Research Article

An Artificial Immune System Algorithm with Social Learning and Its Application in Industrial PID Controller Design

Mingan Wang,[1] Shuo Feng,[2] Chunhui He,[3] Zhonghua Li,[3] and Yu Xue[4]

[1]*School of Information Science and Technology, Huizhou University, Huizhou 516007, China*
[2]*School of Electronic Information and Electrical Engineering, Huizhou University, Huizhou 516007, China*
[3]*School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China*
[4]*School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China*

Correspondence should be addressed to Shuo Feng; fs503@hzu.edu.cn and Zhonghua Li; lizhongh@mail.sysu.edu.cn

A novel artificial immune system algorithm with social learning mechanisms (AIS-SL) is proposed in this paper. In AIS-SL, candidate antibodies are marked with an elitist swarm (ES) or a common swarm (CS). Correspondingly, these antibodies are named ES antibodies or CS antibodies. In the mutation operator, ES antibodies experience self-learning, while CS antibodies execute two different social learning mechanisms, that is, stochastic social learning (SSL) and heuristic social learning (HSL), to accelerate the convergence process. Moreover, a dynamic searching radius update strategy is designed to improve the solution accuracy. In the numerical simulations, five benchmark functions and a practical industrial application of proportional-integral-differential (PID) controller tuning is selected to evaluate the performance of the proposed AIS-SL. The simulation results indicate that AIS-SL has better solution accuracy and convergence speed than the canonical opt-aiNet, IA-AIS, and AAIS-2S.

1. Introduction

Artificial immune systems (AIS) are intelligent computational models or algorithms inspired by the principles of human immune system with the characteristics of self-organization, learning, memory, adaptation, robustness, and scalability [1]. By imitating the immune process of human immune system, AIS has been developed as an effective tool for scientific computing and engineering applications [2], such as function optimization [3, 4], data mining [5], pattern recognition [6, 7], anomaly detection [8, 9], Internet of Things [10], and industrial control systems [11–14].

In the past few years, most AIS and their variations were derived from four major models or theories [1], that is, clone selection, negative selection, danger theory, and artificial immune network. Artificial immune network model is based on the theory of immune network, which states that the immune cells act as a mutually reinforcing network constructed by the matching between paratopes and idiotopes of antibodies. Consequently, there are interactions between not only antibodies and antigens but also antibodies themselves, which causes suppression of the antibodies.

Nunes De Casto and Von Zuben proposed an early version of artificial immune network (aiNet) for data analysis, which is aiming at reducing data redundancy to compress the input set representation [15]. Further, aiNet was developed as opt-aiNet to solve multimodal optimization problems [16]. Due to well-designed immune operators, opt-aiNet is capable of both exploration and exploitation. And soon, Stibor and Timmis presented a closeness measure between the input data set and the compressed output data set by considering the compression quality of opt-aiNet [17]. As a revised version of opt-aiNet, dopt-aiNet [18] was proposed for function optimization in dynamic environments. Inspired by omnioptimization, Coelho and von Zuben proposed an improved omni-aiNet [19], which is able to adjust the population size dynamically. As a result, high redundancy within the population is avoided. Cob-aiNet [20] designed an evaluation method of concentration to direct immune operators. By introducing the elitist-learning policy of PSO, aiNet-EL [21]

and its enhanced version [22] were proposed. Clearly, Both of these two algorithms discriminate elitist antibodies and nonelitist antibodies during mutation operation, and their results indicate that the elitist-learning mechanism is effective to improve the convergence speed. In order to enhance the adaptability of AIS, an improved adaptive artificial immune system (IA-AIS) [23] was designed, where the cloning operator and mutation operator are based on affinity. To maintain the tradeoff between exploration and exploitation, an adaptive AIS algorithm with two swarms, that is, AAIS-2S [24], was proposed. AAIS-2S separates the candidate antibodies into an elitist swarm (ES) and a common swarm (CS) by their affinity, where the antibodies experience self-learning and elitist-learning, respectively. All these achievements indicate that the artificial immune networks have been studied deeply and are of interest to researchers.

Note that in AAIS-2S, CS antibodies just learn from some ES antibody, which brings in the potential risk of falling into the local optimum. This is because the current ES antibody can probably not evolve as the final global optimum in the future, especially in some complex multimodal functions. To alleviate such potential risk, two social learning mechanisms, that is, stochastic social learning (SSL) [25] and heuristic social learning (HSL) [26], were proposed, respectively. As their extended research, this paper integrates SSL with HSL into an AIS algorithm and gets a new algorithm named AIS-SL for complex optimization problems. AIS-SL is able to choose either SSL or HSL as its learning strategy during mutation. In this case, AIS-SL may be subdivided into two different algorithms, that is, AIS-SSL and AIS-HSL. In addition, a dynamic searching radius update strategy is designed. The proposed AIS-SL is expected to quickly capture the optimal solution of complex optimization problems. A series of comparative simulations about benchmarks and industrial PID controller will prove the high-performance and effectiveness of the proposed AIS-SL in both convergence speed and solution accuracy.

The rest of this paper is organized as follows. Section 2 reviews the related theories of immune systems including aiNet, opt-aiNet, IA-AIS, and AAIS-2S. Subsequently, Section 3 presents the theoretical analyses and the technical details of the proposed AIS-SL with different social learning mechanisms, that is, AIS-SSL and AIS-HSL. And Section 4 compares the performances of AIS-SL with opt-aiNet, IA-AIS, and AAIS-2S by a series of numerical simulations and a practical industrial PID application. Finally, Section 5 draws some conclusions.

2. Reviews of Immune Systems

2.1. Human Immune System. In the natural environment, different kinds of pathogens, such as harmful germs and viruses, may flow into our bodies, which lead to deadly damage. Thanks to the powerful immune systems, our bodies can be kept healthy and secure. Once it invades the organism, the pathogen becomes *antigen* and provokes the immune response. If such antigen is detected for the first time, which is referred to as the *primary response*, the immune cells will propagate a mass of cloned *antibodies* to destroy the antigen.

During the propagation, the antibodies undergo somatic hypermutation. The matching between antibody and antigen is measured by *affinity*. The higher the affinity is, the better the candidate antibody matches the specific antigen. In the immune network theories, there is a suppression process because immune cells can also recognize each other. As the reactions continue, immune cells become mature and further differentiate into *plasma cells* or *memory cells*. When the same antigen appears again, memory cells will quickly react, which is referred to as the *secondary response*. It is worth mentioning that the secondary response is triggered much more sharply and instantly than the primary response.

2.2. Artificial Immune System. Generally, most of theoretical problems and engineering applications can be modeled as optimization problems. As a typical AIS model, inspired by immune network theories, aiNet is designed to solve these practical problems. Corresponding to human immune systems, aiNet has some new concepts, where the term *antigen* indicates the objective problem to be solved, the term *antibody* represents the feasible solution of the objective problem, and the term *affinity* is used to evaluate the value of the feasible solution to the objective problem.

In the initialization phase, N candidate antibodies are randomly produced to form the population $\text{Ab}(t) = \{\text{Ab}_1(t), \text{Ab}_2(t), \ldots, \text{Ab}_N(t)\}$, where $\text{Ab}_i(t)$ is the ith individual in the tth time epoch. And the affinity of these antibodies is evaluated by $\text{Aff}(\cdot)$. In each time epoch, any antibody is cloned for a number of offsprings. And then, these clones except for the parent one experience the mutation operator. Only the one with the highest affinity among these clones is selected to remain. If the current average affinity is not significantly different from that of the previous time epoch, the suppression operator is activated. Among antibodies whose similarity or distance is less than a suppression threshold Th_s, all but the one with the highest affinity are suppressed. And then a number of randomly generated antibodies are recruited. Repeat these iterative process until the termination condition is met.

Opt-aiNet [16] is a representative of aiNet algorithms. And it employs a *uniform cloning operator* where each parent antibody is cloned for a fixed number of Nc and introduces an *affinity-based Gaussian mutation* (AGM) where the mutated level is determined by affinity. In addition, the Euclidean distance measure is employed to evaluate similarity, and the suppression threshold is also fixed, which is determined by trial and error. Please see [16] for more details.

