

Article

Protection Strategy Selection Model Based on Genetic Ant Colony Optimization Algorithm

Xinzhan Li, Yang Zhou *, Xin Li, 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: Industrial control systems (ICS) are facing an increasing number of sophisticated and damaging multi-step attacks. The complexity of multi-step attacks makes it difficult for security protection personnel to effectively determine the target attack path. In addition, most of the current protection models responding to multi-step attacks have not deeply studied the protection strategy selection method in the case of limited budget. Aiming at the above problems, we propose a protection strategy selection model based on the Genetic Ant Colony Optimization Algorithm. The model firstly evaluates the risk of ICS through the Bayesian attack graph; next, the target attack path is predicted from multiple angles through the maximum probability attack path and the maximum risk attack path; and finally, the Genetic Ant Colony Optimization Algorithm is used to select the most beneficial protection strategy set for the target attack path under limited budget. Compared with the Genetic Algorithm and Ant Colony Optimization Algorithm, the Genetic Ant Colony Optimization Algorithm proposed in this paper can handle the local optimal problem well. Simulation experiments verify the feasibility and effectiveness of our proposed model.



Citation: Li, X.; Zhou, Y.; Li, X.; Xu, L.; Zhao, D. Protection Strategy Selection Model Based on Genetic Ant Colony Optimization Algorithm. *Mathematics* **2022**, *10*, 3938. <https://doi.org/10.3390/math10213938>

Academic Editor: António Lopes

Received: 23 September 2022

Accepted: 20 October 2022

Published: 24 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: Bayesian attack graph; genetic algorithm; ant colony optimization algorithm; reinforcement learning; protection strategy

MSC: 68W50

1. Introduction

Industrial control system is a general term for a type of control system used in industrial production, and it is also the core of infrastructure [1,2]. It is widely used in electric power, transportation, water conservancy, chemical industry and other fields. Modern industrial control systems no longer operate in isolation but use computer technologies such as the Internet to enhance business processes. The result of this development is an increased risk of attack on industrial control systems [3]. Industrial control systems have vulnerabilities with different threat levels [4,5]. Attackers use these vulnerabilities to launch targeted attacks, which cause serious consequences such as property damage and environmental pollution. For example, in 2000, the newly built Maroochy sewage treatment plant in Queensland, Australia was attacked by a network, which caused untreated sewage to be discharged into waterways and parks [6]. In 2015, a power grid in Ukraine was attacked by a cyber-attack, which caused large-scale power outages in the relevant area [7]. The above cases remind us that we need to do a good job with security protection for industrial control systems.

In such security incidents, the attack behavior is multi-step, which makes it difficult for security engineers to determine the target attack path. In addition, many small- and medium-sized enterprises are often limited in the funding they can provide in choosing a set of protection strategies. Therefore, how to choose the most beneficial protection strategy set

for industrial control system under the condition of limited budget has extensive practical significance.

It is an NP-hard problem to select a set of protection strategies that are most beneficial and cannot exceed the budget from the set of protection strategies to be selected. Heuristic algorithms can find near-optimal solutions to such problems in a reasonable amount of time. Therefore, many researchers use heuristic algorithms to select the most beneficial protection strategies for industrial control systems. Among them, Dewri et al. [8] first quantified the potential damage in the system and the cost of implementing a set of protection strategies, and then used the Genetic Algorithm (GA) [9] to select the most beneficial protection strategies. Wang et al. [10] first used the attack graph to evaluate the risk of the system, and then used the Ant Colony Optimization Algorithm (ACO) [11] to select the most beneficial protection strategies. The Genetic Algorithm used by Dewri et al. simulates the evolutionary mechanism of life. Selection operation, crossover operation and mutation operation make Genetic Algorithms have the ability of global optimization. However, the algorithm is greatly affected by the initial population. If the initial population is not selected properly, the algorithm is likely to fall into local optimum. The Ant Colony Optimization Algorithm used by Wang et al. has a pheromone positive feedback mechanism. This mechanism makes the Ant Colony Optimization Algorithm have a better convergence speed. However, Ant Colony Optimization Algorithm is highly dependent on pheromones, which makes it easy to fall into local optimum.

In recent years, for the improvement of heuristic algorithm, many researchers dynamically select a set of appropriate parameters for the heuristic algorithm through reinforcement learning to improve the performance of the algorithm. Liu et al. [12] used the Q-learning algorithm to adjust the parameters of Particle Swarm Optimization Algorithm. Huynh et al. [13] used the Q-learning algorithm to adjust the parameters of Differential Evolution Algorithm. Reinforcement learning improves the convergence speed of the algorithm greatly, but the improvement of the convergence accuracy is limited. Therefore, we need an improved algorithm that can greatly improve the convergence speed and convergence accuracy.

Considering the global optimization ability of Genetic Algorithms and the fast convergence ability of the Ant Colony Optimization Algorithm, this paper combines a Genetic Algorithm and Ant Colony Optimization Algorithm to construct Genetic Ant Colony Optimization Algorithm and uses reinforcement learning to dynamically select the relevant parameters of the algorithm. In the Genetic Ant Colony Optimization Algorithm, reinforcement learning dynamically selects the relevant parameters of the ant colony module and the genetic module, the ant colony module provides the initial population for the genetic module, and the genetic module updates the pheromone of the ant colony module. This mechanism enables the Genetic Ant Colony Optimization Algorithm to quickly and accurately select the most beneficial set of protection strategies for industrial control systems under limited budget conditions.

Before choosing protection strategies, we need to evaluate the risk status of the system and predict the target attack path based on the evaluation results. Then, we select the most beneficial set of protection strategies that do not exceed the budget for the target attack path. In terms of risk assessment, vulnerability assessment criteria directly affect the accuracy and objectivity of assessment. At present, the most compatible and widely used vulnerability assessment system is the Common Vulnerability Scoring System (CVSS) [14]. Each scoring element of the system is jointly formulated by experts in the field of international information security, so it has strong professionalism and persuasion. Therefore, this paper uses CVSS to calculate the probability of atomic attack and combines the Bayesian attack graph to calculate the probability of each attribute node being attacked, so as to grasp the overall risk status. In terms of predicting the target attack path, we propose to predict the target attack path from two aspects: the maximum probability attack path and the maximum risk attack path. Among them, the maximum probability attack path is regarded as the target attack path with the lowest attack difficulty. On the basis of the maximum

probability attack path, we comprehensively consider the attack probability and attack benefit to calculate the maximum risk attack path. Additionally, the maximum risk attack path is regarded as the target attack path with the greatest attack damage. Then, we use the Genetic Ant Colony Optimization Algorithm to select the most beneficial protection strategy set that does not exceed the budget for the target attack path. Finally, we tested the performance of the Genetic Ant Colony Optimization Algorithm many times under the conditions of different population sizes. The test results are compared with the Genetic Algorithm, Ant Colony Optimization Algorithm, Reinforcement learning-based Particle Swarm Optimization Algorithm [12] and Reinforcement learning-based Differential Evolution Algorithm [13]. The test results show that the convergence accuracy of the Genetic Ant Colony Optimization Algorithm is better than other comparison algorithms in both small-scale populations and large-scale populations. The main contributions of this paper are as follows:

- Assess the risk of ICS using the Common Vulnerability Scoring System and the Bayesian attack graph.
- The target attack path is predicted by the combination of the maximum probability attack path and the maximum risk attack path.
- Combining the advantages of the Genetic Algorithm and Ant Colony Optimization Algorithm, the optimal protection strategy selection problem under limited budget is calculated.
- Reinforcement learning is used to dynamically select relevant parameters of the Genetic Ant Colony Optimization Algorithm.

In other words, we select the most beneficial protection strategy set to prevent multi-step attacks on the target attack path through an improved algorithm. The remainder of this paper is organized as follows: Section 2 discusses related work. Section 3 describes the various components of the proposed model. Section 4 presents a risk assessment for ICS. Section 5 predicts the target attack path. Section 6 introduces the method for selecting the optimal protection strategy set. Section 7 presents the experimental results. Finally, we conclude this paper in Section 8. The Appendix A provides all the parameters in this paper and their descriptions through the nomenclature table.

2. Related Work

In this section, we review research related to risk assessment and optimization algorithms. Meng et al. [15] proposed a two-stage differential evolution algorithm with novel parameter control. In the first stage, a mutation strategy based on historical solutions is adopted. In the second stage, a mutation strategy based on inferior solutions is adopted. This method improves the convergence accuracy and convergence speed of the algorithm. However, the influence of the initial population size on the algorithm is not fully considered. Zamani et al. [16] propose a quantum-based avian navigation optimizer algorithm (QANA) for solving large-scale global optimization problems. The QANA algorithm improves the accuracy and speed of convergence by introducing long-term and short-term memories, a V-echelon communication topology and quantum-based navigation strategies. Nadimi-Shahraki et al. [17] propose an improved gray wolf optimization algorithm for global optimization and engineering design problems. The improved gray wolf optimization algorithm adopts a dimensional learning-based hunting strategy. This strategy maintains the diversity between local search and global search by building a neighborhood for each wolf, which improves the convergence accuracy of the algorithm. Zamani et al. [18] propose a starling murmuration optimizer (SMO) algorithm to solve continuous and engineering optimization tasks. The SMO algorithm introduces a dynamic multi-flock construction and three new search strategies, including separating, diving and whirling. This mechanism improves the convergence accuracy and convergence speed of the algorithm. Chakraborty et al. [19] propose an improved whale optimization algorithm for solving large-scale optimization problems. The improved whale optimization algorithm introduces a new selection parameter β and modifies the coefficient vectors A and C to

improve the convergence accuracy of the algorithm. However, the search process of the algorithm relies on random values, which affects the convergence speed of the algorithm. Wu et al. [20] proposed a deception defense performance evaluation method based on a dynamic Bayesian attack graph, and used cumulative probability to calculate the target attack path, but did not consider the impact of attack benefit on the attack path. Zhang et al. [21] proposed a flexible attack graph generation algorithm based on a graph data model, and predicted the target attack path from the perspective of asset value, but did not consider the impact of attack difficulty on the attack path. Xu et al. [22] proposed a method to assess the risk level of node vulnerability based on the attack graph, and designed different vulnerability scanning cycles for nodes with different risk levels according to the assessment results, but did not further analyze the target attack paths that attackers might take. Yang et al. [23] proposed a method combining absorption Markov chain and attack graph to predict the attack path, and used the Particle Swarm Optimization Algorithm to select the optimal protection strategy, but did not consider the impact of attack benefit on the attack path and did not discuss the local optimum problem of the Particle Swarm Optimization Algorithm. Boudermine et al. [24] proposed a method to generate an attack graph that considers the dynamic behavior of the system to identify new attack paths, and to evaluate the security of the system by measuring the probability of different components changing over time, but did not analyze the target attack paths that an attacker might take. Wang et al. [25] proposed a method to calculate the risk probability of each node in the attack graph using the Common Vulnerability Scoring System (CVSS), and on this basis, used the cumulative probability method to evaluate the risk status of the nodes on the attack path, but did not consider the impact of attack gains on attack paths. Poolsappasit et al. [26] proposed a Bayesian network-based risk management framework, and used the Genetic Algorithm to calculate the optimal protection strategy, but this method is prone to fall into local optimum. Liu et al. [27] proposed a state attack–defense graph model, and based on this, the possible attack paths were obtained, and then the protection strategy was given according to the vulnerability of the attack path, but the cost and benefit of the protection strategy were not considered. Zukhri et al. [28] proposed to combine the evolution steps of the Genetic Algorithm into the Ant Colony Optimization Algorithm, so as to solve the traveling salesman problem more efficiently. Based on this, this paper improves the algorithm to solve the problem of optimal protection strategy selection under limited budget.

