments, we find that the Lagrange multiplier method also can be applied to other reinforcement learning algorithms to solve the safety problems of the agent. The safe Q-learning algorithm is based on a constrained Markov decision processes, so the algorithm can also be applied to constrained models, such as models with limited resources or minimum cost, multi-objective models, and limited speed for agents.

The safe Q-learning algorithm can be further extended to the modeling and solving of constrained problems. In the future, we are looking forward to solving the constrained problem of the agent by applying the safe Q-learning algorithm.

## REFERENCES

[1] B. Luo, D. Liu, T. Huang, and D. Wang, "Model-free optimal tracking control via critic-only Q-learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 10, pp. 2134–2144, Oct. 2016.

[2] R. S. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[3] V. Mnih, 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, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.
[4] A. E. L. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electron. Imag.*, vol. 19, pp. 70–76, Jan. 2017.
[5] M. C. Machado, M. G. Bellemare, E. Talvitie, J. Veness, M. Hausknecht, and M. Bowling, "Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents," *J. Artif. Intell. Res.*, vol. 61, pp. 5573–5577, Mar. 2017.
[6] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 1437–1480, 2015.
[7] G. D. Luenberger, *Investment Science*. New York, NY, USA: Oxford Univ. Press, 2013.
[8] C. Gehring and D. Precup, "Smart exploration in reinforcement learning using absolute temporal difference errors," in *Proc. Int. Conf. Auton. Agents Multi-Agent Syst.*, Saint Paul, MN, USA, 2013, pp. 1037–1044.
[9] K. Driessens and S. Džeroski, "Integrating guidance into relational reinforcement learning," *Mach. Learn.*, vol. 57, no. 3, pp. 271–304, 2004.
[10] D. D. Castro, A. Tamar, and S. Mannor, "Policy gradients with variance related risk criteria," in *Proc. 29th Int. Conf. Mach. Learn.*, Edinburgh, Scotland, vol. 1, 2012, pp. 1–8.
[11] J. Mahmoudimehr and P. Sebghati, "A novel multi-objective dynamic programming optimization method: Performance management of a solar thermal power plant as a case study," *Energy*, vol. 168, pp. 796–814, Feb. 2018.
[12] E. Altman, *Constrained Markov Decision Processes*. Boca Raton, FL, USA: CRC Press, 1999.
[13] A. Shenhav, S. Musslick, F. Lieder, W. Kool, T. L. Griffiths, J. D. Cohen, and M. M. Botvinick, "Toward a rational and mechanistic account of mental effort," *Annu. Rev. Neurosci.*, vol. 40, no. 1, pp. 99–124, 2017.
[14] Z. Wei, J. Xu, Y. Lan, J. Guo, and X. Cheng, "Reinforcement learning to rank with Markov decision process," in *Proc. 40th Int. SIGIR Conf. Res. Develop. Inf. Retr.*, 2017, pp. 945–948.
[15] M. El Chamie, Y. Yu, B. Açikmese, and M. Ono, "Controlled Markov processes with safety state constraints," *IEEE Trans. Autom. Control*, vol. 64, no. 3, pp. 1003–1018, Mar. 2019.
[16] Y. He, Z. Zhang, F. R. Yu, N. Zhao, H. Yin, V. C. M. Leung, and Y. Zhang, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10433–10445, Nov. 2017.
[17] D. Abel, D. E. Hershkowitz, and M. L. Littman, "Near optimal behavior via approximate state abstraction," in *Proc. 33rd Int. Conf. Mach. Learn.*, vol. 48, Jun. 2016, pp. 2915–2923.
[18] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.
[19] T. Schaul, J. Quan, and I. Antonoglou, "Prioritized experience replay," *Comput. Sci.*, to be published.
[20] Y. Li, Z. Hou, Y. Feng, and R. Chi, "Data-driven approximate value iteration with optimality error bound analysis," *Automatica*, vol. 78, pp. 79–87, Apr. 2017.
[21] R. M. Golden, "Adaptive learning algorithm convergence in passive and reactive environments," *Neural Comput.*, vol. 30, no. 10, pp. 2805–2832, 2018.
[22] A. Gattami, "Reinforcement learning for multi-objective and constrained Markov decision processes," 2019, *arXiv:1901.08978*. [Online]. Available: https://arxiv.org/abs/1901.08978
[23] A. Bušić and S. Meyn, "Action-constrained Markov decision processes with Kullback-Leibler cost," in *Proc. 31st Conf. Learn. Theory*, vol. 75, 2018, pp. 1431–1444.
[24] V. S. Borkar, "An actor-critic algorithm for constrained Markov decision processes," *Syst. Control Lett.*, vol. 54, no. 3, pp. 207–213, Mar. 2005.
[25] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 22–31.
[26] W. Min, and U. Topcu, "Constrained cross-entropy method for safe reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–11.
[27] J. Martyna, "Power allocation in cognitive radio with distributed antenna system," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems* (Lecture Notes in Computer Science), vol. 10531. Cham, Switzerland: Springer, 2017, pp. 745–754.