The number of clones and suppression threshold of opt-aiNet is fixed, which it is necessary to tune for specific problems. Moreover, the simple proximate linear relationship between mutated antibodies and their affinity impairs the solution accuracy, and the slight difference in mutation scale between antibodies reduces the convergence speed seriously. Taking into account these disadvantages, IA-AIS [23] is proposed to adapt complex optimization problems. In IA-AIS, an *affinity-based* cloning operator is proposed, where the antibody with higher affinity has more offsprings. And a *controlled affinity-based Gaussian mutation* (CAGM) operator is designed to make antibodies with lower affinity mutate
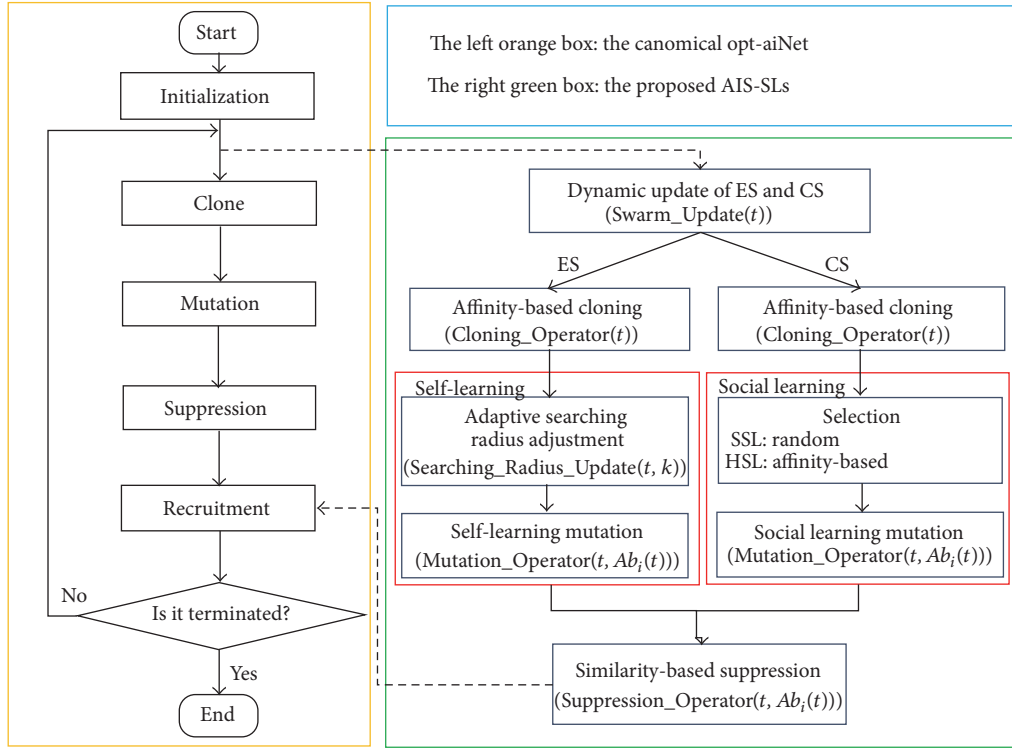
Figure 1: Flowchart of the proposed AIS-SL algorithm.

much more than those with greater affinity. In addition, the suppression threshold can be adjusted dynamically and is proportional to the similarity between antibodies. Please see [23] for more details.

Considering that antibodies in both opt-aiNet and IA-AIS evolve independently, which will impair the exploration and decelerate the convergence process, AAIS-2S [24] is proposed to introduce a learning mechanism. In AAIS-2S, the candidate antibodies are grouped into an elitist swarm (ES) and a common swarm (CS) by their affinity. The ES antibodies experience a self-learning mutation operator, while the CS antibodies go through an elitist-learning mutation operator to accelerate the convergence. And these two swarms are updated every other fixed number of time epochs following a dynamic swarm update strategy, which guarantees that those better CS antibodies can be luckily updated into ES. In addition, the searching radius is adjusted adaptively based on the distance among ES antibodies. Please see [24] for more details.

3. The Proposed Artificial Immune System with Social Learning Mechanisms

In order to improve the solution quality and the convergence speed, this paper proposes an artificial immune system with social learning mechanisms (AIS-SL) for optimization. Figure 1 gives the flowchart of the proposed AIS-SL algorithm. The self-learning mechanism and the social learning mechanism are denoted in the red boxes, respectively. The candidate antibodies are regrouped into ES and CS in the swarm update

phase. ES antibodies experience the affinity-based cloning operator and the self-learning mutation operator, where the searching radius is updated adaptively by a well-designed mechanism. And each CS antibody learns from a target antibody which is selected randomly for AIS-SSL or obtained by using affinity for AIS-HSL. It is worth pointing out that each CS antibody undergoes the social learning mutation operator. Hence, in the mutation process, two social learning scenarios are considered, and the resultant AIS-SSL and AIS-HSL mutation operators are redesigned. The suppression operator is triggered if the suppression condition is met. Repeat the above process until it is terminated.

The technical details of AIS-SL are presented as follows.

3.1. Clone. In AIS-SL, the cloned number of a parent antibody is determined by its affinity. The greater the affinity of parent antibody is, the more offsprings are reproduced. And the number of clones is nonlinear to the affinity of the parent antibody. Thus, the pseudocode of the cloning operator is shown in Algorithm 1, where Nc_{max} and Nc_{min} are the maximum and minimum number of offsprings, respectively, and n is the power factor of the control function. All antibodies, including CS antibodies and ES antibodies, should execute the same cloning operator only once during each time epoch.

3.2. Mutation. The ES antibodies have higher affinity and act as memory cells which react much more sharply and instantly in the secondary immune response. As a result, these ES antibodies play an important role in local search and experience a *self-learning* mutation operator. On the other

```
PROCEDURE Cloning_Operator(t)
 Aff_max = max{Aff(Ab_i(t)) | i = 1, 2, ..., N};
 Aff_min = min{Aff(Ab_i(t)) | i = 1, 2, ..., N};
FOR i = 1 to N DO
    Aff*(i) = (Aff(Ab_i(t)) − Aff_min)/(Aff_max − Aff_min);
    Nc_i(t) = round((Nc_max − Nc_min) * Aff*(Ab_i(t))^n + Nc_min);
END FOR
END PROCEDURE
```

ALGORITHM 1: Pseudocode of the cloning operator.

```
PROCEDURE Mutation_Operator(t, Ab_i(t))
IF Ab_i(t) ∈ ES THEN
    ΔAb_i(t) = rand n * λ_i(t);
ELSE
    Select Ab_j(t) from ES;
    IF Aff(Ab_i(t)) < Aff(Ab_j(t)) THEN
       ΔAb_i(t) = rand * (Ab_j(t) − Ab_i(t));
    ELSEIF Aff(Ab_i(t)) < Aff(Ab_e(t)) THEN
       ΔAb_i(t) = rand * (Ab_e(t) − Ab_i(t));
    ELSE
       ΔAb_i(t) = rand n * λ_i(t);
    END IF
END IF
 Ab_i(t + 1) = Ab_i(t) + ΔAb_i(t);
END FOR
END PROCEDURE
```

ALGORITHM 2: Pseudocode of the mutation operator.

hand, the CS antibodies play a role of global search under the guidance of ES antibodies, so all CS antibodies experience *social learning* to accelerate the convergence process. If the affinity of one CS antibody is less than that of the antibody selected from ES, it learns from the selected ES antibody. Or if its affinity is greater than the selected ES antibody but less than the best ES antibody, it selects the best ES antibody to learn from. Otherwise, it executes the self-learning mutation operator. Consequently, the pseudocode of the mutation operator is shown in Algorithm 2, where rand is a uniform random variable, rand n is a Gaussian random variable with zero mean and standard deviation 1, $Ab_e(t)$ is the best ES antibody in affinity, and $\lambda_i(t)$ is the searching radius of $Ab_i(t)$ in the tth time epoch.

