Article

Optimal Security Protection Strategy Selection Model Based on Q-Learning Particle Swarm Optimization

Xin Gao, Yang Zhou *, Lijuan Xu and Dawei Zhao *

Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China

* Correspondence: zhouy@sdas.org (Y.Z.); zhaodw@sdas.org (D.Z.)

Abstract: With the rapid development of Industrial Internet of Things technology, the industrial control system (ICS) faces more and more security threats, which may lead to serious risks and extensive damage. Naturally, it is particularly important to construct efficient, robust, and low-cost protection strategies for ICS. However, how to construct an objective function of optimal security protection strategy considering both the security risk and protection cost, and to find the optimal solution, are all significant challenges. In this paper, we propose an optimal security protection strategy selection model and develop an optimization framework based on Q-Learning particle swarm optimization (QLPSO). The model performs security risk assessment of ICS by introducing the protection strategy into the Bayesian attack graph. The QLPSO adopts the Q-Learning to improve the local optimum, insufficient diversity, and low precision of the PSO algorithm. Simulations are performed on a water distribution ICS, and the results verify the validity and feasibility of our proposed model and the QLPSO algorithm.

Keywords: Bayesian attack graph; optimal protection strategy; Q-Learning; particle swarm optimization

1. Introduction

An industrial control system refers to the equipment, system, network, and controller applied in industrial production, whose main functions are to operate, control, and assist industrial automation production [1–3]. ICS is mainly divided into four parts: Supervisor Control And Data Acquisition (SCADA) [4], Distributed Control System (DCS) [5], Programmable Logic Controller (PLC) [6], and Process Control System (PCS) [7]. ICS is widely used in electric power, nuclear power plants, petroleum, and other industries [8,9]. Due to the increasingly close connection between ICS and the Internet, ICS often faces complex network attacks, which pose a huge threat to the social economy and people’s security. In recent years, cyber-attack incidents against ICS have occurred frequently, such as the Iranian “Stuxnet” incident in 2010 [10] and the Ukraine power grid cyber-attack in 2015 [11], causing serious property damage. However, there are essential differences between ICS and IT, which results in the unavailability of the traditional IT security protection strategies, such as access control, firewall, and so on, for the protection of ICS [12–15].

Recently, a hot research topic in the protection strategies for ICS has been to focus on the active protection, which implements active protection strategies based on the risk assessment on the current security risk of the ICS. Shameli-Sendi et al. proposed a retrospective burst response method based on an adaptive and cost-sensitive model [16]. This method takes into account the effectiveness of the application response. However, this approach does not meet the security requirements of ICS well. K. Deb et al. proposed the non-dominated sorting GA algorithm, which can keep the diversity of solutions at a high level [17]. Genetic algorithms generate defense and attack strategies, while fitness functions are used to infer dominant strategies. This method can maintain a high accuracy, but it will
fall into the local optimal and cannot solve the optimal solution. Granadillo et al. proposed a geometric model to select a best-response combination based on the response Return on Investment (RORI) index [18]. However, this approach ignores attack modeling. Attack modeling is critical because cyber attacks are becoming more destructive these days, and the response time is the determining factor. Miehling et al. used the concept of POMDP to model defense issues [19]. Their goal is to provide the best dynamic defense strategy against ongoing cyber attacks against protected systems. However, this approach uses only a single metric: the cost of deployment costs of attack or defense operations, and cannot quantify the problem well.

In the stage of implementing protection strategies, each protection strategy is accompanied with costs. How to balance the benefits and costs of protection under the constraint of limited resources is a typical optimization problem. Constructing such a reasonable optimization problem and finding its optimal solution are the main challenges.

In this paper, we propose an optimal security protection strategy selection model based on QLPSO. In order to evaluate the security situation of ICS, we introduce the protection strategy into the Bayesian attack graph, and calculate the probability of each attribute node being attacked to obtain the risk value. When choosing the optimal protection strategy, the general PSO algorithm often falls into a local optimum. To solve this problem, QLPSO is proposed, which updates the parameters of PSO algorithm through Q-Learning. Finally, we verify the validity and feasibility of our model and the QLPSO algorithm for a water distribution ICS.

2. Related Work

A lot of research has been done on constructing the optimal security protection strategy for various complex systems. Jaquith proposed security metrics, such as attack cost, defense implementation cost, attack impact, operation cost, and other indicators to define the factors of optimal solution [20]. However, this approach lacks specific and commonly used measurement systems to reliably evaluate countermeasures. S. Bandyopadhyay et al. proposed the single-objective optimization problem and multi-objective optimization problem to determine the optimal strategy [21]. Nevertheless, they did not discuss how to find the optimal strategy. Poolsappasit et al. proposed a multi-index quantitative analysis method based on cost and benefit, and calculated the optimal security protection strategy through genetic algorithm [22], while it is also easy to fall into the local optimum. Yigit et al. developed a network defense algorithm under limited budget conditions [23]. However, the algorithm only considers the minimum cost, does not consider the attack benefit, and lacks comprehensive measurement. Lei et al. developed a Markov game-based strategy selection mechanism for a balance between the defensive revenue and network service quality [24]. However, the method has high time cost. Herold et al. defined the response selection approach based on user-defined cost metrics to counteract security incidents in complex network environments [25]. However, this method does not meet the ICS security requirements well due to the ignorance of the balance between security risks and protection costs. S. A. Butler proposed a multi-attribute risk assessment framework, in which several complex metrics are introduced, such as total security control cost, attack strategy cost, and so on [26,27]. Roy et al. proposed a cost-effective countermeasure system for cyber attacks using various functions to form the objective function [28]. Viduto et al. proposed a new risk assessment and optimization model to solve the problem of security countermeasure selection, in which the total cost of security control and the total risk level of the system together constitute the objective function [29]. In this approach, historical databases are used without consideration for the first experienced or zero-day attacks, exposing the system to undue risk. R. Dewri proposed Genetic Algorithm (GA) and S. Wang et al. proposed Ant Colony Optimization (ACO), which utilizes single-objective and multi-objective optimization cost functions to select the optimal set of strategies [30,31]. Similarly, the indicators used in the methodology are inconsistent with globally accepted standards. B. Kordy et al. proposed a game selection method combining ADTree modeling
with integer linear programming [32,33]. The only response considered in this approach is to fix the vulnerabilities, and the many aspects (risk, cost, budget constraints, etc.) that define the optimal response plan are not considered. Speicher et al. proposed a Stackelberg programming algorithm that models the game choice problem as a two-person game in which the defender applies mitigation strategies (to minimize the probability of a successful attack) and the attacker tries to counter and maximize his chances of successfully executing the attack [34]. However, in order to improve the efficiency, this method only considers the critical attack path. Zenitani et al. proposed a multi-objective cost-benefit optimization algorithm, which can replace the use of multi-objective optimization algorithms and obtain the optimal solution through a series of iterations [35]. However, this method has only been tested in some networks, and its performance in practice cannot be guaranteed.