[28] R. Andreani, L. D. Secchin, and P. J. S. Silva, "Convergence properties of a second order augmented Lagrangian method for mathematical programs with complementarity constraints," *SIAM J. Optim.*, vol. 28, no. 3, pp. 2574–2600, 2018.
[29] M. Li, "Generalized Lagrange multiplier method and KKT conditions with an application to distributed optimization," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 2, pp. 252–256, Feb. 2019.
[30] T.-W. Kuan, J.-F. Wang, J.-C. Wang, P.-C. Lin, and G.-H. Gu, "VLSI design of an SVM learning core on sequential minimal optimization algorithm," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 4, pp. 673–683, Apr. 2012.
[31] L. Qi and Z. Wei, "On the constant positive linear dependence condition and its application to SQP methods," *SIAM J. Optim.*, vol. 10, no. 4, pp. 963–981, 1999.
[32] F. Farina, A. Garulli, A. Giannitrapani, and G. Notarstefano, "Asynchronous distributed method of multipliers for constrained Nonconvex optimization," in *Proc. Eur. Control Conf. (ECC)*, vol. 103, Jun. 2018, pp. 243–253.
[33] M. Turchetta, F. Berkenkamp, and A. Krause, "Safe exploration in finite Markov decision processes with Gaussian processes," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 1–15.
[34] M. Pecka and T. Svoboda, "Safe exploration techniques for reinforcement learning—An overview," in *Modelling and Simulation for Autonomous Systems* (Lecture Notes in Computer Science), vol. 8906. Cham, Switzerland: Springer, pp. 357–375, 2014.
[35] A. Tamar, H. Xu, and S. Mannor, "Scaling up robust MDPs by reinforcement learning," *Comput. Sci.*, to be published.
[36] A. Basu, T. Bhattacharyya, and V. S. Borkar, "A learning algorithm for risk-sensitive cost," *Math. Oper. Res.*, vol. 33, no. 4, pp. 880–898, 2008.
[37] L. Xia, "Optimization of Markov decision processes under the variance criterion," *Automatica*, vol. 73, pp. 269–278, Nov. 2016.
[38] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with Gaussian processes," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 493–496.
[39] D. A. D. M. Moreira, K. V. Delgado, and L. N. D. Barros, "Risk-sensitive Markov decision process with limited budget," in *Proc. Brazilian Conf. Intell. Syst. (BRACIS)*, vol. 1, Oct. 2018, pp. 109–114.
[40] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A Lyapunov-based approach to safe reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–10.
[41] A. Fachantidis, M. E. Taylor, and I. Vlahavas, "Learning to teach reinforcement learning agents," *Mach. Learn. Knowl. Extraction*, vol. 1, no. 1, pp. 21–42, 2019.
[42] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1608–1639, 2010.
[43] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intell. Service Robot.*, vol. 9, no. 1, pp. 1–29, 2016.
[44] K. Komatsu and H. Takata, "Nonlinear feedback control of stabilization problem via formal linearization using Taylor expansion," in *Proc. Int. Symp. Inf. Theory Its Appl.*, Dec. 2008, pp. 1–5.
[45] W. Y. Sun, C. X. Xu, and D. T. Zhu, *Optimization Method*. Beijing, China: Higher Education Press, 2010.
[46] M. J. Cánovas, N. Dinh, D. H. Long, and J. Parra, "An approach to calmness of linear inequality systems from Farkas lemma," *Optim. Lett.*, vol. 13, no. 2, pp. 295–307, 2019.
[47] S. R. Blackburn, C. Homberger, and P. Winkler, "The minimum Manhattan distance and minimum jump of permutations," *J. Combinat. Theory Ser. A*, vol. 161, pp. 364–386, Jan. 2019.

**YANGYANG GE** is currently pursuing the master's degree with the School of Computer Science and Technology, Soochow University. Her main research interest includes safe reinforcement learning.

**FEI ZHU** received the Ph.D. degree. He is currently an Associate Professor with the School of Computer Science and Technology, Soochow University. His main research interests include deep learning, reinforcement learning, text mining, and pattern recognition. He studies to design and applies machine learning algorithms to solve health data science, health informatics, predictive analytics, and personalized data-driven decision support problems. He is a member of China Computer Federation.

**XINGHONG LING** received the Ph.D. degree. He is currently an Associate Professor with the School of Computer Science and Technology, Soochow University. His main research interests include artificial intelligence, reinforcement learning, and deep reinforcement learning. He is a member of China Computer Federation.

**QUAN LIU** received the Ph.D. degree. He is currently a Professor with the School of Computer Science and Technology, Soochow University. His main research interests include intelligence information processing, automated reasoning, and machine learning.