In such canonical AIS as opt-aiNet and IA-AIS, the searching radius is fixed, which impairs the convergence speed and the solution accuracy. It is because that any elitist antibody can easily step out of the optimum if it appears close to the optimum and its searching radius is too large. Or if the searching radius is too small, the convergence speed of AIS will be slowed down. Thus, $\lambda_i(t)$ in AIS-SL can be updated dynamically, which is determined by

$$\lambda_i(t) = \begin{cases} \lambda_i(t-1), & \text{if } \mathrm{Aff}(Ab_i(t)) > \mathrm{Aff}(Ab_i(t-1)), \\ \dfrac{\lambda_i(t-1)}{2}, & \text{elseif } \dfrac{\lambda_i(t-1)}{2} \geq \lambda_{\min}, \\ \lambda_0(k), & \text{otherwise,} \end{cases} \quad (1)$$

where $\lambda_{\min}$ is the lower bound of the searching radius, and

$$\lambda_0(k) = \begin{cases} \dfrac{\lambda_0(k-1)}{2}, & \text{if } \dfrac{\lambda_0(k-1)}{2} \geq \lambda_{\min}, \\ \lambda_0(0), & \text{otherwise.} \end{cases} \quad (2)$$

For $Ab_i(t)$, according to (1) and (2), the initialized searching radius is set to be $\lambda_0(0)$ whose value is equal to the half of the threshold $\mathrm{Th}_s(t)$. After the suppressor, the distance between any two antibodies is greater than $\mathrm{Th}_s(t)$, so $\lambda_0(0) = \mathrm{Th}_s(t)/2$ can maximize the searching area without overlap. If the affinity of $Ab_i(t)$ is improved after mutation, $\lambda_i(t)$ maintains its value. Otherwise, $\lambda_i(t)$ will be halved. However, the searching radius cannot be less than a lower bound $\lambda_{\min}$, because too small searching radius will decrease the convergence speed severely. As a result, when $\lambda_i(t)$ is less than $\lambda_{\min}$, $\lambda_i(t)$ will be set to $\lambda_0(k)$ which is decreased to the half of its previous value $\lambda_0(k-1)$. Also, $\lambda_0(k)$ cannot be less than $\lambda_{\min}$. And $\lambda_0(k)$ will be reset to $\lambda_0(0)$ when it is less than $\lambda_{\min}$. The pseudocode of the dynamic searching radius update strategy is shown in Algorithm 3.

In the proposed AIS-SL, two selection mechanisms are proposed to determine which ES antibody to be learned from. They are *stochastic social learning* (SSL) and *heuristic social learning* (HSL). In SSL, the ES antibody to be learned is selected randomly and the selection probability of every ES antibody is uniform. In other words, the selection probability is not related to the antibody individual but the population scale of ES, that is, N_{elitist} ($N_{\text{elitist}} = N * p_{\text{elitist}}$, $p_{\text{elitist}} \in (0, 1)$). In HSL, the ES antibody to be learned is determined by affinity. In detail, the selection probability $p(Ab_j(t))$ for the ES antibody $Ab_j(t)$ follows

$$p\left(Ab_j(t)\right) = \frac{\mathrm{Aff}^*\left(Ab_j(t)\right)}{\sum_{i=1}^{N_{\text{elitist}}} \mathrm{Aff}^*\left(Ab_i(t)\right)}. \quad (3)$$

Obviously, the selection probability is nonuniform. The higher the affinity is, the greater selection probability the antibody has. And then, the *roulette* method is employed for HSL to select the ES antibody to learn.

Assume that the optima of multimodal function is significantly different in affinity. The higher the affinity is, the greater probability the antibody has to evolve as the global optimum. So, AIS-HSL will have faster convergence speed. However, if ES does not overlap the global optimum, AIS-HSL has risk to fall into some local optima. If the optima

```
PROCEDURE Searching_Radius_Update(t, k)
IF  k = 0  THEN
    λ_0(k) = Th_s/2;
END IF
IF  t = 0  THEN
    λ_i(t) = Th_s(t)/2;
ELSEIF Aff(Ab_i(t)) > Aff(Ab_i(t − 1))  THEN
    λ_i(t) = λ_i(t − 1);
ELSEIF λ_i(t − 1)/2 ≥ λ_min  THEN
    λ_i(t) = λ_i(t − 1)/2;
ELSE
    IF λ_0(k − 1)/2 ≥ λ_min  THEN
        λ_0(k) = λ_0(k − 1)/2;
    ELSE
        λ_0(k) = λ_0(0);
    END IF
    λ_i(t) = λ_0(k);
    k = k + 1;
END IF
END PROCEDURE
```

ALGORITHM 3: Pseudocode of the dynamic searching radius update strategy.

```
PROCEDURE Suppression_Operator(t)
D_max = max{D_{i,j}(t) | i, j = 1, 2, …, N,  i ≠ j};
D_min = min{D_{i,j}(t) | i, j = 1, 2, …, N,  i ≠ j};
Th_s(t) = D_min + ξ * (D_max − D_min);
END PROCEDURE
```

ALGORITHM 4: Pseudocode of the suppression operator.

are similar in affinity, AIS-HSL will have similar performance to AIS-SSL. In the extremely special condition where ES has only one antibody, two algorithms are same in effect.

3.3. Suppression. The proposed AIS-SL employs the dynamic suppressor which is the same to IA-AIS. And the threshold $Th_s(t)$ is proportional to the similarity of the antibody population, whose pseudocode is shown in Algorithm 4, where $D_{i,j}$ is the Euclidean distance between the ith antibody and the jth antibody in the tth time epoch, and $\xi \in (0, 1)$. After the suppression, a number of randomly generated antibodies are recruited to keep the population size at the number of N.

3.4. Swarm Update. As we know, it is possible that some CS antibodies capture higher affinity than those ES antibodies due to the elitist-learning. Consequently, it is necessary to update two swarms and allow these better CS antibodies to enter into ES. Thus, in the swarm update, all antibodies are sorted by their affinity in a *descending* order. Then, the first $N_{elitist}$ antibodies step into ES and the others step into CS. Note that after the initialization, all antibodies experience the self-learning mechanism until the suppression operator is triggered. This allows all antibodies to evolve within an

enough small region around them. So it is helpful to find the peaks of the desired problem. As a result, in the first swarm update, all antibodies belong to ES, while CS is empty. The pseudocode of the swarm update mechanism is shown in Algorithm 5.

4. Numerical Simulations and Results

In the numerical simulations, five benchmark functions are selected from *2005 CEC Special Session on Real-Parameter Optimization* [27] to examine the solution accuracy and the convergence speed of AIS-SL. Furthermore, as a practical industrial application, the PID controller tuning is considered to evaluate the proposed AIS-SL. In addition, AIS-SL is compared with opt-aiNet [16], IA-AIS [23], and AAIS-2S [24] in optimization performance.

4.1. Benchmark Functions. The five selected benchmark functions are listed as follows:

   (i) F_2-Shifted Schwefel's Problem 1.2: unimodal;

   (ii) F_6-Shifted Rosenbrock's Function: multimodal;

   (iii) F_{10}-Shifted Rotated Rastrigin's Function: multimodal;

   (iv) F_{11}-Shifted Rotated Weierstrass Function: multimodal;

   (v) F_{12}-Schwefel's Problem 2.13: multimodal.

4.2. Parameter Sensitivity Analysis. There are three key parameters in the proposed AIS-SL, that is, the population scale of ES ($p_{elitist}$), the power factor of the cloning operator (n), and the suppression scale (ξ), which need to be tuned. In this subsection, five benchmark functions in 10D are used to discuss the impact of these three parameters. For the sake of clear observations, the results are expressed in lg (i.e., $\log 10$) scale.

Figure 2 represents the impact of the population scale of ES $p_{Elitist}$ on error, where $n = 2$ and $\xi = 0.1$. As seen from Figures 2(a) and 2(b), $p_{elitist}$ has little impact on error for F_2, F_{10}, and F_{11}. However for F_{12}, the error is much larger when $p_{elitist}$ is smaller than 0.2. And the impact is more significant for F_6, where the error is much greater when $p_{elitist}$ is greater than 0.4 for SSL and 0.5 for HSL. The reason can be got in Figure 3, which shows the impact of $p_{elitist}$ on the convergence speed for F_9. The greater $p_{elitist}$ is, the slower the convergence speed of the proposed AIS-SL is. As a result, $p_{elitist}$ is usually set to be 0.2–0.4.

Figure 3 represents the impact of the power factor (n) of the cloning operator on error, where $p_{Elitist} = 0.25$ and $\xi = 0.1$. As seen from Figures 3(a) and 3(b), it is obvious that n has little impact on error for all the benchmark functions. For each function, no matter what value n has, both of AIS-SSL and AIS-HSL have similarity optimized results.