The above-mentioned research focuses mainly on balancing benefits and costs, and on constructing the objective function, and the research on the algorithm for selecting the optimal protection strategy has not been discussed in-depth. In addition, there are some other problems, such as the use of indicators that are not consistent with globally accepted standards, and so on. Based on the above research, this paper proposes an optimal security protection strategy selection model based on QLPSO. The model mainly answers the following questions: (1) how to evaluate the security risk of the ICS; (2) how to determine the objective function according to the security risk of the ICS and protection cost; (3) how to choose the optimal protection strategy.

3. Model Framework

Our model can be divided into three modules: ICS security risk assessment, construction of objective function, and optimal protection strategy selection. The framework of the model is shown in Figure 1.

Figure 1. Optimal security protection selection model based on QLPSO.

(1) ICS security risk assessment: First, we build a Bayesian attack graph based on the network configuration and asset information of the ICS [36], then calculate the exploit success rate of each edge of the attack graph, and construct the local conditional probability distribution (LCPD) table according to the exploit success rate. Finally, the prior probabilities of all attribute nodes being attacked are calculated [37];
(2) Construction of the objective function: Firstly, we construct all possible protection strategies based on the network configuration and asset information of the ICS, then quantify the attack benefit and protection cost, and finally construct the objective function based on both the attack benefit and protection cost;

(3) Optimal protection strategy selection: First, we design the Q-Learning particle swarm optimization algorithm (QLPSO), then solve the objective function using the QLPSO, and finally find the optimal protection strategy.

4. ICS Security Risk Assessment

4.1. Definition of Bayesian Attack Graph

The Bayesian attack graph is a directed acyclic graph, defined as: \( BAG = (S, E, A, P) \), where

1. \( S = \{ S_1, S_2, \ldots, S_n \} \) is the set of all attribute nodes of the attack graph.
2. \( E = \{ \ldots, E_{ij}, \ldots \} \) is the set of all directed edges of the attack graph, where \( E_{ij} \) has two end nodes \( S_i \) and \( S_j \), and \( S_i \) is the parent node and \( S_j \) the child node.
3. \( A = \{ A_1, A_2, \ldots, A_n \} \) means atomic attack. \( A_i = 1 \) means that the attack has been launched, otherwise \( A_i = 0 \).
4. \( P = \{ P(S_1), P(S_2), \ldots, P(S_n) \} \) is the set of the probabilities that the attribute nodes can be attacked. \( P(S_i) \) indicates the success probability of attribute node \( S_i \) of being attacked.

The Bayesian attack graph is established by combining the network configuration and asset information, and can be directly constructed using the MulVAL tool [38].

4.2. Calculation of Success Probability of Vulnerability Exploitation

We adopt the vulnerability scoring system CVSS (common vulnerability scoring system [39]) to calculate the probability of successful exploitation of each vulnerability, which is given by

\[
p(S_i) = 2 \times AV \times AC \times AU
\]

where \( AV \), \( AC \), and \( AU \) are the CVSS availability indicators. The specific scores of CVSS indicators are shown in Table 1. \( AV \) is the attack route value. The higher the \( AV \), the farther the attacker can launch an attack. \( AC \) is the attack complexity value. The higher the \( AC \) score, the lower the attack complexity of the attacker. \( AU \) is the authentication value. The higher the \( AU \) score, the less authentication times the attacker launches attacks. \( v_i \) represents the exploit between the current node and its parent node.

Table 1. CVSS score.

| Index                      | Rank                  | Score   |
|----------------------------|-----------------------|---------|
| Access Vector (AV)         | local access          | 0.395   |
|                            | adjacent network access| 0.646  |
|                            | network accessible    | 1.0     |
| Access complexity (AC)     | high                  | 0.35    |
|                            | medium                | 0.61    |
|                            | low                   | 0.71    |
| Authentication (AU)        | multiple instances of authentication | 0.45 |
|                            | single instance of authentication | 0.56 |
|                            | no authentication     | 0.704   |

4.3. Calculation of Local Conditional Probability Distribution (LCPD)

Conditional probability [40] represents the successful exploitation probability of an attribute node under the influence of its parent node set, denoted by \( p(S_i \mid Par(S_i)) \). \( Par(S_i) \)
refers to the set of parent nodes of the $S_i$. The relationship between the attribute node and its parent node is shown in Figure 2. $d_i$ represents the respective type of each attribute node, which is divided into two types: AND and OR.

![Figure 2. Attribute node and its parent node dependencies.](Image)

When $d_i = \text{AND}$, attribute node $S_i$ is only exploited when all its parent nodes are exploited,

$$p(S_i | \text{Par}(S_i)) = \prod_{S_j \in \text{Par}(S_i)} p(S_j)$$

When $d_i = \text{OR}$, attribute node $S_i$ can be exploited when any of its parent nodes are exploited,

$$p(S_i | \text{Par}(S_i)) = 1 - \prod_{S_j \in \text{Par}(S_i)} [1 - p(S_j)]$$

### 4.4. Prior Probability Calculation

The prior probability of an attribute node $S_i$ refers to the probability that the attribute node $S_i$ can be reached under static conditions. It is expressed as the joint probability of the attribute node $S_i$ and all attribute nodes in the path to reach the attribute node $S_i$.