3. Model Structure

The protection strategy selection model based on the Genetic Ant Colony Optimization Algorithm is shown in Figure 1, which is divided into three parts.

- ICS risk assessment: Firstly, a Bayesian attack graph is generated based on the detected vulnerability information and asset information. Then, the atomic attack probability of each edge in the attack graph is calculated according to the Common Vulnerability Scoring System, and the local conditional probability of the attribute node is calculated according to the atomic attack probability. Finally, the unconditional probability of the attribute node is calculated according to the above probability, so as to evaluate the risk of ICS.
- Predict target attack paths: This paper predicts the target attack path from two perspectives. One is the probability angle, which calculates the maximum probability attack path according to the unconditional probability of the attribute node, so as to predict the target path with the lowest attack difficulty. The other is the angle of probability and profit, which calculates the maximum risk attack path according to the unconditional probability of the attribute node and the attack benefit, so as to predict the target path with the most damage.
- Select the optimal protection strategy set: Firstly, the Genetic Ant Colony Optimization Algorithm (GACO) is constructed by combining the Genetic Algorithm (GA), Ant Colony Optimization Algorithm (ACO) and Q-learning Algorithm. Then, the Genetic

Ant Colony Optimization Algorithm is used to select the optimal protection strategy set that does not exceed the budget from the protection strategy set to be selected.

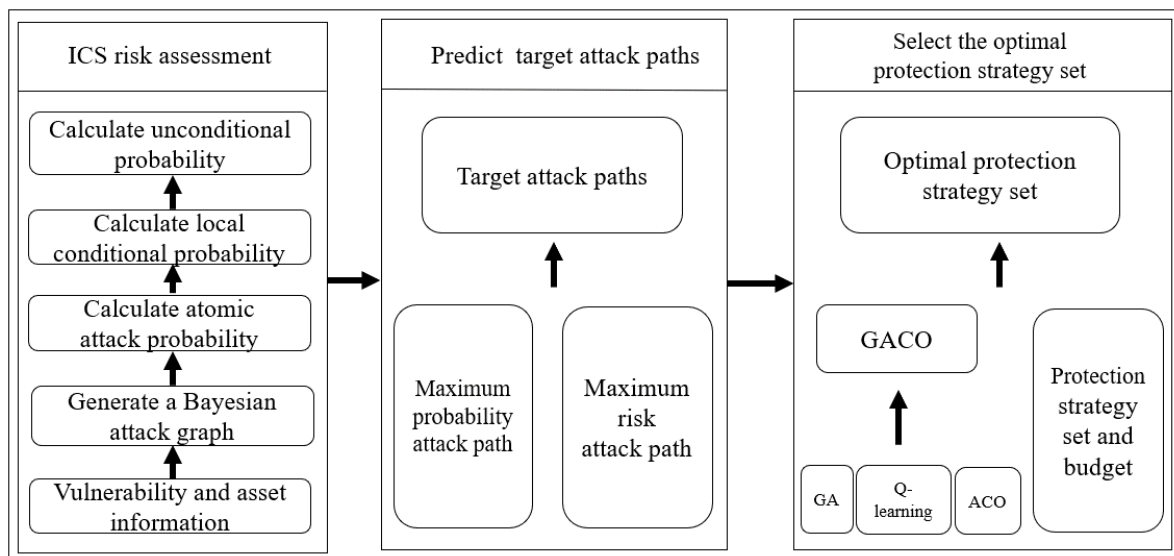


Figure 1. Protection strategy selection model.

4. ICS Risk Assessment

The attack graph is a directed graph that shows the sequence and effect of attacks that an attacker may launch [29,30]. Common attack graph types include state attack graph and attribute attack graph [31]. Among them, the state attack graph has the problem of state explosion [32], so this paper selects the attribute attack graph. In a Bayesian network, the occurrence probability of a node is only related to its parent node. Similarly, in the attack graph, whether the vulnerability of a node is exploited is only related to its parent node [33]. Therefore, this paper combines the attack graph with the Bayesian network to construct the Bayesian attack graph to evaluate the risk of ICS.

4.1. Definition of Bayesian Attack Graph

Definition 1. The Bayesian attack graph is a directed acyclic graph, defined as $BAG = (S, A, R, E, P)$, where:

(1) $S = S_{init} \cup S_{mid} \cup S_{aim}$ represents a set of attribute nodes. Among them, S_{init} is the initial attribute node set, S_{mid} is the intermediate attribute node set and S_{aim} is the target attribute node set. In addition, $S_i = \{0, 1\}$, 0 means the node is not occupied by the attacker, and 1 means the node is occupied by the attacker.

(2) $A = \{a_i | i = 1, 2, \dots, n\}$ represents the set of atomic attacks. Among them, $a_i = 1$ indicates that the attack has occurred, and $a_i = 0$ indicates that the attack has not occurred.

(3) R represents the relationship between parent and child nodes, which can be represented by a two-tuple $\langle S_j, d_j \rangle$. Among them, $d_j = \text{AND}$ means that the attack can be successful only if all parent nodes that reach S_j are true; $d_j = \text{OR}$ means that there is a parent node whose state is true, and the attack can be successful.

(4) $E = \{E_i | i = 1, 2, \dots, n\}$ represents the set of directed edges that describe the causal relationship of aggressive behavior. Among them, $(S_{pre}, S_{next}) \in E_i$ represents a directed edge from S_{pre} to S_{next} .

(5) P represents the unconditional probability of the node, which describes the probability of the node being attacked in the Bayesian attack graph.

4.2. Probability Calculation Based on Bayesian Attack Graph

4.2.1. Atomic Attack Probability

The Common Vulnerability Scoring System is a widely recognized vulnerability assessment standard [34,35]. This paper calculates the atomic attack probability based on the vulnerability exploitable metrics provided by the Common Vulnerability Scoring System. Among them, the exploitable metrics include Access Vector (AV), Access Complexity (AC) and Authentication (AU). The specific exploitable metrics are shown in Table 1.

Table 1. Exploitable metrics.

Index	Rank	Score
Access Vector (AV)	local access	0.395
	local network accessible	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

Definition 2. The atomic attack probability $P(v_i)$ is the probability that an attacker successfully completes an atomic attack by exploiting the vulnerability v_i . The calculation formula is as follows:

$$P(v_i) = 2 \times AV \times AC \times AU \quad (1)$$

4.2.2. Local Conditional Probability

Definition 3. The local conditional probability reflects the probability of an attribute node being attacked under the influence of its parent node. For the attribute node S_j in the Bayesian attack graph, if the node $S_j \in S_{mid} \cup S_{aim}$, then $S_i \in Pa[S_j]$, and the local conditional probability of S_j is expressed as $P(S_j | Pa[S_j])$. According to the difference of the parent–child node relationship, the calculation formula is as follows:

(1) if $d_j = \text{AND}$

$$P(S_j | Pa[S_j]) = \begin{cases} 0, & \exists S_i \in Pa[S_j], S_i = 0, \\ \prod_{S_i=1} P(v_i), & \text{otherwise.} \end{cases} \quad (2)$$

(2) if $d_j = \text{OR}$

$$P(S_j | Pa[S_j]) = \begin{cases} 0, & \forall S_i \in Pa[S_j], S_i = 0, \\ 1 - \prod_{S_i=1} [1 - P(v_i)], & \text{otherwise.} \end{cases} \quad (3)$$

4.2.3. Unconditional Probability

Definition 4. The unconditional probability is the joint probability of the current attribute node and all its ancestor nodes. If the node $S_j \in S_{mid} \cup S_{aim}$, the unconditional probability of this node is expressed as $P(S_j)$, and the calculation formula is as follows:

$$P(S_j) = \prod_{j=1}^n P(S_j | Pa[S_j]) \quad (4)$$

5. Predict Target Attack Paths

This paper expands on the maximum probability attack path [36,37], combines the attack probability and attack benefits, proposes the maximum risk attack path and analyzes the target attack path under different conditions from multiple angles.

5.1. Maximum Probability Attack Path

Definition 5. Multiply the unconditional probabilities $P(S_j)$ of all attribute nodes S_j on an attack path W , and the product is the attack reachability probability $P(W)$ of the path. The calculation formula is as follows.

$$P(W) = \prod_{j=1}^n P(S_j), \quad S_j \in W \quad (5)$$

The maximum probability attack path is the path with the highest attack probability and is used as the target attack path with the lowest attack difficulty.

5.2. Maximum Risk Attack Path

Definition 6. $V(S_j)$ represents the attack benefit of the atomic attack successfully capturing the attribute node S_j , then the risk value $R(S_j)$ of S_j is the product of the attack benefit of the node and the unconditional probability $P(S_j)$, and the calculation formula is as follows:

$$R(S_j) = V(S_j) \times P(S_j) \quad (6)$$

Therefore, the risk values $R(S_j)$ of attribute nodes S_j on an attack path W are accumulated, and the sum is the overall risk $R(W)$ of the path. The calculation formula is as follows:

$$R(W) = \sum_{j=1}^n R(S_j), \quad S_j \in W \quad (7)$$

The maximum risk attack path is the path with the highest overall risk and is used as the target attack path with the greatest attack damage.

6. Select the Optimal Protection Strategy Set

In this section, this paper first quantifies the protection metrics based on the National Vulnerability Database (NVD) and expert experience [38]. Then, aiming at the local optimal problem of the Genetic Algorithm and Ant Colony Optimization Algorithm, we propose to construct the Genetic Ant Colony Optimization Algorithm.