Figure 4 represents the impact of the suppression scale ξ on error, where $p_{Elitist} = 0.25$ and $n = 2$. As seen from Figures 4(a) and 4(b), ξ has little impact on error for F_2, F_{10}, and F_{11}. But for F_6 optimized by AIS-HSL, the error is much larger

```
PROCEDURE Swarm_Update(t)
IF t = 0 THEN
      ES = Ab(t);
      CS = ∅;
ELSE
   Sort Ab(t) by affinity in a descending order, i.e., Aff(Ab_{i1}(t)) ≥ ⋯ ≥ Aff(Ab_{iN}(t));
      ES = {Ab_{i1}(t), ..., Ab_{iNelitist}(t)};
      CS = {Ab_{iNelitist+1}(t), ..., Ab_{iN}(t)};
END IF
 Ab_i(t + 1) = Ab_i(t) + ΔAb_i(t);
END PROCEDURE
```
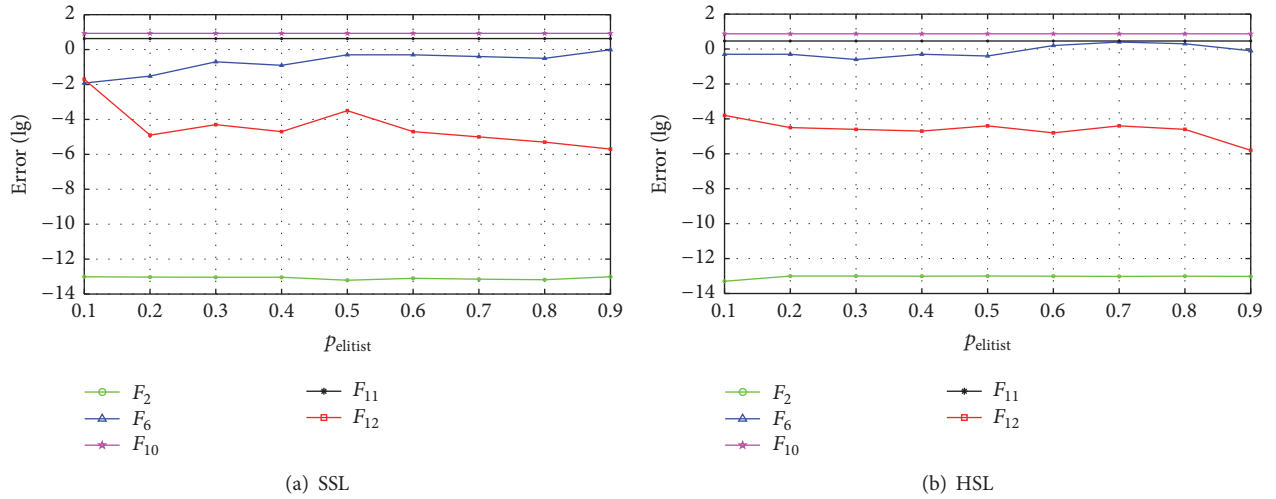
ALGORITHM 5: Pseudocode of the swarm update strategy.

FIGURE 2: Impact of p_{elitist} on error.

TABLE 1: Parameters of the proposed AIS-SL algorithm including AIS-SSL and AIS-HSL.

Operator	Cloning	Mutation	Suppressor	Others
Parameter	$n = 2$	$\lambda_{\min} = 1e-15$	$\xi = 0.2$	$p_{\text{elitist}} = 0.25$

when ξ is greater than 0.3. And for F_{12}, the error increases with ξ, when ξ is greater than 0.3 for SSL and 0.2 for HSL. And the error is also larger if p_{Elitist} is smaller than 20% for F_{12}. Above all, ξ is usually set to be 0.2–0.3.

4.3. Parameter Settings. For the sake of fairness, the initialized population size of these five algorithms is 100, the maximum number of time epochs is 1000 for 2D and 5000 for 10D, and all the simulations are repeated for 50 trials. In the cloning operator, the number of clones for opt-aiNet is 20, while the maximum and minimum numbers of clones are 20 and 4 for the other algorithms, respectively. According to the results of parameter sensitivity analysis, all the other parameters of the proposed AIS-SL are listed in Table 1. Please see [16, 20, 21] for the rest parameters of the other three algorithms.

4.4. Numerical Simulation Results. Tables 2 and 3 present the numerical simulation results for five benchmark functions, including the best, worst, mean, and standard deviation (Std) of error in 2D and 10D, respectively, where the best results are typed in bold. As seen from the results, both AIS-SSL and AIS-HSL have perfect performance. In the simulations of 2D especially, the proposed AIS-SSL and AIS-HSL can both capture the global optimum for every benchmark function in every trial. For 10D, although the proposed AIS-SSL and AIS-HSL cannot reach the global optimum in every trial, they are able to obtain desired solutions which are much better than those of opt-aiNet, IA-AIS, and AAIS-2S. It is obvious that the proposed AIS-SL has much more accurate solutions than opt-aiNet, IA-AIS, and AAIS-2S. Further, by comparing the results of AIS-SSL and AIS-HSL, it is easy to find that they perform equally for 2D problems while it is hard to tell which is better.

Figures 5–9 show the average convergence process for five benchmark functions, respectively. In order to distinguish the results obviously, the results are expressed in lg (i.e., log 10) scale. In every subfigure of Figures 5–9, it is obvious that the proposed AIS-SSL and AIS-HSL have much faster convergence speed than opt-aiNet, IA-AIS, and AAIS-2S. For
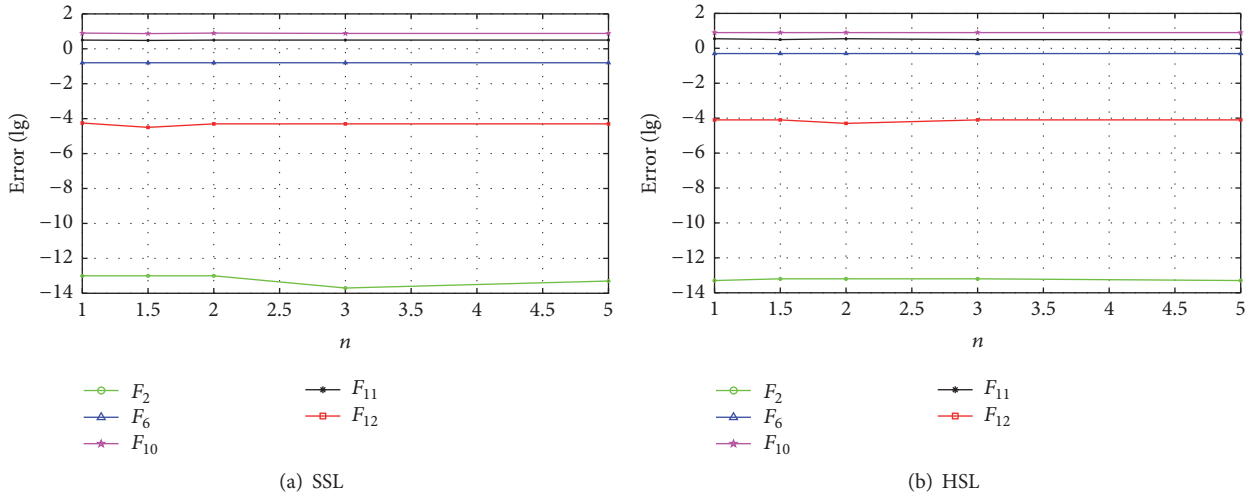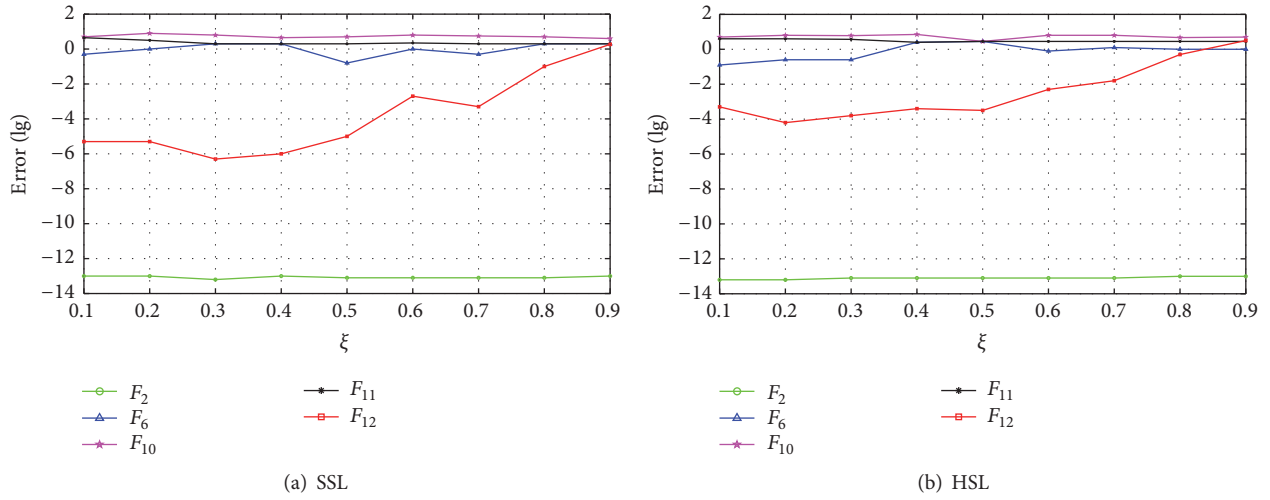
(a) SSL