The prior probability of attribute node $S_i$ is calculated as follows. When $d_i = \text{AND}$, attribute node $S_i$ is only exploited when all its parent nodes are exploited,

$$P(S_i) = \prod_{S_j \in \text{Par}(S_i)} P(S_j) p(S_i)$$

When $d_i = \text{OR}$, attribute node $S_i$ can be exploited when any of its parent nodes are exploited,

$$P(S_i) = 1 - \prod_{S_j \in \text{Par}(S_i)} [1 - P(S_j)] p(S_i)$$

When the LCPD tables are calculated, the prior probability of each attribute node is equal to the conditional probability of the attribute node multiplied by the conditional probability of its parent node.

### 5. Construction of Objective Function

#### 5.1. Protection Strategy

The $M$ is a set of protection strategies, denoted as $M = \{M_1, M_2, \ldots, M_n\}$, where $M_i$ is a protection strategy that can be operated on the attribute node $S_i$ to reduce its risk of being attacked. $M_i = 1$ represents that the protection strategy is enabled, otherwise $M_i = 0$. When the protection strategy is enabled, the exploit success probability of attribute nodes will be affected, followed by a certain reduction in the exploitation probability. That is

$$p(S_i | M_i = 1) < p(S_i | M_i = 0).$$

As a consequence, we have

$$P(S_i | M \neq 0) \leq P(S_i | M = 0).$$
Implementing protection strategies will inevitably incur protection costs. The protection cost is represented by \( \text{COST} = \{\text{COST}_1, \text{COST}_2, \ldots, \text{COST}_n\} \), where \( \text{COST}_i \) represents the cost of implementing the protection strategy \( M_i \). \( \text{COST}_i \) is defined as:

\[
\text{COST}_i = \omega_i \times \text{value} \times 100
\]

where \( \omega_i \) is the normalized weight of the protection strategies, and \( \text{value} \) represents the value of the asset. Therefore, the total protection cost of implementing the protection strategies \( M = \{M_1, M_2, \ldots, M_n\} \) is given by:

\[
C(M) = \sum_{i=1}^{n} M_i \text{COST}_i.
\]

5.2. Attack Benefit

The attack benefit of attribute node \( S_i \) is expressed as \( AG(S_i) \), which refers to the attack benefit obtained by successfully attacking attribute node \( S_i \). \( AG(S_i) \) can be calculated as:

\[
AG(S_i) = P(S_i) \times \text{value}.
\]

In addition, the attack benefit of attribute node \( S_i \) under the protection strategies \( M \) is given by:

\[
AG(S_i|M) = P(S_i|M) \times \text{value}.
\]

Therefore, the total attack benefit under the protection strategies \( M \) can be obtained by summing the benefits of all attribute nodes, that is:

\[
AG(M) = \sum_{S_i \in S} AG(S_i|M).
\]

5.3. Attack Benefit-Protection Cost Objective Function

Under the above definitions of attack benefit and protection cost, it is easier to illustrate our objective as being to minimize both the attack benefit and the protection cost. Therefore, the objective function can be expressed as:

\[
\min (\delta AG(M) + (1 - \delta)C(M))
\]

subject to

\[
C(M) < B
\]

where \( \delta \) and \( 1 - \delta \) are the preference weights of attack benefit and defense cost, respectively, and \( 0 \leq \delta \leq 1 \). \( B \) is the constraint of total protection cost.

6. Q-Learning Particle Swarm Optimization Algorithm

6.1. Particle Swarm Optimization

In 1995, Kennedy and Eberhardt proposed the PSO algorithm, which is motivated by imitating the foraging activities of birds for solving single-objective optimization problems [42]. The state of each particle \( i \) in the particle swarm has two fundamental properties: velocity \( V_i \) and position \( X_i \), where:

\[
V_i = (V_{i1}, V_{i2}, \ldots, V_{iD}), i = 1, 2, \ldots, N
\]

\[
X_i = (X_{i1}, X_{i2}, \ldots, X_{iD}), i = 1, 2, \ldots, N
\]

Among them, \( V_{ij} \) represents the velocity of the \( i \)th particle in the \( j \)th dimension, and \( X_{ij} \) represents the position of the \( i \)th particle in the \( j \)th dimension. \( D \) is the dimension.

All particles have memory, which enables them to remember their local optimal position \( P \). At the same time, all \( P \) of different particles will be shared in the whole
population, and the optimal $P$ is regarded as the global optimal position $G_p$ reached by the population. Based on the obtained $P$ and $G_p$, all particles in the population are updated in position and velocity using the following equations [43].

$$V_{id}^{t+1} = \text{weight} \times V_{id}^{t} + Lp \times \text{rand}(0,1) \times (P_{id}^{t} - X_{id}^{t})$$
$$+ Lb \times \text{rand}(0,1) \times (G_p^{t} - X_{id}^{t})$$

(17)

$$X_{id}^{t+1} = X_{id}^{t} + V_{id}^{t}$$

(18)

Among them, $weight$ is the inertia weight, $Lp$ is the self-learning factor, $Lb$ is the global learning factor, rand $(0,1)$ is a random number in $[0,1]$. $V_{id}^{t}$ represents the $d$th dimensional velocity of the $i$th particle in the $t$th generation. $P_{id}^{t}$ indicates the local optimal particle of the $i$th particle in the $d$th dimension of the $t$th generation. $G_p^{t}$ represents the globally optimal particle in the $d$th dimension of the $t$th generation. $X_{id}^{t}$ represents the $d$th dimensional position of the $i$th particle in the $t$th generation.