6.1. Quantitative Protection Metrics

Definition 7. $M = \{m_i | i = 1, 2, \dots, n\}$ represents a set of protection strategies. Among them, $m_i = 1$ means to select this protection strategy, and $m_i = 0$ means not to select this protection strategy. $C = \{c_i | i = 1, 2, \dots, n\}$ represents a set of protection cost. Among them, the protection cost corresponding to the protection strategy m_i is c_i . MC represents the total protection cost. If the product of m_i and c_i is not 0, it means that the protection strategy m_i is selected, and the value of the product is the protection cost of the protection strategy m_i . The total cost of protection is the sum of the costs of all selected strategies. The calculation formula is as follows:

$$MC = \sum_{i=1}^n m_i \times c_i \quad (8)$$

Definition 8. According to the NVD and CVSS, the attack benefits are divided into confidentiality benefits, integrity benefits and availability benefits. The above three kinds of benefit constitute the attack benefit v_i of the node, and the attack benefit and the protection benefit are numerically consistent. MV represents the total protection benefit. If the product of m_i and v_i is not 0, it means that the protection strategy m_i is selected, and the value of the product is the protection benefit of the

protection strategy m_i . The total protection benefit is the sum of the benefits of all selected strategies. The calculation formula is as follows:

$$MV = \sum_{i=1}^n m_i \times v_i \quad (9)$$

6.2. Reinforcement Learning

Reinforcement learning studies how agents interact with the environment in different states. As shown in Figure 2, a reinforcement learning model consists of five main parts: agent, environment, action, state and reward. Its flow can be described as a cycle of states, actions and rewards.

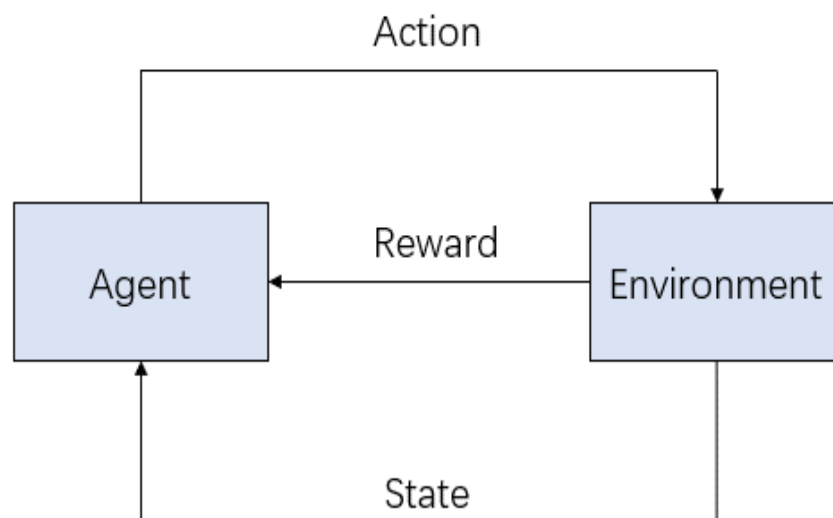


Figure 2. Illustration of the reinforcement learning process.

Q-learning is a model-free reinforcement learning algorithm. It has many advantages, such as it requires no prior knowledge and it learns as the task progresses. The process of Q-learning is: 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 environment gives rewards according to the degree of state change. Then, the Q table is updated according to the current state, next state, action and reward value. The updated formula is as follows:

$$Q_{t+1}(s_t, a_t) = Q(s_t, a_t) + \alpha [r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (10)$$

Among them, s_t is the state of the agent at time t , a_t is the action that the agent can perform at time t , r_t is the reward obtained by performing the action at time t , α is the learning rate, γ is the discount factor, $Q_t(s_t, a_t)$ is the reward value of the action a_t corresponding to the state s_t at time t , and $Q_{t+1}(s_t, a_t)$ is the cumulative reward the agent obtains at time t .

6.3. Genetic Ant Colony Optimization Algorithm

It is known that there are N alternative protection strategies, and the budget of the protection strategy set is B . The protection cost of the i th strategy is denoted c_i , and the protection benefit is denoted v_i . Each strategy can only be selected at most once. It is required to select at least one strategy to be added to the protection strategy set, so that the total protection benefit of the protection strategy set is maximized, and the total protection cost cannot exceed budget B . This problem is similar to the 0-1 knapsack

problem and belongs to a class of NP-hard problems. The Genetic Algorithm and Ant Colony Optimization Algorithm are often used to calculate NP-hard problems as traditional heuristic algorithms. However, both the Genetic Algorithm and Ant Colony Optimization Algorithm have local optimal problems, so this paper proposes to use the Genetic Ant Colony Optimization Algorithm to calculate the NP-hard problem.

The Genetic Algorithm is greatly affected by the initial population. When the initial population is within a reasonable range, the Genetic Algorithm can better play the global optimization ability, otherwise it is easy to fall into the local optimum. The positive feedback mechanism makes the Ant Colony Optimization Algorithm have a better convergence speed. However, the Ant Colony Optimization Algorithm is highly dependent on pheromones, which makes it easy to fall into local optimum. Therefore, according to the characteristics of the Genetic Algorithm and Ant Colony Optimization Algorithm, this paper constructs the Genetic Ant Colony Optimization Algorithm.

6.3.1. The Basic Idea of Genetic Ant Colony Optimization Algorithm

The Genetic Ant Colony Optimization Algorithm adds the Genetic Algorithm to the Ant Colony Optimization Algorithm and uses Q-learning to dynamically select the relevant parameters of the ant colony module and the genetic module. First, the Ant Colony Optimization Algorithm uses the strategy selection probability and the roulette method to select the initial strategy set that does not exceed the limit cost, and the selection result is used as the initial population of Genetic Algorithm. Then, the Genetic Algorithm selects the strategy set that does not exceed the limited cost through the selection operation, crossover operation and mutation operation and uses the selection result to update the pheromone of the Ant Colony Optimization Algorithm. In this way, the parameter intermodulation of the two algorithms is realized, and the Genetic Ant Colony Optimization Algorithm has excellent global optimization ability and fast convergence speed.

6.3.2. Using Q-Learning to Update the Parameters of the Algorithm

At present, many researchers have given the recommended parameter settings of the Genetic Algorithm and Ant Colony Optimization Algorithm. However, the interplay between parameter settings and algorithm performance remains complex. Q-learning can adjust the action according to the learning to obtain the best value. Therefore, we use Q-learning to dynamically select relevant parameters with greater influence.

The parameters that the ant colony module needs to dynamically select are: relative importance of pheromone concentration μ and relative importance of visibility β . The parameters that the genetic module needs to dynamically select are: crossover probability P_c and mutation probability P_m . In both modules, each individual in the population acts as an agent. Each individual has a Q table, and the initial value of the Q table is 0. In the Q table, we set two states for each individual, and each state corresponds to six actions. Table 2 shows the form of the Q table. Each action corresponds to a set of parameter settings. Table 3 shows a set of parameter settings corresponding to each action in the ant colony module. Table 4 shows a set of parameter settings corresponding to each action in the genetic module. In addition, if the individual takes an action and the effect is better than before, we set the reward value r_t to 1 and change the state of the individual to state 2. Otherwise, we set the reward value r_t to 0 and change the state of the individual to state 1. Finally, we update the Q table of the individual.

Table 2. The form of the Q table.

Q Table	action1	action2	action3	action4	action5	action6
state1	$Q(s_1, a_1)$	$Q(s_1, a_2)$	$Q(s_1, a_3)$	$Q(s_1, a_4)$	$Q(s_1, a_5)$	$Q(s_1, a_6)$
state2	$Q(s_2, a_1)$	$Q(s_2, a_2)$	$Q(s_2, a_3)$	$Q(s_2, a_4)$	$Q(s_2, a_5)$	$Q(s_2, a_6)$

Table 3. The corresponding form of actions and parameters in the ant colony module.

action1	action2	action3	action4	action5	action6
$(\mu 1, \beta 1)$	$(\mu 2, \beta 2)$	$(\mu 3, \beta 3)$	$(\mu 4, \beta 4)$	$(\mu 5, \beta 5)$	$(\mu 6, \beta 6)$

Table 4. The corresponding form of actions and parameters in the genetic module.

action1	action2	action3	action4	action5	action6
$(P_c 1, P_m 1)$	$(P_c 2, P_m 2)$	$(P_c 3, P_m 3)$	$(P_c 4, P_m 4)$	$(P_c 5, P_m 5)$	$(P_c 6, P_m 6)$

6.3.3. Steps of Genetic Ant Colony Optimization Algorithm

The steps of the Genetic Ant Colony Optimization Algorithm proposed in this paper are as follows:

- (1) Initialize the population, Q table, state and action of the ant colony module.
- (2) Each ant selects an action from the Q table to obtain a set of recommended parameter settings according to the current state.
- (3) Calculate the strategy selection probability. Each ant calculates the probability of choosing a certain strategy according to Formula (11). In Formula (11), P_i^k represents the probability that the kth ant chooses strategy m_i ; $\tau_i(t)$ represents the pheromone concentration of strategy m_i at time t ; η_i represents the visibility of strategy m_i , and the value of η_i is the unit cost value of strategy m_i ; μ represents the relative importance of pheromone concentration; β represents the relative importance of visibility; and $tabu(k)$ represents the strategy set selected by the kth ant.

$$P_i^k = \begin{cases} \frac{\tau_i^\mu(t) \eta_i^\beta(i)}{\sum \tau_i^\mu(t) \eta_i^\beta(i)}, & m_i \notin tabu(k), \\ 0, & otherwise. \end{cases} \quad (11)$$

- (4) On the basis of step (3), a roulette method is used to randomly select a strategy. If the selected strategy is added to the protection strategy set so that the total protection cost is higher than the limit cost, the strategy will be removed, otherwise the strategy will be retained.

- (5) After each ant completes the strategy selection, compare the benefit of the selected protection strategy set with the previous ones and update the ant's state and Q table according to the comparison result.

- (6) When all the ants complete the strategy selection, the protection strategy set selected by each ant is used as the initial population of the Genetic Algorithm. Additionally, initialize the Q table, state and action of the genetic module.

- (7) The individual selects an action from the Q table to obtain a set of recommended parameter settings according to the current state.

- (8) Calculate the fitness value. Calculate the sum of the benefits of the strategy set corresponding to each individual in the population and use the calculation result as the fitness value of each individual.

- (9) Selection operation. The roulette method is used to select the better individual from the population according to the size of the fitness value.

- (10) Crossover operation. When the crossover probability P_c is satisfied, the crossover point is randomly selected, and the selected individual and the current individual are crossed in pairs to generate offspring individual with the crossover point as the boundary, thereby improving the diversity of the population.

- (11) Mutation operation. When the mutation probability P_m is satisfied, the mutation point is randomly selected on the basis of the crossover operation, and the mutation operation is performed on the offspring individuals by means of single-point mutation, which further improves the global optimization ability of the algorithm.

(12) The fitness value of the offspring individual is compared with the fitness value of the parent individual, and the individual's state and Q table are updated according to the comparison result.