(b) HSL

Figure 3: Impact of n on error.
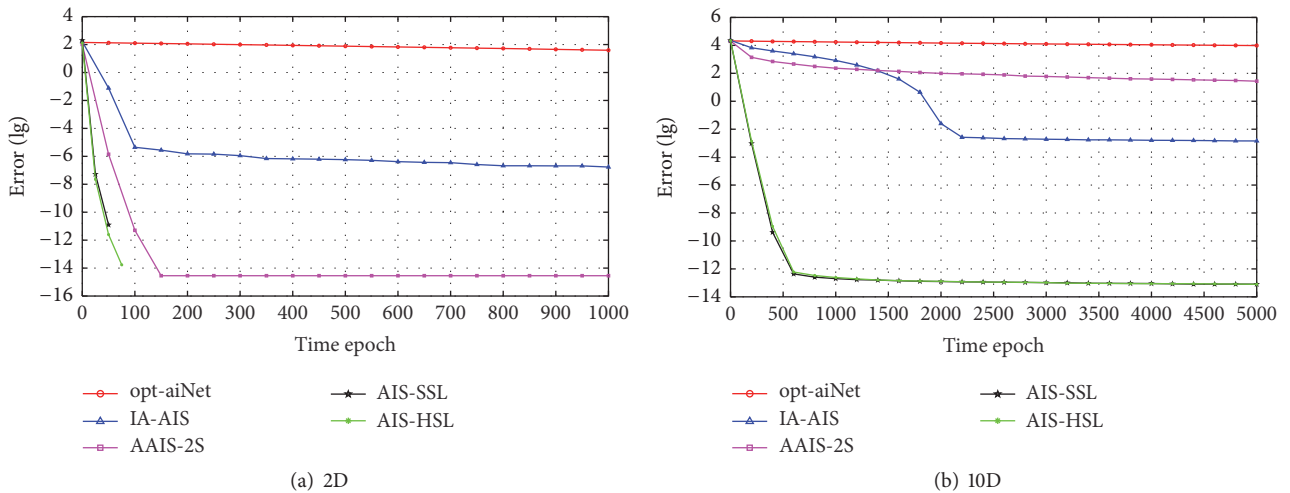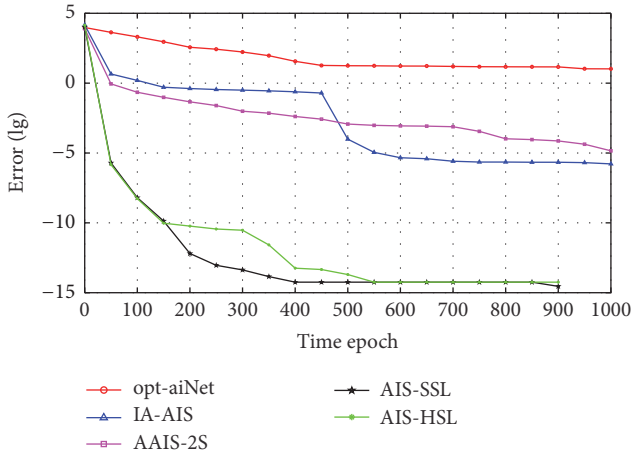

(a) SSL

(b) HSL

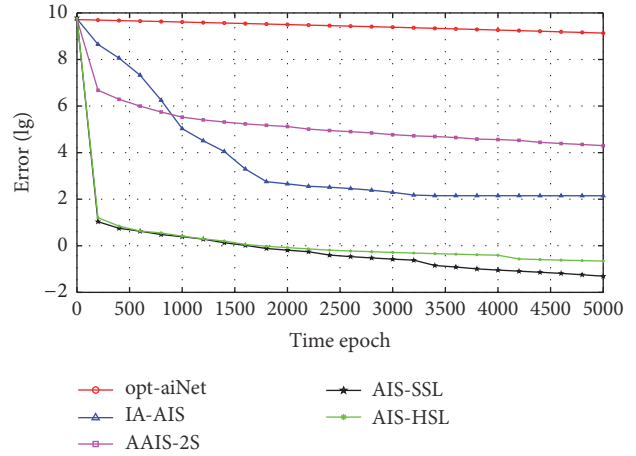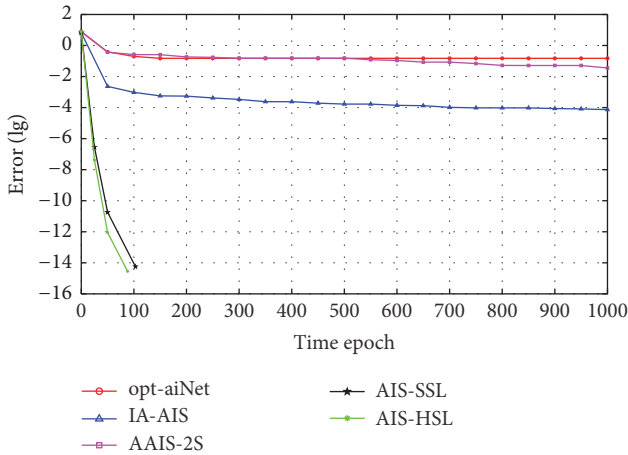Figure 4: Impact of ξ on error.

(a) 2D

(b) 10D

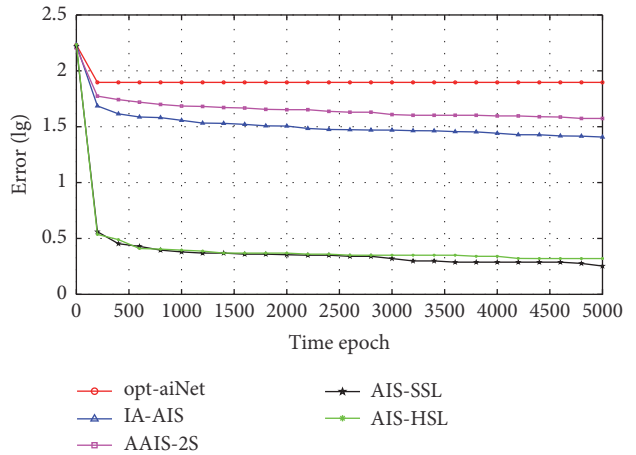Figure 5: Average convergence results of F_2.
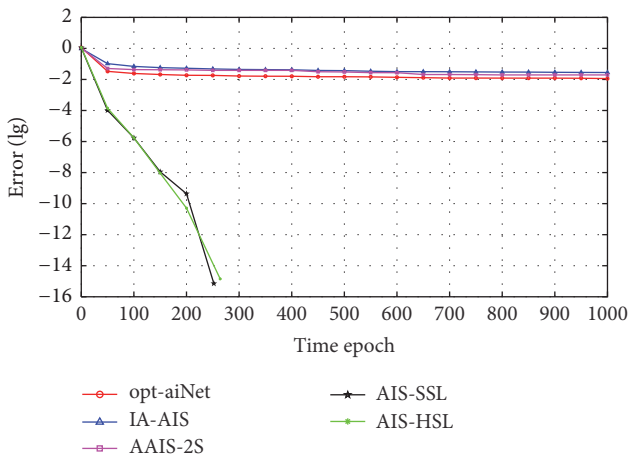
(a) 2D

(b) 10D

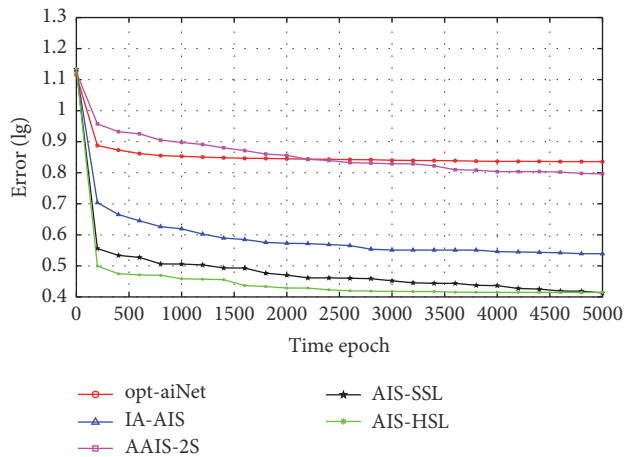FIGURE 6: Average convergence results of F_6.