6.2. Q-Learning

Q-Learning is a learning method proposed by Watkins in 1989 [44]. In Q-Learning, an agent learns in a ‘trial and error’ way, and the reward guides behavior obtained by interacting with the environment, and the goal is to make the agent obtain the maximum reward. Q-Learning has four key elements: $Q$ table, state, action, and reward. The process of Q-Learning is [45]: the agent selects the action with the largest $Q$ value from the $Q$ table according to the state. After the action is completed, the state changes, and the reward is determined according to the quality of the state change, and then according to the current state, the next state, actions, and reward values update the $Q$ table, in turn.

The updated method of the $Q$ table is as follows [46].

$$Q(st_{t+1}, at_{t+1}) = (1 - \alpha)Q(st_{t}, at_{t}) + \alpha [R(st_{t}, at_{t}) + \gamma \max_{a_t} Q(st_{t+1}, at)]$$

(19)

where, $st_{t}$ is the current state, $st_{t+1}$ is the next state, $at_{t}$ is the action taken for the current state, and $at_{t+1}$ is the action taken for the next state. $\alpha$ and $\gamma$ are the learning rate and discount factor, respectively, $R(st_{t}, at_{t})$ is the reward generated by the action $at_{t}$ taken by the state $st_{t}$. $\max_{a_t} Q(st_{t+1}, at)$ refers to the maximum $Q$ value in the case of state $st_{t+1}$.

The model of Q-Learning is shown in Figure 3.

![Figure 3. Q-Learning model.](image)

6.3. Q-Learning Particle Swarm Optimization (QLPSO)

The PSO algorithm can be directly used to solve the optimization objective function Equation (13). However, PSO often falls into local optimum due to its fixed parameters. In this paper, we consider the QLPSO algorithm, which could update the parameters of PSO algorithm through the Q-Learning to avoid the local optimum problem [47]. The state, action, $Q$ table, and reward are also the core elements of the QLPSO algorithm, which are shown in Figure 4.
Unlike PSO, which has only one state, QLPSO has two states: objective space state and decision space state. The objective space state needs to consider the relationship between the particle and the global optimal particle position. The decision space needs to consider the relationship between the fitness of the particle and the fitness of the global optimal particle.

The decision space state has four sub-states: DFarthest, DFarther, DNearer, and DNearest. They respectively represent the relative state of the Euclidean distance between the particle and the global optimal position $G_p$ compared with the size of the search space. The objective space state also has four sub-states: the largest fitness difference, the larger fitness difference, the smaller fitness difference, and the smallest fitness difference. They represent the relative state of the particle’s fitness compared to the global optimal fitness and the global worst fitness difference, respectively. In this article, we only need to consider the fitness value between the two solutions.

The specific information of decision space state and objective space state is shown in Tables 2 and 3 [48].

### Table 2. Decision space state.

| Relative Distance (Rd) | Decision Space State |
|-----------------------|----------------------|
| $0 \leq Rd \leq 0.25 \triangle R$ | DNearest |
| $0.25\triangle R \leq Rd \leq 0.5 \triangle R$ | DNearer |
| $0.5\triangle R \leq Rd \leq 0.75 \triangle R$ | DFarther |
| $0.75\triangle R \leq Rd \leq \triangle R$ | DFarthest |

### Table 3. Objective space state.

| Single-Objective Problem Relative Fitness (Rf) | Objective Space State |
|----------------------------------------------|------------------------|
| $0 \leq Rf \leq 0.25 \triangle F$           | FSmallest              |
| $0.25\triangle F \leq Rf \leq 0.5 \triangle F$ | FSmaller               |
| $0.5\triangle F \leq Rf \leq 0.75 \triangle F$ | FLarger                |
| $0.75\triangle F \leq Rf \leq \triangle F$ | FLargest               |

In Table 2, $Rd$ is the Euclidean distance between a certain particle and the global optimal particle $G_p$. $\triangle R$ is the range of decision space search. In Table 3, $Rf$ is the fitness, which refers to the fitness difference between a certain particle and the global optimal
particle \( Gp \). \( \Delta F \) is the difference between the fitness of the global best particle and the fitness of the global worst particle.

(2) Action

There are also four types of actions, which correspond to different parameters of the particle swarm: weight, \( Lp \), and \( Lb \). Different values of weight, \( Lp \) and \( Lb \) will affect the exploration form of the particle. The larger the weight is, the stronger the global exploration ability and the weaker the local exploration ability. On the contrary, the smaller the weight is, the weaker the global exploration ability and the stronger the local exploration ability. The larger the \( Lp \), the stronger the global exploration ability. The larger the \( Lb \), the stronger the particle convergence ability [49]. The detailed parameter settings are shown in Table 4.

Table 4. Detailed parameter settings of the actions.

| Search Status      | Weight | \( Lp \) | \( Lb \) |
|--------------------|--------|---------|---------|
| Large scale search | 1      | 0.5     | 0.5     |
| Small scale search | 0.8    | 2       | 1       |
| Slow search        | 0.6    | 1       | 2       |
| Fast search        | 0.4    | 0.5     | 0.5     |

(3) \( Q \) table

Since there are four types of objective space states, decision space states, and actions, the \( Q \) table of QLPSO is different from the two-dimensional \( Q \) table used in general Q-Learning. The three-dimensional \( Q \) table used here is a \( 4 \times 4 \times 4 \) three-dimensional \( Q \) table. The three-dimensional \( Q \) table is shown in Figure 5. As shown in the figure, we first determine the state of the objective space and the state of the decision space, such as (the closest distance, the smallest fitness difference). Then, according to the state of the objective space and the state of the decision space, the action with the largest \( Q \) value corresponding to the state is selected.

Figure 5. Three-dimensional \( Q \) table.