(13) Screen populations. Calculate the protection cost of the strategy set corresponding to each individual in the offspring population, and only retain offspring individuals whose protection cost is not higher than the limit cost.

(14) Merge populations. Merge the parent population and the child population, sort according to the size of the individual fitness values, and select the number of individuals consistent with the size of the parent population as the parent population for the next traversal according to the sorting result.

(15) If the iteration number of the Genetic Algorithm is satisfied, go to step (16), otherwise go to step (7).

(16) The pheromone of the Ant Colony Optimization Algorithm is updated according to the protection strategy set obtained by the Genetic Algorithm, and the update formulas are (12) and (13). In Formulas (12) and (13), $\tau_i(t+n)$ represents the pheromone concentration of strategy m_i at time $(t+n)$, ρ represents the volatility coefficient of pheromone, $\Delta\tau_i(t)$ represents the total amount of pheromone left by n ants on strategy m_i , Q is a constant, V_k represents the total benefit of the protection strategy set corresponding to the k th ant and C_k represents the total cost of the protection strategy set corresponding to the k th ant.

$$\tau_i(t+n) = (1-\rho)\tau_i(t) + \Delta\tau_i(t) \quad (12)$$

$$\Delta\tau_i(t) = \sum_{k=1}^n Q \cdot V_k / C_k \quad (13)$$

(17) If the number of iterations of the Ant Colony Optimization Algorithm is satisfied, output the optimal protection strategy set, otherwise go to step (2).

The pseudocode of the Algorithm 1 is as follows:

Algorithm 1 GACO.

Input: protection strategy set, protection budget, population size, maximum number of iterations(*maxIterations*)

Output: optimal protection strategy set

```

1: initialize the population, Q table, state and action of the ant colony module
2: while iteraco  $\leq$  maxIterations do
3:   select the action from Q table
4:   calculate the strategy selection probability
5:   each ant chooses a set of protection strategies
6:   update the ant's state and Q table
7:   initialize the population, Q table, state and action of the genetic module
8:   while iterga  $\leq$  maxIterations do
9:     select the action from Q table
10:    calculate individual fitness value
11:    select operation
12:    crossover operation
13:    mutation operation
14:    update the individual's status and Q table
15:    screen populations
16:    merge populations
17:    iterga ++
18:   end while
19:   update pheromone concentration
20:   iteraco ++
21: end while

```

7. Experiment and Discussion

7.1. Experimental Scene

The simulation environment of this paper is an industrial water distribution system, as shown in Figure 3. The industrial water distribution system consists of three parts: water pump, water pipeline and water tanker. Additionally, the components in the system are controlled by a programmable logic controller (PLC) [39].

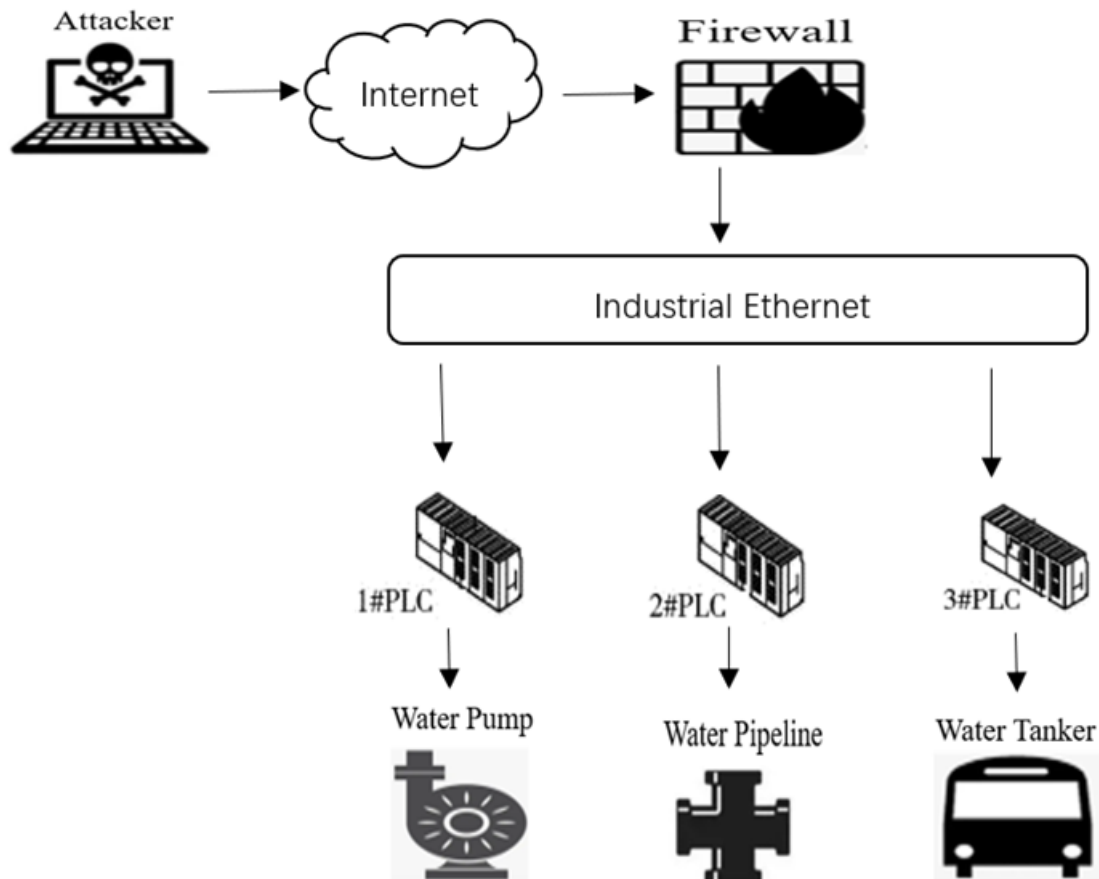


Figure 3. Experimental scene.

7.2. Generate a Bayesian Attack Graph

This paper uses the Nessus [40] tool to scan for vulnerabilities in the system. The vulnerability information of each host is shown in Table 5. Then, the MulVal [41] tool generates an attack graph based on the obtained vulnerability information, network configuration and host configuration information. The attack graph is shown in Figure 4.

Table 5. Vulnerability information.

Host	Vulnerability ID
198.168.0.1	CVE-1999-0517
198.168.0.2	CVE-1999-0517
198.168.0.10	CVE-1999-0517

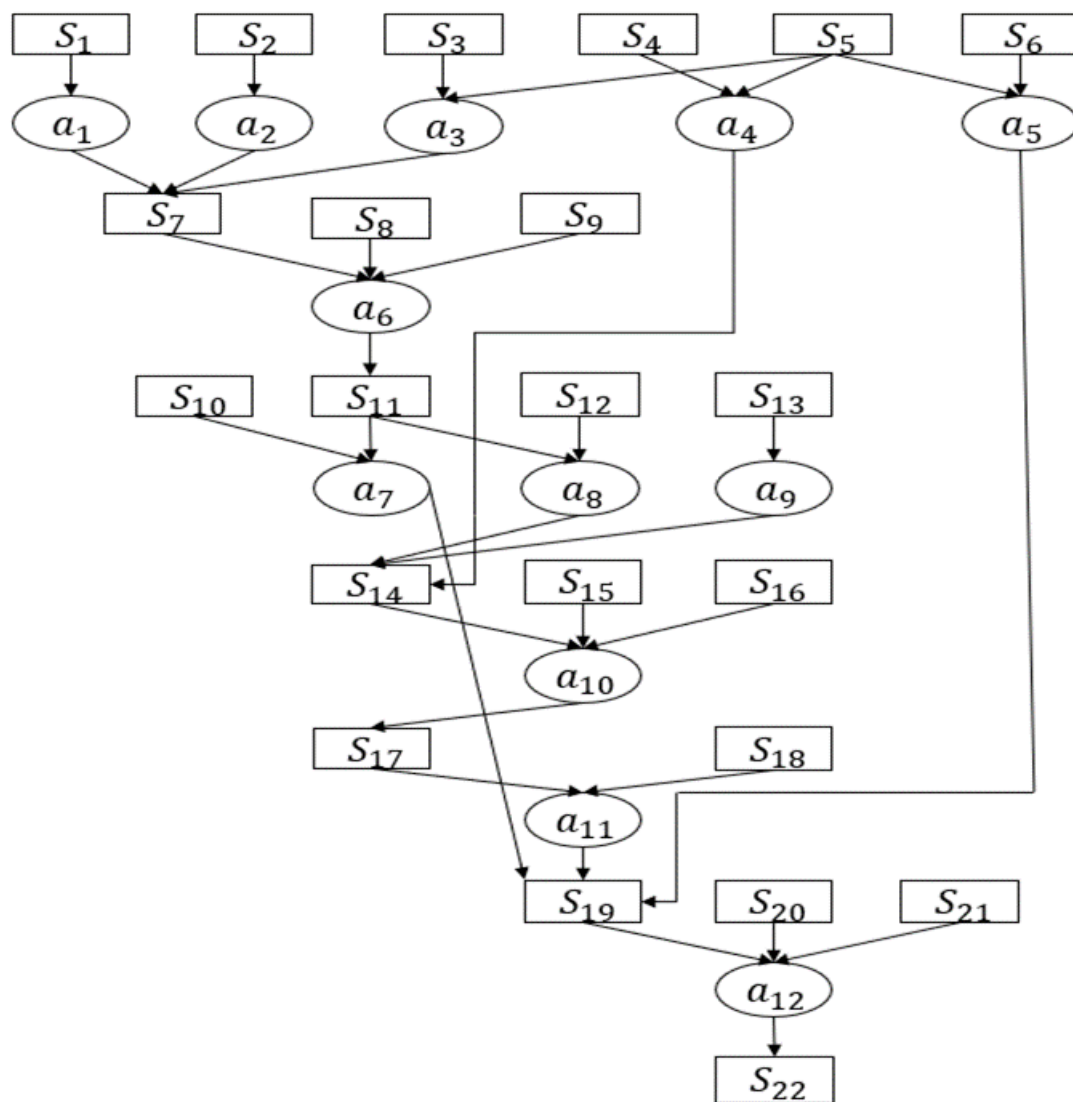


Figure 4. Bayesian attack graph.

7.3. Unconditional Probability of a Node

Table 6 gives the unconditional probability of each attribute node.

Table 6. Unconditional probabilities for each attribute node.

Node	Probability	Node	Probability	Node	Probability
S_1	1.0000	S_2	1.0000	S_3	1.0000
S_4	1.0000	S_5	1.0000	S_6	1.0000
S_7	0.5000	S_8	1.0000	S_9	0.9960
S_{10}	1.0000	S_{11}	1.0000	S_{12}	1.0000
S_{13}	1.0000	S_{14}	0.5000	S_{15}	1.0000
S_{16}	0.9863	S_{17}	0.8372	S_{18}	1.0000
S_{19}	0.5000	S_{20}	1.0000	S_{21}	0.9530
S_{22}	0.5000				

7.4. Quantify Protection Costs and Attack Benefits

Table 7 shows the protection cost and protection benefit of the protection strategy m_i corresponding to the atomic attack a_i .