(a) 2D

(b) 10D

FIGURE 7: Average convergence results of F_{10}.

(a) 2D

(b) 10D

FIGURE 8: Average convergence results of F_{11}.

TABLE 2: Numerical simulation results in 2D (the best is in bold).

| Benchmark	Algorithm	Best	Worst	Mean	Std
F_2	opt-aiNet	$8.334382e-010$	$7.423637e+002$	$4.424344e+001$	$1.172033e+002$
	IA-AIS	$5.649667e-010$	$6.458520e-007$	$2.181431e-007$	$1.829096e-007$
	AAIS-2S	**0**	$5.684342e-014$	$1.136868e-015$	$8.038873e-015$
	AIS-SSL	**0**	**0**	**0**	**0**
	AIS-HSL	**0**	**0**	**0**	**0**
F_6	opt-aiNet	$1.601273e-008$	$6.982493e+001$	$1.332500e+001$	$1.453700e+001$
	IA-AIS	$7.696849e-008$	$6.500257e-006$	$1.330222e-006$	$1.461563e-006$
	AAIS-2S	**0**	$2.765851e-002$	$5.749097e-004$	$3.910412e-003$
	AIS-SSL	**0**	**0**	**0**	**0**
	AIS-HSL	**0**	**0**	**0**	**0**
F_{10}	opt-aiNet	$1.741910e-009$	$9.949597e-001$	$2.188915e-001$	$4.163424e-001$
	IA-AIS	$1.129530e-007$	$2.117819e-004$	$6.593969e-005$	$5.480150e-005$
	AAIS-2S	**0**	$7.527344e-001$	$5.553388e-002$	$1.347386e-001$
	AIS-SSL	**0**	**0**	**0**	**0**
	AIS-HSL	**0**	**0**	**0**	**0**
F_{11}	opt-aiNet	$4.649715e-003$	$2.097592e-002$	$1.119078e-002$	$3.723280e-003$
	IA-AIS	$5.416757e-003$	$4.992415e-002$	$2.602879e-002$	$1.028627e-002$
	AAIS-2S	**0**	$1.332238e-001$	$1.798648e-002$	$2.996216e-002$
	AIS-SSL	**0**	**0**	**0**	**0**
	AIS-HSL	**0**	**0**	**0**	**0**
F_{12}	opt-aiNet	$7.541985e-010$	$5.799400e-007$	$6.917494e-008$	$9.323257e-008$
	IA-AIS	$8.398080e-007$	$1.160427e-004$	$2.146887e-005$	$2.347865e-005$
	AAIS-2S	**0**	$3.548364e-001$	$2.313225e-002$	$6.041159e-002$
	AIS-SSL	**0**	**0**	**0**	**0**
	AIS-HSL	**0**	**0**	**0**	**0**

(a) 2D

(b) 10D
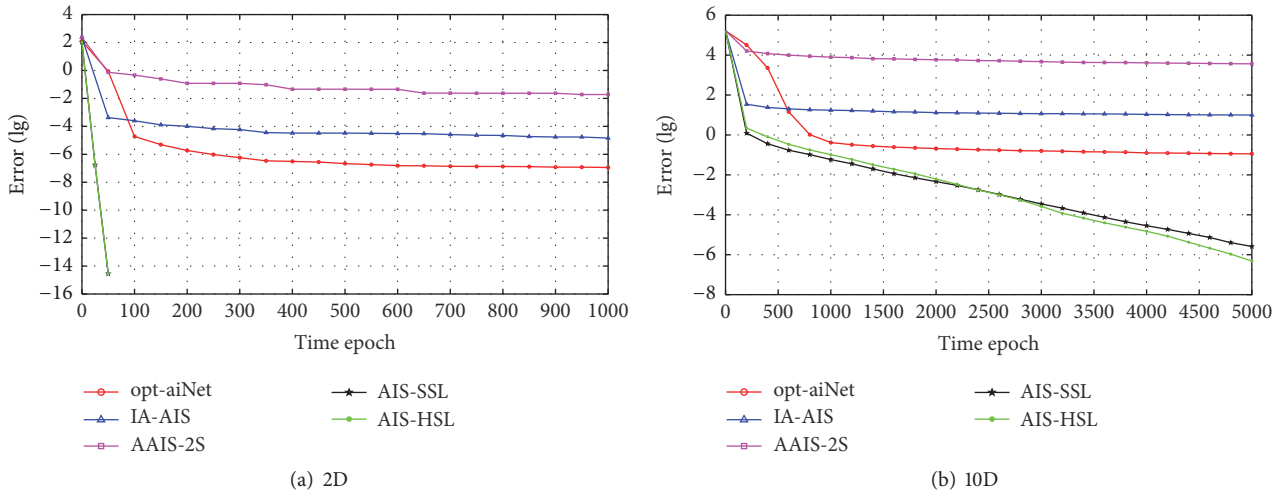
FIGURE 9: Average convergence results of F_{12}.

example in 2D, AIS-SSL and AIS-HSL can both capture the optima easily within about 100, 900, 100, 300, and 50 time epochs for F_2, F_6, F_{10}, F_{11}, and F_{12}, respectively, while the curves of the other algorithms descend more slowly.

4.5. An Industrial Application in PID Controller Design. Proportional integral derivative (PID) controllers are the famous process control techniques. Because PID controllers have simple structure and robust performance, they are widely used in most control systems. In a general PID control system, as shown in Figure 10, there are three components, that is, the proportional component K_p, the differential component $K_d s$, and the integral component K_i/s, where K_p, K_d, and K_i are the gain factors of three control components, respectively. They have great effect on the performance of control systems. K_p controls the response speed of system, K_d controls the

TABLE 3: Numerical simulation results in 10D (the best is in bold).