(4) Rewards

After selecting a certain action, if the fitness value becomes worse, it should be punished. Otherwise, if the fitness value becomes better, it should be rewarded. The reward function defined in this paper is as follows.

\[
R = Rf(state + 1) - Rf(state)
\]
where $R_f(state)$ and $R_f(state + 1)$ represent the fitness values of the current state and the next state, respectively.

The flowchart of the QLPSO algorithm is shown in Figure 6 and described as follows.

1. Initialize the population and $Q$ table;
2. The state of each particle is determined according to its position in the objective space and decision space;
3. The action (parameter) of the particle is determined using the $Q$ table;
4. Update the particles according to the parameters determined in the previous step;
5. Update the $Q$ table according to the reward function;
6. These steps are repeated for all particles in each generation until the number of iterations is reached.

![Figure 6. Flowchart of the QLPSO algorithm.](image)

The QLPSO algorithm is shown below (Algorithm 1). The time complexity and space complexity of the proposed algorithm are $PN \times maxIterations$ and $PN$, respectively, where $PN$ is the population number, and $maxIterations$ is the maximum number of iterations.

### Algorithm 1: The description of QLPSO

**Input:** population $PN$, maximum number of iterations $maxIterations$

**Output:** optimal protection strategy

1. initialize population, $M = \{M_1, M_2, \ldots, M_n\}$ and $Q$ table
2. initialize $P$ and $Gp$
3. while $t \leq maxIterations$ do
4.   while $i \leq PN$ do
5.     decide the state in two spaces
6.     determine the action using $Q$ table
7.     update $V$ and $X$
8.     update $P$ and $Gp$
9.     evaluate the performance
10.    update $Q$ table using reward function
11.    $i++$
12.  end
13. $t++$
14. end
7. Experimental

7.1. Experimental Scenario

The experimental scenario in this paper is a water distribution system, as shown in Figure 7. There are three PLCs in the water distribution system, which respectively control the water pump, the pipeline, and the water tanker in the virtual system. The platform can be used for industrial control security attack and defense drills, security risk assessment, and other related experiments [50].

![Experimental scene](image)

Figure 7. Experimental scene.

7.2. Generating a Bayesian Attack Graph

Asset and vulnerability information is obtained by scanning assets as shown in Figure 8. The asset consists of three PLCs, each of which has the same vulnerability CVE-1999-0517. At the same time, there is some additional information for each PLC. The attack graph of the water distribution system is generated by using the MulVal. It is shown in Figure 9 and Table 5.

![Asset and vulnerability information](image)

Figure 8. Asset and vulnerability information.

| Host      | Vulnerabilities |
|-----------|-----------------|
| 192.168.0.1 | 1               |
| 192.168.0.2 | 1               |
| 192.168.0.10 | 1              |

| Vulnerabilities |
|-----------------|
| 16              |
| 25              |
| 17              |
Figure 9. Bayesian attack graph.

Table 5. Prior probabilities of each node of the Bayesian attack graph.

| Attribute Nodes | Prior Probability | Attribute Nodes | Prior Probability | Attribute Nodes | Prior Probability |
|-----------------|-------------------|-----------------|-------------------|-----------------|-------------------|
| $S_1$           | 0.5000            | $S_2$           | 0.1812            | $S_3$           | 0.5000            |
| $S_4$           | 0.5000            | $S_5$           | 1.0000            | $S_6$           | 1.0000            |
| $S_7$           | 0.3967            | $S_8$           | 0.5000            | $S_9$           | 1.0000            |
| $S_{10}$        | 1.0000            | $S_{11}$        | 1.0000            | $S_{12}$        | 1.0000            |
| $S_{13}$        | 0.8372            | $S_{14}$        | 0.3257            | $S_{15}$        | 0.5000            |
| $S_{16}$        | 1.0000            | $S_{17}$        | 1.0000            | $S_{18}$        | 0.5000            |
| $S_{19}$        | 1.0000            | $S_{20}$        | 0.6793            | $S_{21}$        | 1.0000            |
| $S_{22}$        | 1.0000            | $S_{23}$        | 1.0000            | $S_{24}$        | 0.9863            |
| $S_{25}$        | 0.8000            | $S_{26}$        | 1.0000            | $S_{27}$        | 1.0000            |
| $S_{28}$        | 0.9960            | $S_{29}$        | 0.5814            | $S_{30}$        | 1.0000            |
| $S_{31}$        | 0.5641            | $S_{32}$        | 1.0000            | $S_{33}$        | 1.0000            |
| $S_{34}$        | 0.9530            |                 |                   |                 |                   |

7.3. Attack Benefits and Protection Costs

The prior probabilities of all attribute nodes are calculated through the LCPD table, as shown in Table 6. The quantification standard of cost index of protection operation is shown in Table 7 [34]. In Table 7, $P_i$ refers to the impact of the implementation strategy on vulnerability utilization. The specific operation, cost and other details of the protection strategies are shown in Table 8.
### Table 6. Details of the attack graph.