Table 7. Protection costs and protection benefits.

Attack	Strategy	Protective Action	Cost	Value
a_1	m_1	Disable multi-hop access 192.168.0.1-192.168.0.10	33	50
a_2	m_2	Disable multi-hop access 192.168.0.2-192.168.0.10	33	50
a_3	m_3	Disable direct network access 192.168.0.10	30	52
a_4	m_4	Disable direct network access 192.168.0.2	30	52
a_5	m_5	Disable direct network access 192.168.0.1	30	52
a_6	m_6	Limit remote visit 192.168.0.10	20	5
a_6	m_7	Patch CVE-1999-0517 192.168.0.10	15	25
a_7	m_8	Disable multi-hop access 192.168.0.10	14	26
a_8	m_9	Disable multi-hop access 192.168.0.2-192.168.0.10	33	50
a_9	m_{10}	Disable multi-hop access 192.168.0.1-192.168.0.2	30	56
a_{10}	m_{11}	Limit remote visit 192.168.0.2	33	15
a_{10}	m_{12}	Patch CVE-1999-0517 192.168.0.2	17	30
a_{11}	m_{13}	Disable multi-hop access 192.168.0.1-192.168.0.2	30	56
a_{12}	m_{14}	Limit remote visit 192.168.0.1	18	8
a_{12}	m_{15}	Patch CVE-1999-0517 192.168.0.1	19	40

7.5. Predict Attack Paths

According to Definitions 5 and 6, the path reachability probability and path risk value in the attack graph can be calculated, as shown in Table 8. It can be seen from Table 8 that path 8 is the target attack path with the lowest attack difficulty, and path 6 is the target attack path with the greatest attack damage. Under the limited cost, it is unrealistic to deploy the protection strategy for all nodes. Therefore, we prioritize the deployment of protection strategies for the maximum probability attack path and the maximum risk attack path.

Table 8. Path reachability probability and path risk value.

Index	Attack Path	Probability	Risk
1	$S_1-S_7-S_8-S_9-S_{10}-S_{11}-S_{19}-S_{20}-S_{21}-S_{22}$	0.119	92.000
2	$S_1-S_7-S_8-S_9-S_{11}-S_{12}-S_{14}-S_{15}-S_{16}-S_{17}-S_{18}-S_{19}-S_{20}-S_{21}-S_{22}$	0.049	169.674
3	$S_2-S_7-S_8-S_9-S_{10}-S_{11}-S_{19}-S_{20}-S_{21}-S_{22}$	0.119	92.000
4	$S_2-S_7-S_8-S_9-S_{11}-S_{12}-S_{14}-S_{15}-S_{16}-S_{17}-S_{18}-S_{19}-S_{20}-S_{21}-S_{22}$	0.049	144.671
5	$S_3-S_5-S_7-S_8-S_9-S_{10}-S_{11}-S_{19}-S_{20}-S_{21}-S_{22}$	0.119	93.000
6	$S_3-S_5-S_7-S_8-S_9-S_{11}-S_{12}-S_{14}-S_{15}-S_{16}-S_{17}-S_{18}-S_{19}-S_{20}-S_{21}-S_{22}$	0.049	170.674
7	$S_4-S_5-S_{14}-S_{15}-S_{16}-S_{17}-S_{18}-S_{19}-S_{20}-S_{21}-S_{22}$	0.098	115.674
8	$S_5-S_6-S_{19}-S_{20}-S_{21}-S_{22}$	0.238	50.000
9	$S_{13}-S_{14}-S_{15}-S_{16}-S_{17}-S_{18}-S_{19}-S_{20}-S_{21}-S_{22}$	0.098	117.674

7.6. Experimental Results

Set the protection cost of each attack path to no more than 100. The optional protection strategy for path 8 is the strategy in the set $M1 = \{m_5, m_{14}, m_{15}\}$. The total protection cost of strategy set M1 is 67, so the optimal protection strategy set of path 8 is $M1 = \{m_5, m_{14}, m_{15}\}$. The optional protection strategy for path 6 is the strategy in the set $M2 = \{m_3, m_6, m_7, m_9, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}\}$. The total protection cost of strategy set M2 is 215. Therefore, we need to choose the most beneficial set of protection strategies that do not exceed the budget for path 6. In order to solve this problem, and test the performance of the algorithm proposed in this paper, this paper compares the Genetic Algorithm (GA), Ant Colony Optimization Algorithm (ACO), Reinforcement learning-based Particle Swarm Optimization Algorithm (RLPSO) [12] and Reinforcement learning-based Differential Evolution Algorithm (RLDE) [13].

Specifically, we run each algorithm 100 times with population sizes of 15 and 100, respectively. Additionally, the number of iterations per run is 20. The relevant parameter settings of each algorithm are shown in Table 9. Among them, when the population size is 15, the effect diagrams of each algorithm are shown in Figures 5–9. The effect diagrams of each algorithm when the population size is 100 are shown in Figures 10–14. In both cases, the number of occurrences of local optimal solutions of each algorithm is shown in Table 10.

Table 9. Parameter settings.

Algorithm	Parameter	Value
GACO	α	0.1
	γ	0.9
	ρ	0.1
	Q	1
	$\tau_i(t=0)$	1
	μ, β	[(1,2),(1,3),(1,4),(2,3),(2,4),(2,5)]
	P_c, P_m	[(0.8,0.2),(0.7,0.2),(0.6,0.2), (0.8,0.1),(0.7,0.1),(0.6,0.1)]
GA	P_c	0.8
	P_m	0.1
ACO	μ	1
	β	3
	ρ	0.1
	Q	1
	$\tau_i(t=0)$	1
RLPSO	γ	0.9
	α	0.1
	w, c1, c2	[(0.8,2,1.5),(0.8,1.5,2),(0.8,2,2), (0.6,2,1.5),(0.6,1.5,2),(0.6,2,2)]
RLDE	γ	0.9
	α	0.1
	F, Cr	[(0.6,0.7),(0.6,0.8),(0.7,0.7), (0.8,0.7),(0.8,0.9),(0.9,0.9)]

Table 10. The number of occurrences of local optimal solutions of each algorithm.

Population Size	Algorithm	The Number of Occurrences
15	GACO	9
	GA	37
	ACO	43
	RLPSO	70
	RLDE	65
100	GACO	0
	GA	15
	ACO	19
	RLPSO	8
	RLDE	6

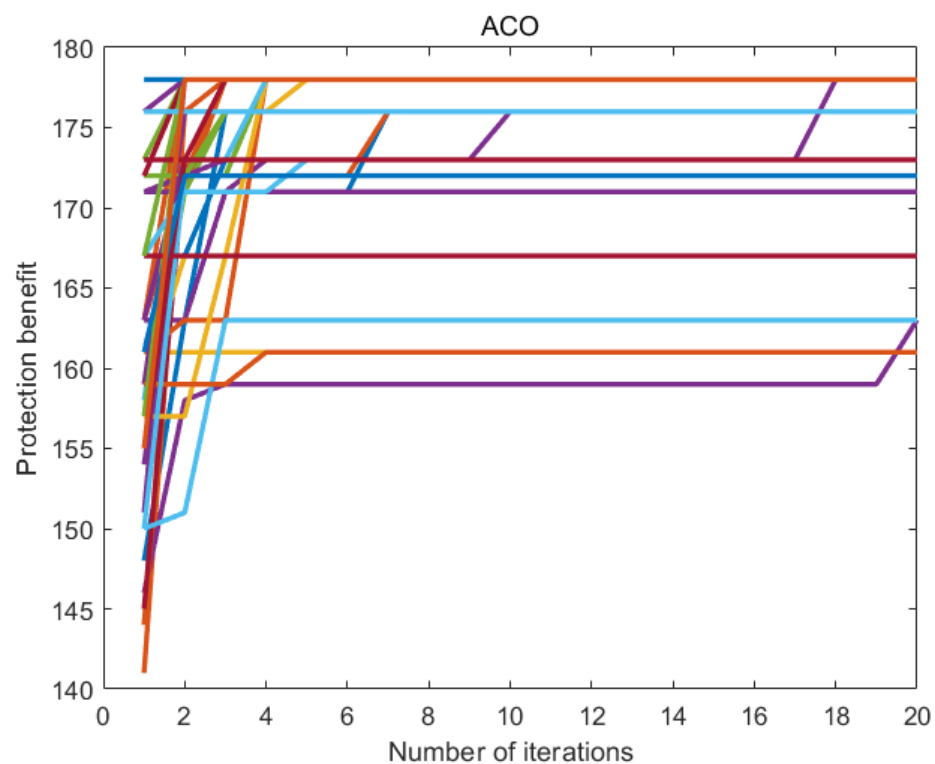


Figure 5. Population size is 15, and the ACO algorithm is run 100 times.

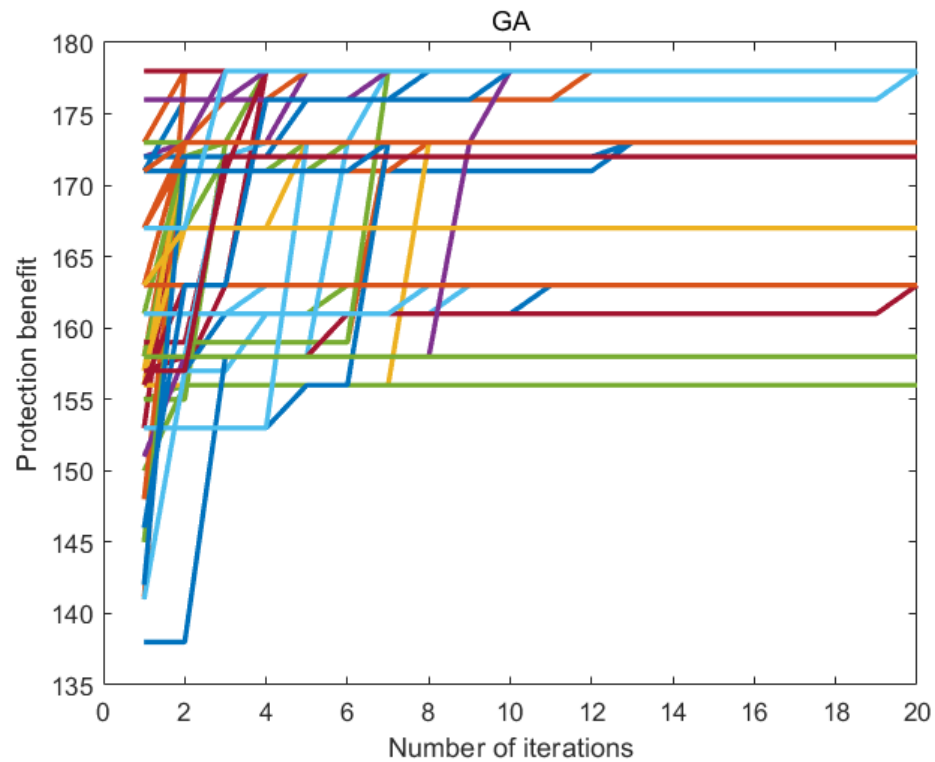


Figure 6. Population size is 15, and the GA algorithm is run 100 times.