| Benchmark	Algorithm	Best	Worst	Mean	Std
F_2	opt-aiNet	$2.332892e + 003$	$1.695573e + 004$	$9.691981e + 003$	$3.404834e + 003$
	IA-AIS	$6.819255e - 004$	$6.819255e - 004$	$1.439519e - 003$	$3.506100e - 004$
	AAIS-2S	$5.654689e + 000$	$7.858485e + 001$	$2.721496e + 001$	$1.389431e + 001$
	AIS-SSL	**0**	**$1.705303e - 013$**	$8.071765e - 014$	$3.824397e - 014$
	AIS-HSL	**0**	$2.273737e - 013$	**$7.844392e - 014$**	**$3.789755e - 014$**
F_6	opt-aiNet	$4.059666e + 007$	$4.284734e + 009$	$1.364979e + 009$	$9.506303e + 008$
	IA-AIS	$1.919524e + 000$	$1.112680e + 003$	$1.404718e + 002$	$2.618577e + 002$
	AAIS-2S	$3.556545e + 003$	$9.837639e + 004$	$1.989282e + 004$	$1.866355e + 004$
	AIS-SSL	**$3.824573e - 004$**	**$4.213768e - 001$**	**$4.943018e - 002$**	**$9.687442e - 002$**
	AIS-HSL	$4.745545e - 003$	$4.559531e + 000$	$2.205072e - 001$	$6.743634e - 001$
F_{10}	opt-aiNet	$2.587223e + 001$	$1.303397e + 002$	$7.886253e + 001$	$2.539665e + 001$
	IA-AIS	$8.496231e + 000$	$3.733400e + 001$	$2.555761e + 001$	$7.982356e + 000$
	AAIS-2S	$2.330297e + 001$	$4.429763e + 001$	$3.749634e + 001$	$5.579124e + 000$
	AIS-SSL	**0**	$2.984877e + 000$	$1.790958e + 000$	$1.134428e + 000$
	AIS-HSL	$9.949591e - 001$	$3.979836e + 000$	$2.089414e + 000$	$1.218571e + 000$
F_{11}	opt-aiNet	$5.696060e + 000$	$8.005699e + 000$	$6.854062e + 000$	$7.114406e + 000$
	IA-AIS	$2.948037e + 000$	$3.949336e + 000$	$3.459549e + 000$	$8.412727e - 001$
	AAIS-2S	$4.558491e + 000$	$8.100967e + 000$	$6.253723e + 000$	$8.078057e + 000$
	AIS-SSL	$8.477571e - 001$	**$3.847625e + 000$**	**$2.588361e + 000$**	$7.297696e - 001$
	AIS-HSL	**$7.120791e - 001$**	$3.867331e + 000$	$2.597338e + 000$	**$6.596910e - 001$**
F_{12}	opt-aiNet	$4.483347e - 002$	$1.999181e - 001$	$1.139036e - 001$	$3.514291e - 002$
	IA-AIS	$6.018482e + 000$	$1.337908e + 001$	$9.997727e + 000$	$1.814684e + 000$
	AAIS-2S	$9.189471e + 002$	$7.018537e + 003$	$3.657929e + 003$	$1.363907e + 003$
	AIS-SSL	$1.017497e - 010$	$1.176995e - 004$	$2.600616e - 006$	$1.665200e - 005$
	AIS-HSL	**$2.302158e - 011$**	**$1.721262e - 005$**	**$4.821827e - 007$**	**$2.425346e - 006$**
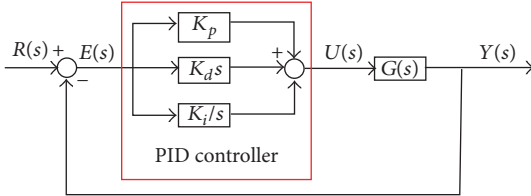
FIGURE 10: The schematic diagram of a general PID controlled system.

dynamic performance, and K_i controls the steady state error. As a result, the design of a PID controller is to select an optimum solution (K_p, K_d, K_i).

To optimize the performance of a PID controlled system, four error indices are selected to be minimized, that is, the integral of the absolute magnitude of the error (IAE), the integral of the square of the error (ISE), the integral of time multiplied by the absolute error (ITAE), and the mean of the square of the error (MSE). These indices are formulated as follows.

$$I_{IAE} = \int_0^T |e(t)| \, dt;$$

$$I_{ISE} = \int_0^T e^2(t) \, dt;$$

$$I_{ITAE} = \int_0^T t\,|e(t)| \, dt;$$

$$I_{MSE} = \sum_{k=1}^n \frac{e^2(kT/n)}{n},$$

$$(4)$$

where T is the integral time and normally no more than the settle time of the desired system, and $e(t)$ is the error between the input and the output.

Assume that the transfer function of a certain controlled process is given by

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{s(s^3 + 6s^2 + 11s + 6)},$$ (5)

and the transfer function of the embedded PID controller is expressed by

$$D_c(s) = K_p + K_d s + \frac{K_i}{s}.$$ (6)

Once the controller parameters (K_p, K_d, K_i) are determined, the corresponding step response curve of the PID controlled system can be figured out. Hence, we can use some analytical methods or intelligent optimization methods to calculate the controller according to 4.5. For the sake of

(a) IAE
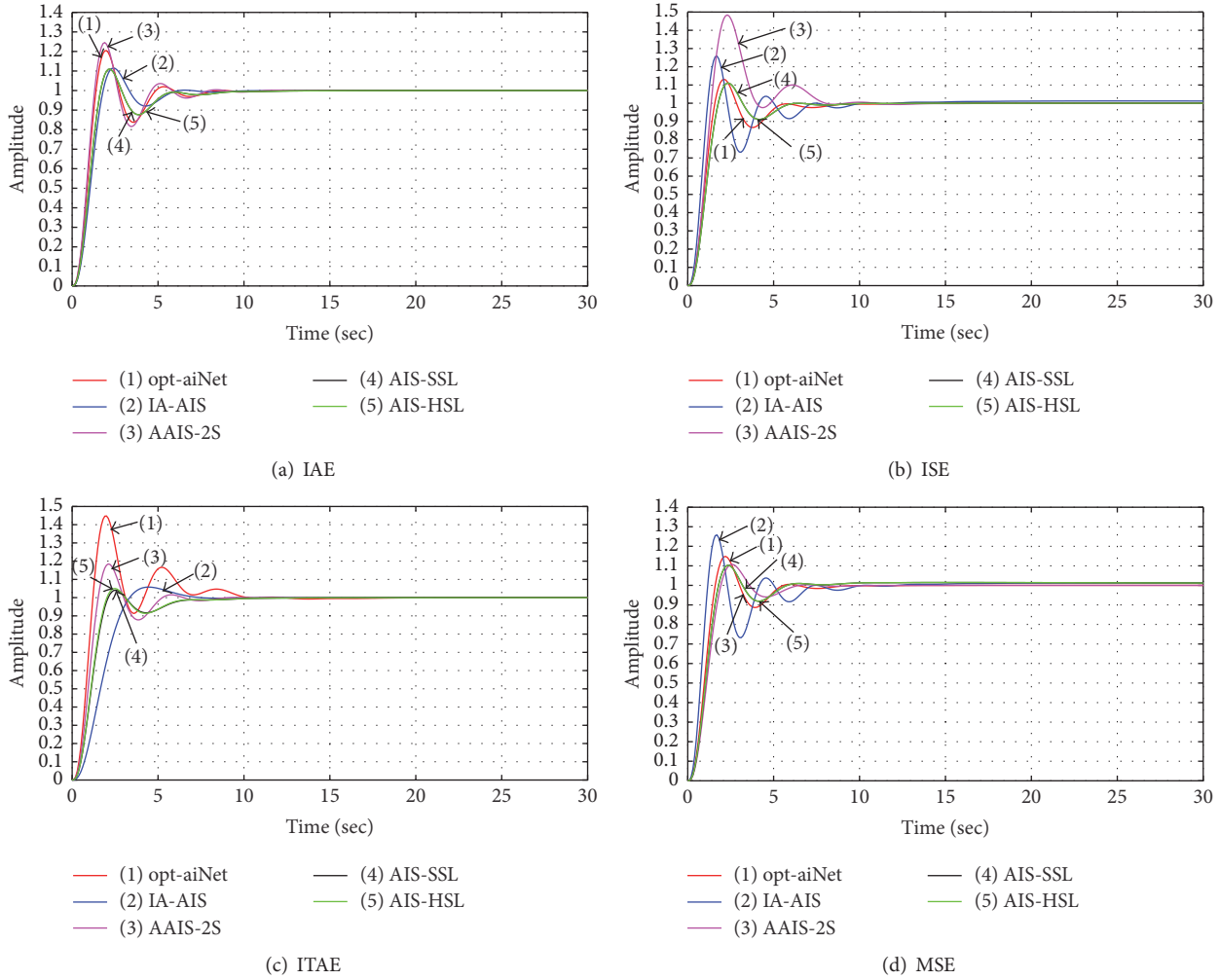

(b) ISE

(c) ITAE

(d) MSE

Figure 11: Step responses of PID controller tuned by five methods to minimize IAE, ISE, ITAE, and MSE.

comparisons, in this application, three optimization methods (i.e., opt-aiNet, IA-AIS, and AAIS-2S) and the proposed AIS-SL (i.e., AIS-SSL and AIS-HSL) algorithms are used to search the best PID controller parameters. The corresponding parameters are set the same as in Section 4.3 and Table 1. For further analysis, the best solutions (K_p, K_d, K_i) optimized by the proposed AIS-SSL and AIS-HSL are specially given in Table 4.