| Node | Node Information                                      |
|------|--------------------------------------------------------|
| 1    | execCode(192.168.0.1, someUser): 0.3623               |
| 2    | RULE 2 (remote exploit of a server program): 0.3623    |
| 3    | netAccess(192.168.0.1, udp, 161): 0.906               |
| 4    | RULE 5 (multi-hop access): 0.3173                      |
| 5    | hacl(192.168.0.10, 192.168.0.1, udp, 161): 1.0        |
| 6    | execCode(192.168.0.10, someUser): 0.3967             |
| 7    | RULE 2 (remote exploit of a server program): 0.3967    |
| 8    | netAccess(192.168.0.10, udp, 161): 0.992              |
| 9    | RULE 5 (multi-hop access): 0.8                        |
| 10   | hacl(192.168.0.1, 192.168.0.10, udp, 161): 1.0        |
| 11   | RULE 5 (multi-hop access): 0.8                        |
| 12   | hacl(192.168.0.2, 192.168.0.10, udp, 161): 1.0        |
| 13   | execCode(192.168.0.2, someUser): 0.389               |
| 14   | RULE 2 (remote exploit of a server program): 0.389    |
| 15   | netAccess(192.168.0.2, udp, 161): 0.9727              |
| 16   | RULE 5 (multi-hop access): 0.8                        |
| 17   | hacl(192.168.0.1, 192.168.0.2, udp, 161): 1.0        |
| 18   | RULE 5 (multi-hop access): 0.3173                      |
| 19   | hacl(192.168.0.10, 192.168.0.2, udp, 161): 1.0        |
| 20   | RULE 6 (direct network access): 0.8                    |
| 21   | hacl(internet, 192.168.0.2, udp, 161): 1.0            |
| 22   | attackerLocated(internet): 1.0                        |
| 23   | networkServiceInfo(192.168.0.2, sun, sunos, udp, 161, someUser): 1.0 |
| 24   | vulExists(192.168.0.2, CVE-1999-0517, sun, remoteExploit, privEscalation): 0.4998 |
| 25   | RULE 6 (direct network access): 0.8                    |
| 26   | hacl(internet, 192.168.0.10, udp, 161): 1.0           |
| 27   | networkServiceInfo(192.168.0.10, sun, sunos, udp, 161, someUser): 1.0 |
| 28   | vulExists(192.168.0.10, CVE-1999-0517, sun, remoteExploit, privEscalation): 0.4998 |
| 29   | RULE 5 (multi-hop access): 0.3112                      |
| 30   | hacl(192.168.0.2, 192.168.0.1, udp, 161): 1.0         |
| 31   | RULE 6 (direct network access): 0.8                    |
| 32   | hacl(internet, 192.168.0.1, udp, 161): 1.0            |
| 33   | networkServiceInfo(192.168.0.1, sun, sunos, udp, 161, someUser): 1.0 |
| 34   | vulExists(192.168.0.1, CVE-1999-0517, sun, remoteExploit, privEscalation): 0.4998 |

### Table 7. Protection cost quantification standard.

| Index              | Weight ($\omega_i$) |
|--------------------|----------------------|
| Disable cost       | 0.357                |
| Disconnect cost    | 0.286                |
| Patch cost         | 0.214                |
| Install cost       | 0.143                |
7.4. Experimental Results

In this paper, assuming the benefits of attack are as important as the costs of protection. The fitness function parameter $\delta$ is set to 0.5, and there is no upper limit to the set-up cost $B$.

### Table 8. Detailed parameters of the protective strategies.

| Protective Strategies | Protective Action                        | COST | $Pi$ |
|-----------------------|------------------------------------------|------|------|
| $M_1$                 | Disconnect 192.168.0.1–192.168.0.10      | 14   | 0.25 |
| $M_2$                 | Disconnect 192.168.0.2–192.168.0.10      | 14   | 0.25 |
| $M_3$                 | Disconnect Internet 192.168.0.10         | 15   | 0.30 |
| $M_4$                 | Disconnect Internet 192.168.0.2          | 15   | 0.30 |
| $M_5$                 | Disconnect Internet                      | 22   | 0.05 |
| $M_6$                 | Disable Internet                         | 22   | 0.05 |
| $M_7$                 | Disable Internet direct network access   | 22   | 0.05 |
| $M_8$                 | Disable Internet 192.168.0.1             | 15   | 0.30 |
| $M_9$                 | Disable multi-hop access 192.168.0.10    | 12   | 0.10 |
| $M_{10}$              | Disable udp                              | 12   | 0.10 |
| $M_{11}$              | Disable direct network access 192.168.0.10 | 12   | 0.10 |
| $M_{12}$              | Disable direct network access 192.168.0.2 | 12   | 0.10 |
| $M_{13}$              | Disable direct network access 192.168.0.1 | 12   | 0.10 |
| $M_{14}$              | Disable netAccess 192.168.0.10           | 12   | 0.10 |
| $M_{15}$              | Disable networkService 192.168.0.10      | 18   | 0.45 |
| $M_{16}$              | Patch CVE-1999-0517 192.168.0.10         | 18   | 0.45 |
| $M_{17}$              | Disable service programs                  | 25   | 0.20 |
| $M_{18}$              | Disconnect 192.168.0.10–192.168.0.1      | 14   | 0.25 |
| $M_{19}$              | Disable execCode 192.168.0.10            | 20   | 0.30 |
| $M_{20}$              | Disconnect 192.168.0.10                  | 20   | 0.30 |
| $M_{21}$              | Disconnect 192.168.0.10–192.168.0.2      | 14   | 0.25 |
| $M_{22}$              | Disconnect 192.168.0.1–192.168.0.2       | 14   | 0.25 |
| $M_{23}$              | Disable multi-hop access 192.168.0.1     | 20   | 0.30 |
| $M_{24}$              | Disable multi-hop access 192.168.0.2     | 20   | 0.30 |
| $M_{25}$              | Disable netAccess 192.168.0.2            | 20   | 0.30 |
| $M_{26}$              | Disable netAccess                        | 12   | 0.10 |
| $M_{27}$              | Disable networkService 192.168.0.2       | 14   | 0.25 |
| $M_{28}$              | Patch CVE-1999-0517 192.168.0.2          | 18   | 0.45 |
| $M_{29}$              | Disable execCode 192.168.0.2             | 25   | 0.20 |
| $M_{30}$              | Disable multi-hop access                 | 20   | 0.30 |
| $M_{31}$              | Disconnect 192.168.0.2–192.168.0.1      | 14   | 0.25 |
| $M_{32}$              | Disable netAccess 192.168.0.1            | 12   | 0.10 |
| $M_{33}$              | Disable service program 192.168.0.1      | 12   | 0.10 |
| $M_{34}$              | Disable networkService 192.168.0.1       | 18   | 0.45 |
| $M_{35}$              | Patch CVE-1999-0517 192.168.0.1          | 18   | 0.45 |
| $M_{36}$              | Install ids                              | 30   | 0.20 |
The experimental results are shown in Figure 10. The blue curve is the experimental result without protection strategies. The fitness value is always at 546.0160, and the attack benefit is very large, which means that without protection strategies, the attacker can very easily launch attacks to obtain attack benefit. The red curve is the experimental result of the ordinary particle swarm. It can be seen that the fitness value is always 430.0326, and the attack benefit plus protection cost is still quite large, indicating that the optimal protection strategy has not been obtained. The yellow curve is the experimental result of the Q-Learning particle swarm. It can be seen that the fitness value tends to be stable at 327.9708, and the attack benefit plus protection cost reaches the minimum value. The experiments show that the Q-Learning particle swarm algorithm found the optimal protection strategy \([0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]\), and the attack benefit plus protection cost reached a minimum of 332.4039.