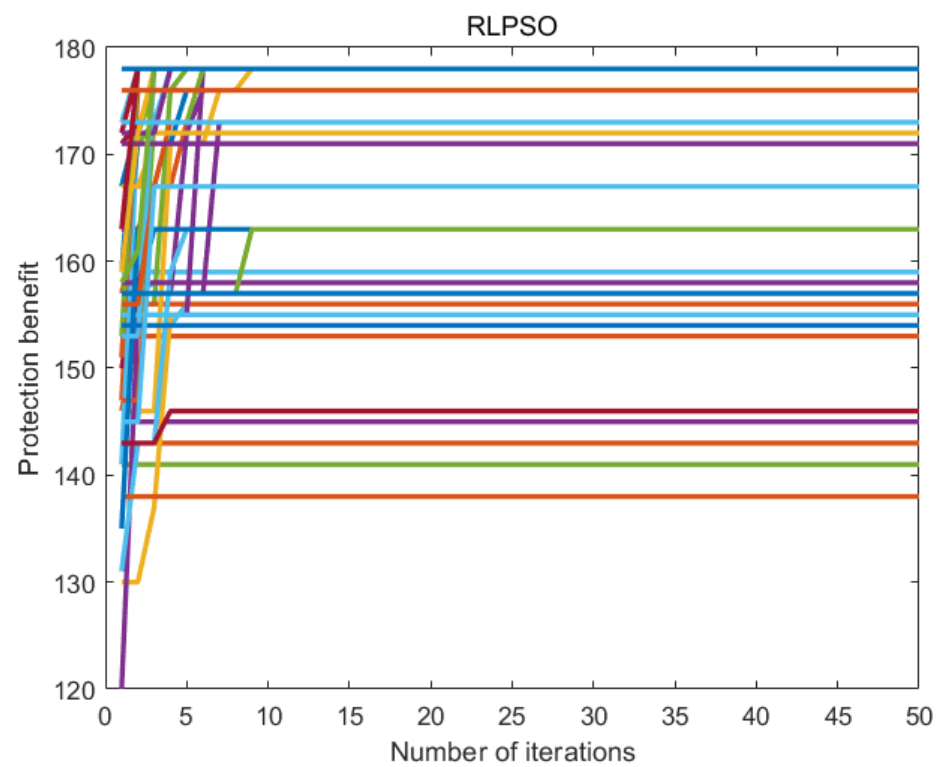


Figure 7. Population size is 15, and the RLPSO algorithm is run 100 times.

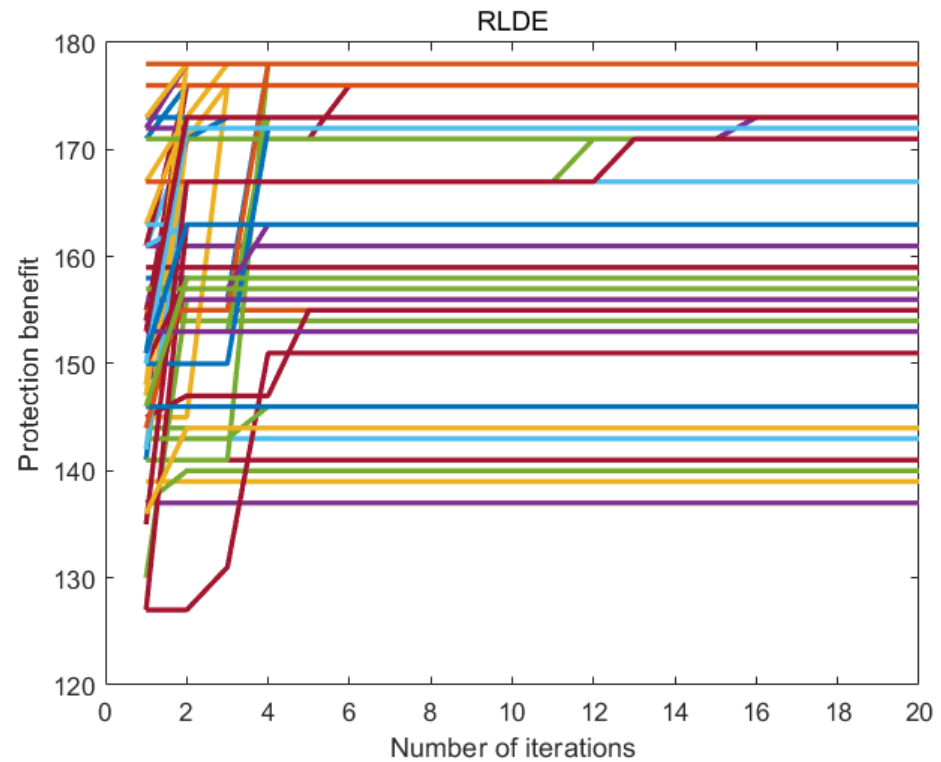


Figure 8. Population size is 15, and the RLDE algorithm is run 100 times.

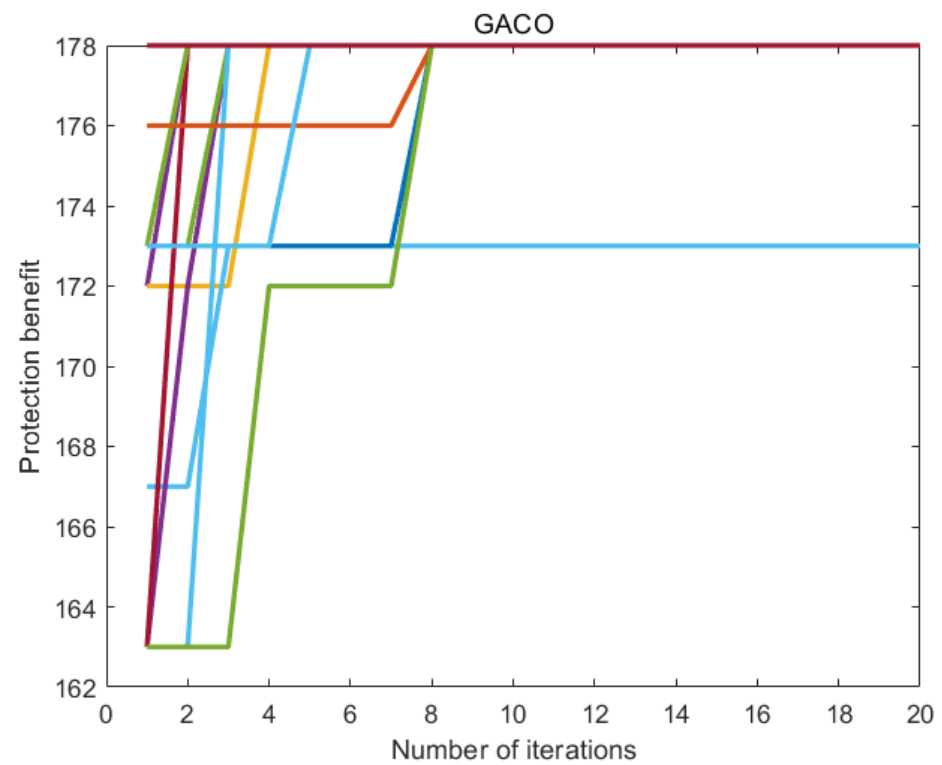


Figure 9. Population size is 15, and the GACO algorithm is run 100 times.

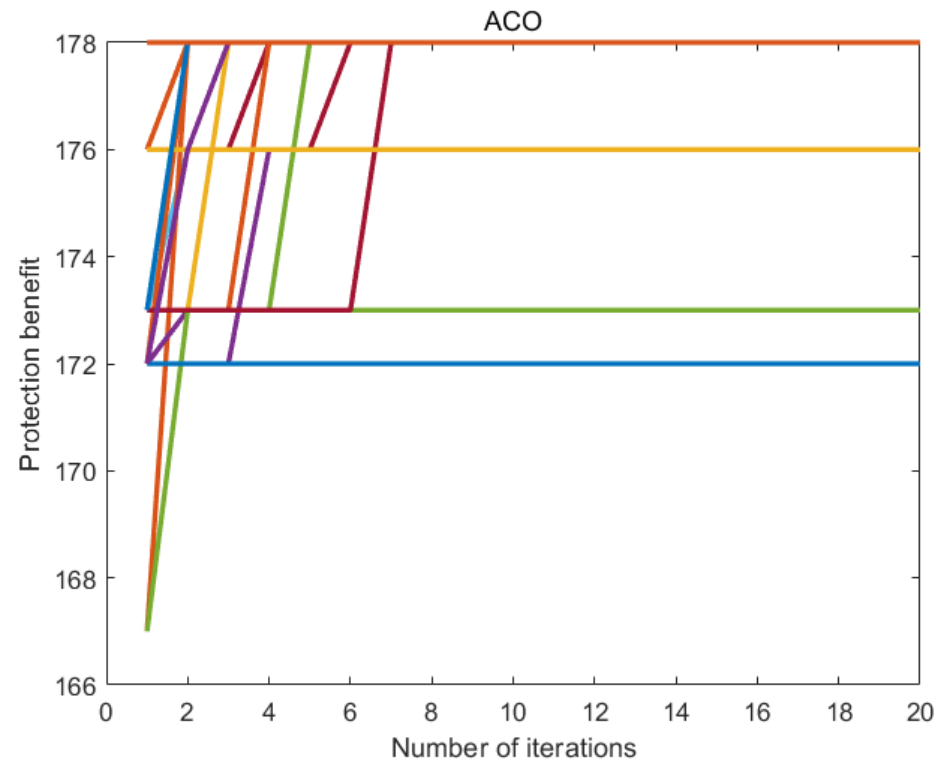


Figure 10. Population size is 100, and the ACO algorithm is run 100 times.

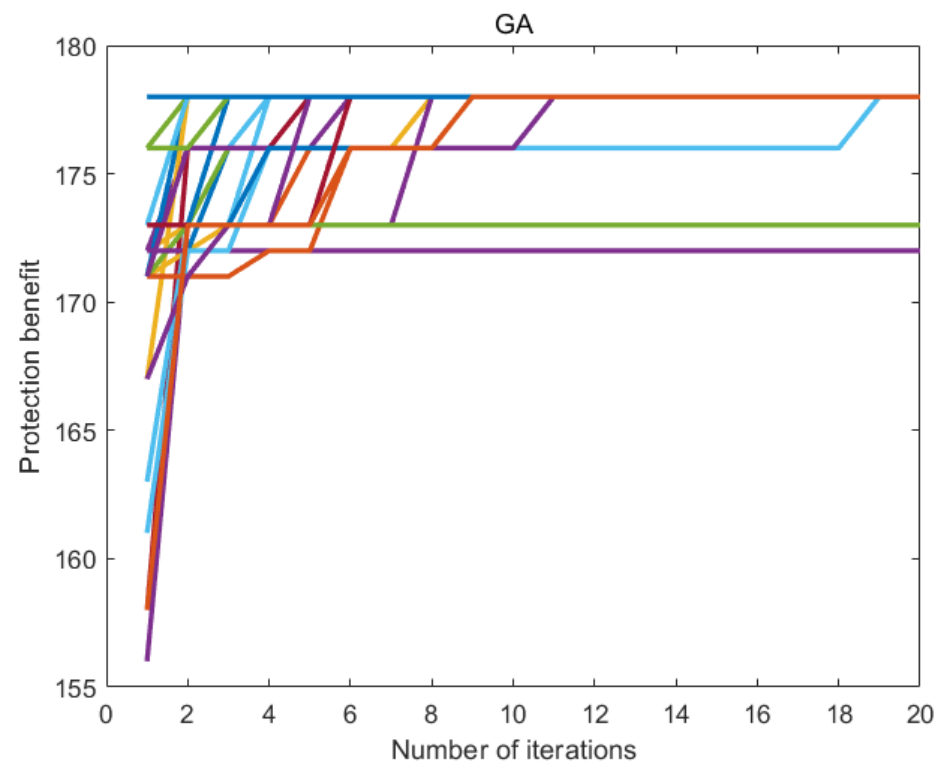


Figure 11. Population size is 100, and the GA algorithm is run 100 times.

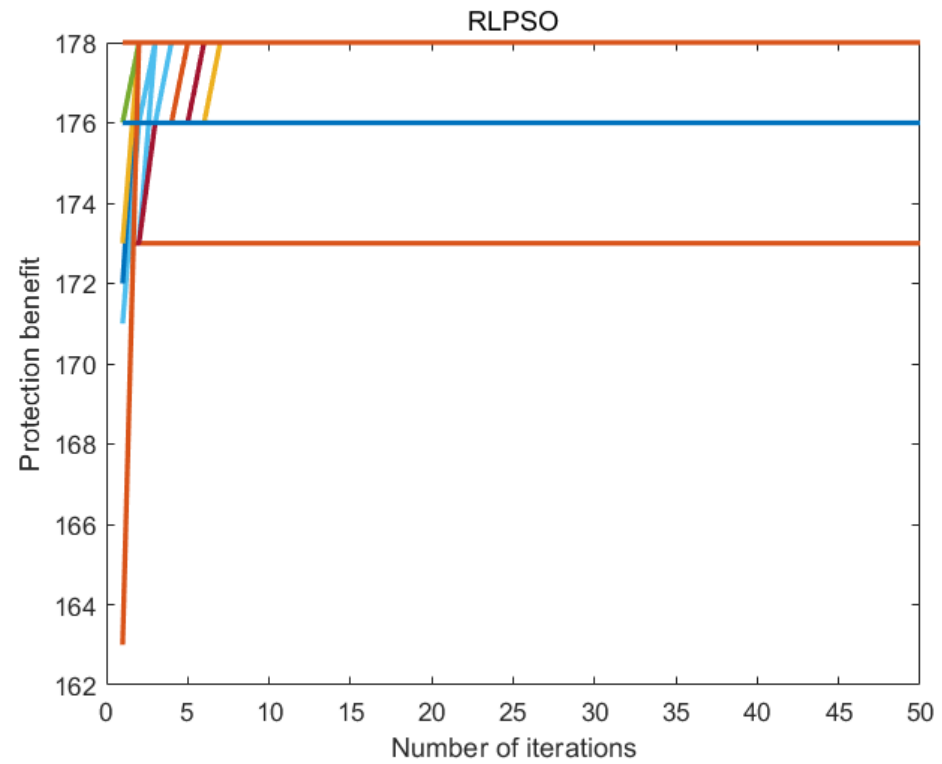


Figure 12. Population size is 100, and the RLPSO algorithm is run 100 times.

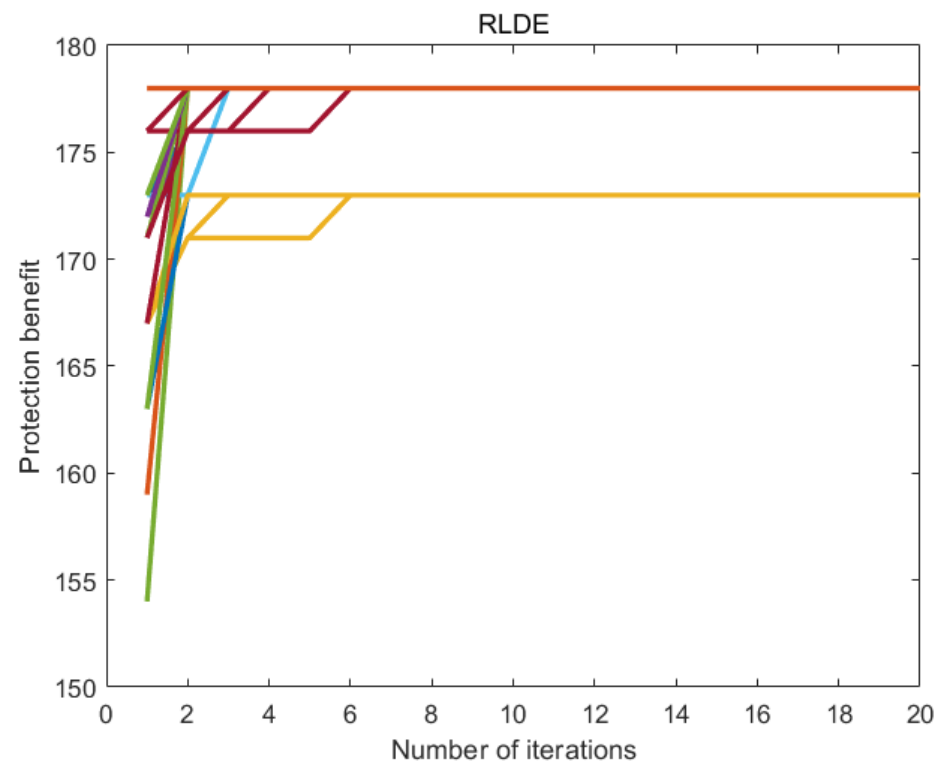


Figure 13. Population size is 100, and the RLDE algorithm is run 100 times.

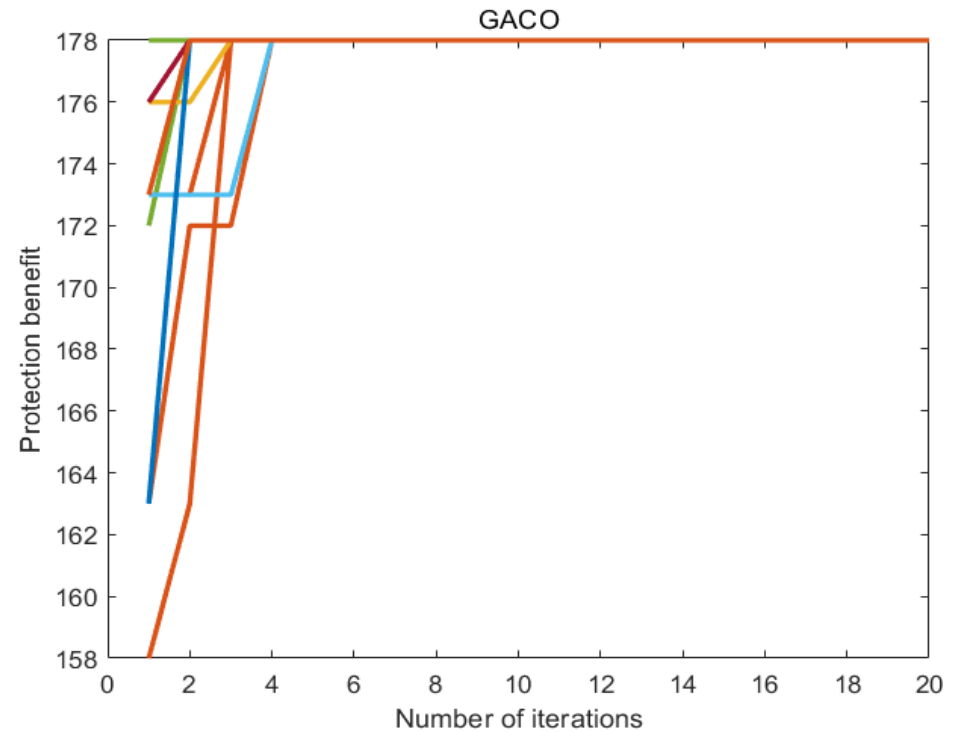


Figure 14. Population size is 100, and the GACO algorithm is run 100 times.

7.7. Discussion

First of all, the experimental results show that the maximum protection benefit of path 6 under the limited budget is 178, the corresponding protection strategy set is $M = \{m_3, m_{12}, m_{13}, m_{15}\}$ and the protection cost of the protection strategy set is 96. Secondly, through these two sets of comparative experiments, we found that the convergence accuracy of the Genetic Ant Colony Optimization Algorithm is better than other comparative algorithms in both small-scale populations and large-scale populations. Specifically, although the Genetic Ant Colony Optimization Algorithm also has a local optimal problem under the condition of a population size of 15, the Genetic Ant Colony Optimization Algorithm has much fewer local optimal solutions than other comparative algorithms. Additionally, under the condition that the population size is 100, the Genetic Ant Colony Optimization Algorithm has no local optimum problem. This good result is because the genetic module and the ant colony module in the Genetic Ant Colony Optimization Algorithm achieve mutual adjustment, and Q-learning dynamically selects the appropriate parameter settings for the algorithm. Finally, due to the addition of Q-learning to the Genetic Ant Colony Optimization Algorithm, the algorithm needs to use more memory space to store the Q table. This is the aspect that the algorithm proposed in this paper needs to be further optimized.