![Experimental results](image)

**Figure 10.** Experimental results.

It is proved that the optimal protection strategy obtained by QLPSO algorithm has a good protection effect and minimizes the attack benefit and protection cost. Therefore, the water distribution system is well protected by the strategy selected through QLPSO.

8. Conclusions

Constructing the optimal security protection strategy is of great significance to the ICS. Since the traditional PSO algorithm tends to fall into local optimum when choosing the optimal protection strategy, in this paper, we propose an optimal security protection strategy selection model and Q-Learning particle swarm optimization algorithm. The algorithm can easily select the optimal protection strategy and the experimental results verify the feasibility and effectiveness of our model and the QLPSO algorithm.

**Author Contributions:** Methodology, X.G., Y.Z. and D.Z.; Formal analysis, X.G. and Y.Z.; Resources, L.X.; Writing—original draft, X.G.; Writing—review & editing, L.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Major Program for Technological Innovation 2030-New Generation Artificial Intelligence (2020AAA0107700), the National Natural Science Foundation of China (62172244), the Shandong Provincial Natural Science Foundation (ZR2020YQ06, ZR2021MF132), the Young innovation team of colleges and universities in Shandong province (2021KJ001), the Pilot Project for Integrated Innovation of Science, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) (2022JBZ01-01).
Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Brändle, M.; Naedele, M. Security for process control systems: An overview. *IEEE Secur. Priv.* **2008**, *6*, 24–29. [CrossRef]

2. Fan, X.; Fan, K.; Wang, Y.; Zhou, R. Overview of cyber-security of industrial control system. In Proceedings of the 2015 international conference on cyber security of smart cities, industrial control system and communications (SSIC), Shanghai, China, 5–7 August 2015; pp. 1–7.

3. Wilhoit, K. Who’s really attacking your ICS equipment? *Trend Micro* **2013**, *10*.

4. Clarke, G.; Reynolds, D.; Wright, E. *Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems*; Elsevier: Newnes, Australia, 2004.

5. van Schuppen, J.H.; Boutin, O.; Kempker, P.L.; Komenda, J.; Masopust, T.; Pambakian, N.; Ran, A.C. Control of distributed systems: Tutorial and overview. *Eur. J. Control* **2011**, *17*, 579–602. [CrossRef]

6. Babu, B.; Ijyas, T.; Muneer, P.; Varghese, J. Security issues in SCADA based industrial control systems. In Proceedings of the 2017 2nd International Conference on Anti-Cyber Crimes (ICACC), Abha, Saudi Arabia, 26–27 March 2017; pp. 47–51.

7. Wang, Y. SCM/ERP/MES/PCS integration for process enterprise. In Proceedings of the 29th Chinese Control Conference, Beijing, China, 29–31 July 2010; pp. 5329–5332.

8. Li, Z.; Shahidehpour, M.; Aminifar, F. Cybersecurity in distributed power systems. *Proc. IEEE* **2017**, *105*, 1367–1388. [CrossRef]

9. Cruz, T.; Rosa, L.; Proença, J.; Maglaras, L.; Aubigny, M.; Lev, L.; Jiang, J.; Simões, P. A cybersecurity detection framework for supervisory control and data acquisition systems. *IEEE Trans. Ind. Inform.* **2016**, *12*, 2236–2246. [CrossRef]

10. Chen, T.M.; Abu-Nimeh, S. Lessons from stuxnet. *IEEE Trans. Ind. Inform.* **2011**, *7*, 44, 91–93. [CrossRef]

11. Sun, C.C.; Hahn, A.; Liu, C.C. Cyber security of a power grid: State-of-the-art. *Int. J. Electr. Power Energy Syst.* **2018**, *99*, 45–56. [CrossRef]

12. Zhao, D.; Wang, L.; Wang, Z.; Xiao, G. Virus propagation and patch distribution in multiplex networks: Modeling, analysis, and optimal allocation. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1755–1767. [CrossRef]

13. Stouffer, K.; Falco, J.; Scarfone, K. Guide to industrial control systems (ICS) security. *NIST Spec. Publ.* **2011**, *800*, 16.

14. David, A. *Multiple Efforts to Secure Control Systems Are under Way, But Challenges Remain*; Technical report; US Government Accountability Office (US GAO): Washington DC, USA, 2007.

15. Zhao, D.; Xiao, G.; Wang, Z.; Wang, L.; Xu, L. Minimum dominating set of multiplex networks: Definition, application, and identification. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 7823–7837. [CrossRef]

16. Shameli-Sendi, A.; Desfossez, J.; Dagenais, M.; Jabbarifar, M. A Retroactive-Burst Framework for Automated Intrusion Response System. *J. Comput. Netw. Commun.* **2013**, *2013*, 134760.1–134760.8. [CrossRef]

17. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]

18. Gonzalez-Granadillo, G.; Garcia-Alfaró, J.; Alvarez, E.; El-Barbori, M.; Debar, H. Selecting optimal countermeasures for attacks against critical systems using the attack volume model and the RORI index. *Comput. Electr. Eng.* **2015**, *47*, 13–34. [CrossRef]

19. Miehling, E.; Rasouli, M.; Teneketzis, D. Optimal defense policies for partially observable spreading processes on Bayesian attack trees. In Proceedings of the 2015 international conference on cyber security of smart cities, industrial control system and communications (SSIC), Shanghai, China, 5–7 August 2015; pp. 1–7.