8. Conclusions

Aiming at the problem of protection strategy selection under limited budget, this paper proposes a protection strategy selection model based on the Genetic Ant Colony Optimization Algorithm. First, the risk profile of the system is assessed according to the Common Vulnerability Scoring System and the Bayesian attack graph. Secondly, the target attack path is predicted from the two aspects of the maximum probability attack path and the maximum risk attack path. Then, based on the Genetic Algorithm, Ant Colony Optimization Algorithm and Q-learning Algorithm, this paper proposes the Genetic Ant Colony Optimization Algorithm. Finally, by testing the performance of the algorithm many times, we found that the convergence accuracy of the Genetic Ant Colony Optimization Algorithm is better than other comparison algorithms in both small-scale population and large-scale population conditions. At present, the Genetic Ant Colony Optimization Algorithm needs to occupy more memory space to store the Q table. In the future, we plan to use the Deep Q Network to replace Q-learning in the Genetic Ant Colony Optimization Algorithm to further improve the performance of the algorithm.

Author Contributions: Conceptualization, Y.Z.; formal analysis, Y.Z.; methodology, X.L. (Xinzhan Li), L.X. and D.Z.; resources, X.L. (Xin Li); writing—original draft, X.L. (Xinzhan Li); writing—review and editing, L.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Project of China (2020AAA0107700), the National Natural Science Foundation of China (62172244), the Shandong Provincial Natural Science Foundation (ZR2020YQ06, ZR2021MF132) and the Young innovation team of colleges and universities in Shandong province (2021KJ001).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Appendix A

Table A1 shows all the parameters in this paper and their descriptions.

Table A1. Parameters and their descriptions.

Parameters	Descriptions
S	the set of attribute nodes
A	the set of atomic attacks
R	the relationship between parent and child nodes
E	the set of directed edges
$P(v_i)$	the atomic attack probability against vulnerability v_i
$P(S_j Pa[S_j])$	the local conditional probability of attribute node S_j
$P(S_j)$	the unconditional probability of attribute node S_j
$P(W)$	the attack reachability probability of the path W
$R(S_j)$	the risk value of attribute node S_j
$R(W)$	the overall risk of the path W
M	a set of protection strategies
C	a set of protection cost
MC	the total protection cost
MV	the total protection benefit
s_t	the state of the agent at time t
a_t	the action that the agent can perform at time t
r_t	the reward obtained by performing the action at time t
α	the learning rate
γ	the discount factor
$Q_t(s_t, a_t)$	the reward value of the action a_t corresponding to the state s_t at time t
$Q_{t+1}(s_t, a_t)$	the cumulative reward the agent obtains at time t
P_i^k	the probability that the k th ant chooses strategy m_i
$\tau_i(t)$	the pheromone concentration of strategy m_i at time t
η_i	the visibility of strategy m_i
μ	the relative importance of pheromone concentration
β	the relative importance of visibility
$tabu(k)$	the strategy set selected by the k th ant
ρ	the volatility coefficient of pheromone
Q	a constant
V_k	the total benefit of the protection strategy set corresponding to the k th ant
C_k	the total cost of the protection strategy set corresponding to the k th ant

References

1. 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.
2. 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. [\[CrossRef\]](#)
3. Knowles, W.; Prince, D.; Hutchison, D.; Disso, J.F.P.; Jones, K. A survey of cyber security management in industrial control systems. *Int. J. Crit. Infrastruct. Prot.* **2015**, *9*, 52–80. [\[CrossRef\]](#)
4. Zhao, D.; Wang, L.; Wang, Z.; Xiao, G. Virus propagation and patch distribution in multiplex networks: modeling, analysis, and optimal allocation. *IEEE Trans. Inf. Forens. Secur.* **2019**, *14*, 1755–1767. [\[CrossRef\]](#)
5. 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.* **2020**, *51*, 7823–7837. [\[CrossRef\]](#)
6. Hemsley, K.E.; Fisher, E. *History of Industrial Control System Cyber Incidents*; Technical Report; Idaho National Lab.(INL): Idaho Falls, ID, USA, 2018.
7. 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. [\[CrossRef\]](#)
8. 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. [\[CrossRef\]](#)

9. Whitley, D. A genetic algorithm tutorial. *Stat. Comput.* **1994**, *4*, 65–85. [\[CrossRef\]](#)
10. Wang, S.; Zhang, Z.; Kadobayashi, Y. Exploring attack graph for cost-benefit security hardening: A probabilistic approach. *Comput. Secur.* **2013**, *32*, 158–169. [\[CrossRef\]](#)
11. Dorigo, M.; Maniezzo, V.; Coloni, A. Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **1996**, *26*, 29–41. [\[CrossRef\]](#)
12. 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.
13. Huynh, T.N.; Do, D.T.; Lee, J. Q-Learning-based parameter control in differential evolution for structural optimization. *Appl. Soft Comput.* **2021**, *107*, 107464. [\[CrossRef\]](#)
14. Zuoguang, W.; Wei, Q.; Liu, W.W. Quantitative risk assessment of industrial control systems based on attack-tree and CVSS. *Appl. Res. Comput.* **2016**, *12*, 3785–3790.
15. Meng, Z.; Yang, C. Two-stage differential evolution with novel parameter control. *Inf. Sci.* **2022**, *596*, 321–342. [\[CrossRef\]](#)
16. Zamani, H.; Nadimi-Shahraki, M.H.; Gandomi, A.H. QANA: Quantum-based avian navigation optimizer algorithm. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104314. [\[CrossRef\]](#)
17. Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S. An improved grey wolf optimizer for solving engineering problems. *Expert Syst. Appl.* **2021**, *166*, 113917. [\[CrossRef\]](#)
18. Zamani, H.; Nadimi-Shahraki, M.H.; Gandomi, A.H. Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization. *Comput. Methods Appl. Mech. Eng.* **2022**, *392*, 114616. [\[CrossRef\]](#)
19. Chakraborty, S.; Saha, A.K.; Chakraborty, R.; Saha, M. An enhanced whale optimization algorithm for large scale optimization problems. *Knowl.-Based Syst.* **2021**, *233*, 107543. [\[CrossRef\]](#)
20. Wu, H.; Gu, Y.; Cheng, G.; Zhou, Y. Effectiveness Evaluation Method for Cyber Deception Based on Dynamic Bayesian Attack Graph. In Proceedings of the 2020 3rd International Conference on Computer Science and Software Engineering, Beijing, China, 22–24 May 2020; pp. 1–9.
21. Zhang, Y.; Wang, B.; Wu, C.; Wei, X.; Wang, Z.; Yin, G. Attack Graph-based Quantitative Assessment for Industrial Control System Security. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 1748–1753.
22. Xu, K.; Wang, X.; Xu, H.; Dong, N.; Han, M.; Zhou, X. A vulnerability scanning scheme based on attack graph for smart grid industrial control system. In *Proceedings of the IOP Conference Series: Earth and Environmental Science*; IOP Publishing: Bristol, UK, 2021; Volume 645, p. 012060.
23. Yang, J.; Yang, Y. Optimal Security Protection Selection Strategy Based on Markov Model Attack Graph. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2021; Volume 2132, p. 012020.
24. Boudermine, A.; Khatoun, R.; Choyer, J.H. Attack Graph-based Solution for Vulnerabilities Impact Assessment in Dynamic Environment. In Proceedings of the 2022 5th Conference on Cloud and Internet of Things (CIoT), Marrakech, Morocco, 28–30 March 2022; pp. 24–31.
25. Wang, L.; Islam, T.; Long, T.; Singhal, A.; Jajodia, S. An attack graph-based probabilistic security metric. In *Proceedings of the IFIP Annual Conference on Data and Applications Security and Privacy*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 283–296.
26. Poolsappasit, N.; Dewri, R.; Ray, I. Dynamic security risk management using bayesian attack graphs. *IEEE Trans. Dependable Secur. Comput.* **2011**, *9*, 61–74. [\[CrossRef\]](#)
27. Liu, G.; Li, Q.; Zhang, H. Defense strategy generation method for network security based on state attack-defense graph. *J. Comput. Appl.* **2013**, *33* (Suppl. S1), 121–125.
28. Zukhri, Z.; Paputungan, I.V. A hybrid optimization algorithm based on genetic algorithm and ant colony optimization. *Int. J. Artif. Intell. Appl.* **2013**, *4*, 63. [\[CrossRef\]](#)
29. Ye, Z.; Guo, Y.; Wang, C.; Ju, A.K. Survey on application of attack graph technology. *J. Commun.* **2017**, *38*, 121–132.
30. Chung, C.J.; Khatkar, P.; Xing, T.; Lee, J.; Huang, D. NICE: Network intrusion detection and countermeasure selection in virtual network systems. *IEEE Trans. Dependable Secur. Comput.* **2013**, *10*, 198–211. [\[CrossRef\]](#)
31. Luo, Z.; Yang, X.; Liu, J.; Xu, R. Network intrusion intention analysis model based on Bayesian attack graph. *J. Commun.* **2020**, *41*, 160–169.
32. Li, Y.; Wang, C.Z.; Huang, G.Q.; Zhao, X.; Zhang, B.; Li, Y.C. A survey of architecture and implementation method on cyber security situation awareness analysis. *Acta Electron. Sin.* **2019**, *47*, 927.
33. Xiu-juan, W.; Bo, S.; Yan-wen, L.; Cong-bin, X. Computer Network Vulnerability Assessment Based on Bayesian Attribute Network. *J. Beijing Univ. Posts Telecommun.* **2015**, *38*, 110.
34. Mell, P.; Scarfone, K.; Romanosky, S. Common vulnerability scoring system. *IEEE Secur. Priv.* **2006**, *4*, 85–89. [\[CrossRef\]](#)
35. Ruohonen, J. A look at the time delays in CVSS vulnerability scoring. *Appl. Comput. Inform.* **2019**, *15*, 129–135. [\[CrossRef\]](#)
36. Ye, Y.; Xu, X.S.; Jia, Y.; Qi, Z.C. An attack graph-based probabilistic computing approach of network security. *Jisuanji Xuebao (Chin. J. Comput.)* **2010**, *33*, 1987–1996. [\[CrossRef\]](#)
37. Chen, X.; Fang, B.; Tan, Q.; Zhang, H. Inferring attack intent of malicious insider based on probabilistic attack graph model. *Chin. J. Comput.* **2014**, *37*, 62–72.
38. Gao, N.; Gao, L.; He, Y.; Wang, F. Optimal security hardening measures selection model based on Bayesian attack graph. *Comput. Eng. Appl.* **2016**, *52*, 125–130.

-
39. 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.
 40. Harrison, L.; Spahn, R.; Iannaccone, M.; Downing, E.; Goodall, J.R. Nv: Nessus vulnerability visualization for the web. In Proceedings of the Ninth International Symposium on Visualization for Cyber Security, Seattle, WA, USA, 15 October 2012; pp. 25–32.
 41. 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.