20. Herold, N.; Wachs, M.; Posselt, S.A.; Carle, G. An optimal metric-aware response selection strategy for intrusion response systems. In *Proceedings of the International Symposium on Foundations and Practice of Security*, Quebec City, QC, Canada, 24–25 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 68–84.

21. Butler, S.A. Security attribute evaluation method: A cost-benefit approach. In *Proceedings of the 24th International Conference on Software Engineering*, Orlando, FL, USA, 19–25 May 2002; pp. 232–240.

22. Butler, S.A.; Fischbeck, P. Multi-attribute risk assessment. In *Proceedings of the Symposium on Requirements Engineering for Information Security*, Raleigh, NC, USA, 16 October 2002; Volume 2.

23. Roy, A.; Kim, D.S.; Trivedi, K.S. Scalable optimal countermeasure selection using implicit enumeration on attack countermeasure trees. In *Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, Boston, MA, USA, 25–28 June 2012; pp. 1–12.
29. Viduto, V.; Maple, C.; Huang, W.; López-Peréz, D. A novel risk assessment and optimisation model for a multi-objective network security countermeasure selection problem. *Decis. Support Syst.* 2012, 53, 599–610. [CrossRef]

30. Dewri, R.; Ray, I.; Poolsappasit, N.; Whitley, D. Optimal security hardening on attack tree models of networks: A cost-benefit analysis. *Int. J. Inf. Secur.* 2012, 11, 167–188. [CrossRef]

31. Wang, S.; Zhang, Z.; Kadobayashi, Y. Exploring attack graph for cost-benefit security hardening: A probabilistic approach. *Comput. Secur.* 2013, 32, 158–169. [CrossRef]

32. Kordy, B.; Wideł, W. How well can I secure my system? In Proceedings of the International Conference on Integrated Formal Methods, Turin, Italy, 20–22 September 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 332–347.

33. Fila, B.; Wideł, W. Exploiting attack–defense trees to find an optimal set of countermeasures. In Proceedings of the 2020 IEEE 33rd Computer Security Foundations Symposium (CSF), Boston, MA, USA, 22–26 June 2020; pp. 395–410.

34. Speicher, P.; Steinmetz, M.; Kinnemann, R.; Simeonovski, M.; Pellegrino, G.; Hoffmann, J.; Backes, M. Formally reasoning about the cost and efficacy of securing the email infrastructure. In Proceedings of the 2018 IEEE European Symposium on Security and Privacy (EuroS&P), London, UK, 24–26 April 2018; pp. 77–91.

35. Zenitani, K. A multi-objective cost–benefit optimization algorithm for network hardening. *Int. J. Inf. Secur.* 2022, 21, 813–832. [CrossRef]

36. Frigault, M.; Wang, L. Measuring network security using bayesian network-based attack graphs. In Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference, Turku, Finland, 28 July–1 August 2008; pp. 698–703.

37. Liu, Y.; Man, H. Network vulnerability assessment using Bayesian networks. In Proceedings of the Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005, Orlando, FL, USA, 28–29 March 2005; Volume 5812, pp. 61–71.

38. Ou, X.; Govindavajhala, S.; Appel, A.W. MulVAL: A Logic-based Network Security Analyzer. In Proceedings of the USENIX Security Symposium, Baltimore, MD, USA, 31 July–5 August 2005; Volume 8, pp. 113–128.

39. mell, P.; Scarnone, K.; Romanosky, S. Common vulnerability scoring system. *IEEE Secur. Priv.* 2006, 4, 85–89. [CrossRef]

40. Gao, N.; Gao, L.; He, Y.; Lei, Y.; Gao, Q. Dynamic security risk assessment model based on Bayesian attack graph. *J. Sichuan Univ. (Eng. Sci. Ed.)* 2016, 48, 111–118.

41. Gao, N.; Gao, L.; Yiyue, H.E.; Wang, F. Optimal security hardening measures selection model based on Bayesian attack graph. *Comput. Eng. Appl.* 2016, 52, 125–130.

42. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN’95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.

43. Clerc, M. *Particle Swarm Optimization*; John Wiley & Sons: Hoboken, NJ, USA, 2010; Volume 93.

44. Jang, B.; Kim, M.; Hariarima, G.; Kim, J.W. Q-learning algorithms: A comprehensive classification and applications. *IEEE Access* 2019, 7, 133653–133667. [CrossRef]

45. Watkins, C.J.; Dayan, P. Q-learning. *Mach. Learn.* 1992, 8, 279–292. [CrossRef]

46. Clifton, J.; Labor, E. Q-learning: Theory and applications. *Annu. Rev. Stat. Its Appl.* 2020, 7, 279–301. [CrossRef]

47. Liu, Y.; Lu, H.; Cheng, S.; Shi, Y. An adaptive online parameter control algorithm for particle swarm optimization based on reinforcement learning. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 815–822.

48. Abed-Alguni, B.H.; Paul, D.J.; Chalup, S.K.; Henskens, F.A. A comparison study of cooperative Q-learning algorithms for independent learners. *Int. J. Artif. Intell.* 2016, 14, 71–93.

49. Meerza, S.I.A.; Islam, M.; Uzzal, M.M. Q-learning based particle swarm optimization algorithm for optimal path planning of swarm of mobile robots. In Proceedings of the 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 3–5 May 2019; pp. 1–5.

50. Xu, L.; Wang, B.; Wu, X.; Zhao, D.; Zhang, L.; Wang, Z. Detecting Semantic Attack in SCADA System: A Behavioral Model Based on Secondary Labeling of States-Duration Evolution Graph. *IEEE Trans. Netw. Sci. Eng.* 2021, 9, 703–715. [CrossRef]