Correspondingly, the step response curves of the PID control system using opt-aiNet, IA-AIS, AAIS-2S, AIS-SSL, and AIS-HSL can be presented in Figure 11. It is obvious that AIS-SSL and AIS-HSL have the least settling time and the smallest overshoot and can obtain the smallest value in all of the four indices by comparison with other three methods. Particularly, PID controller obtains ideal step response curves which are tuned by the proposed AIS-SSL or AIS-HSL when ITAE is used, because their overshoots are both less than 5% which is seen from the red and black plots shown in Figure 11(c). For instance in Figure 11(c), AIS-SSL obtains the overshoot of only about 4% and the settle time of 5.9 s, while AIS-HSL gets the overshoot of about 5% and the settling

Table 4: The best solutions (K_p, K_d, K_i) optimized by AIS-SSL and AIS-HSL.

Error index	AIS-SSL	AIS-HSL
IAE	$(5.212351e + 0,$ $1.178512e + 1,$ $6.281795e - 4)$	$(5.212400e + 0,$ $1.178509e + 1,$ $6.251776e - 4)$
ISE	$(5.470874e + 0,$ $1.026061e + 1,$ $8.098120e - 4)$	$(5.470873e + 0,$ $1.026061e + 1,$ $8.098113e - 4)$
ITAE	$(4.547576e + 0,$ $9.014709e + 0,$ $5.260469e - 6)$	$(4.652396e + 0,$ $9.294985e + 0,$ $7.234183e - 6)$
MSE	$(5.270874e + 0,$ $1.026061e + 1,$ $8.098122e - 2)$	$(5.270873e + 0,$ $1.026061e + 1,$ $8.098108e - 2)$

time of 5.8 s. And in Figures 11(a), 11(b), and 11(d), AIS-SSL and AIS-HSL have similar performance due to their approximately overlapping step response curves, where the

results in overshoot are 10.97%, 10.99%, and 10.23%, and results in settling time are 5.2 s, 5.4 s, and 5.2 s, respectively.

In the details of comparisons, AIS-SL harvests improvement in overshoot and settling time of AAIS-2S by 10.90% and 26.76% for IAE and by 25.19% and 27.03% for ISE. Furthermore, for ITAE, AIS-SSL (or AIS-HSL) reduces the overshoot and the settling time of opt-aiNet by 28.09% (or 27.46%) and 35.87% (or 36.96%). And for MSE, compared with IA-AIS, AIS-SL improves the overshoot and the settling time by 12.38% and 23.53%. These results fully prove that the proposed AIS-SL is more effective in tuning a PID controller.

5. Conclusions

This paper proposes an artificial immune system algorithm with social learning mechanisms (AIS-SL) for complex optimization problems. Considering that the simple elitist-learning has great risk of falling into the optima, the proposed AIS-SL considers two social learning mechanisms, that is, stochastic social learning (SSL) and heuristic social learning (HSL). In addition, a dynamic searching radius update strategy is proposed to improve the solution accuracy. In the numerical simulations, the performance of the proposed AIS-SL is compared with opt-aiNet, IA-AIS, and AAIS-2S in five benchmark functions and a practical application of PID controller tuning. According to the numerical simulation results, both AIS-SSL and AIS-HSL can obtain the global optimum or desired solutions much more quickly and more accurately and are more effective in such an industrial PID application than opt-aiNet, IA-AIS, and AAIS-2S.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

References

[1] J. Timmis, A. Hone, T. Stibor, and E. Clark, "Theoretical advances in artificial immune systems," *Theoretical Computer Science*, vol. 403, no. 1, pp. 11–32, 2008.

[2] D. Dasgupta, S. Yu, and F. Nino, "Recent advances in artificial immune systems: models and applications," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 1574–1587, 2011.

[3] R. Shang, L. Jiao, F. Liu, and W. Ma, "A novel immune clonal algorithm for MO problems," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 35–50, 2012.

[4] S. Yang and X. Yao, "Population-based incremental learning with associative memory for dynamic environments," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 5, pp. 542–561, 2008.

[5] T. D. Do, S. C. Hui, A. C. M. Fong, and B. Fong, "Associative classification with artificial immune system," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 217–228, 2009.

[6] A. J. Graaff and A. P. Engelbrecht, "A local network neighbourhood artificial immune system for data clustering," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 260–267, IEEE, Singapore, September 2007.

[7] A. A. Freitas and J. Timmis, "Revisiting the foundations of artificial immune systems for data mining," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 4, pp. 521–540, 2007.

[8] P. K. Harmer, P. D. Williams, G. H. Gunsch, and G. B. Lamont, "An artificial immune system architecture for computer security applications," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pp. 252–280, 2002.

[9] J. Greensmith, J. Twycross, and U. Aickelin, "Dendritic cells for anomaly detection," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 664–671, Vancouver, Canada, July 2006.

[10] M. Wang, S. Feng, C. Ouyang, and Z. Li, "RFID tag oriented data allocation method using artificial immune network," in *Proceedings of the 27th Chinese Control and Decision Conference (CCDC '15)*, pp. 5218–5223, IEEE, Qingdao, China, May 2015.

[11] Z. Li and C. He, "Optimal scheduling-based RFID reader-to-reader collision avoidance method using artificial immune system," *Applied Soft Computing Journal*, vol. 13, no. 5, pp. 2557–2568, 2013.

[12] H. P. Pan, W. H. Wang, and J. F. Gao, "Study on a variable arguments PID controller based on improved artificial immune algorithm," in *Proceedings of the 30th Chinese Control Conference*, pp. 3752–3755, Yantai, China, July 2011.

[13] T. Jansen and C. Zarges, "Reevaluating immune-inspired hypermutations using the fixed budget perspective," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 5, pp. 674–688, 2014.

[14] F. P. A. Lima, M. L. M. Lopes, A. D. P. Lotufo, and C. R. Minussi, "An artificial immune system with continuous-learning for voltage disturbance diagnosis in electrical distribution systems," *Expert Systems with Applications*, vol. 56, pp. 131–142, 2016.

[15] L. Nunes De Casto and F. J. Von Zuben, "An evolutionary immune network for data clustering," in *Proceedings of the 6th Brazilian Symposium on Neural Networks (SBRN '00)*, vol. 1, io de Janeiro, Brazil, November 2000.

[16] L. N. de Castro and J. Timmis, "An artificial immune network for multimodal function optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '02)*, vol. 1, pp. 699–704, IEEE, Honolulu, Hawaii, USA, May 2002.

[17] T. Stibor and J. Timmis, "An investigation on the compression quality of aiNet," in *Proceedings of the IEEE Symposium on Foundations of Computational Intelligence (FOCI '07)*, pp. 495–502, IEEE, Honolulu, Hawaii, USA, April 2007.

[18] F. O. De França, F. J. Von Zuben, and L. N. De Castro, "An artificial immune network for multimodal function optimization on dynamic environments," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '05)*, pp. 289–296, Washington, DC, USA, June 2005.

[19] G. P. Coelho and F. J. von Zuben, "Omni-aiNet: an immune-inspired approach for omni optimization," in *Proceedings of the 5th International Conference on Artificial Immune System*, pp. 294–308, Springer, Oeiras, Portugal, September 2006.

[20] G. P. Coelho and F. J. von Zuben, "A concentration-based artificial immune network for continuous optimization," in *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 1–8, Barcelona, Spain, July 2010.

[21] Z. H. Li, Y. N. Zhang, and H.-Z. Tan, "An efficient artificial immune network with elite-learning," in *Proceedings of the 3rd International Conference on Natural Computation (ICNC '07)*, pp. 213–217, IEEE, Haikou, China, August 2007.

[22] Z. H. Li, Y. N. Zhang, H.-Z. Tan, and Z. Y. Chen, "An enhanced artificial immune network with elitist-learning capability for optimization problems," *Control Theory & Applications*, vol. 26, no. 3, pp. 283–290, 2009.

[23] Z. Li, Y. Zhang, and H.-Z. Tan, "IA-AIS: an improved adaptive artificial immune system applied to complex optimization problems," *Applied Soft Computing*, vol. 11, no. 8, pp. 4692–4700, 2011.

[24] Z. H. Li, C. H. He, and H.-Z. Tan, "The AAIS-2S optimization: an efficient adaptive artificial Immune System with Two Swarms," in *Proceedings of the 4th International Conference on Computational Intelligence and Software Engineering*, pp. 5–8, Scientific Research, Wuhan, China, December 2012.

[25] Z. Li and C. He, "The AIS-SoL optimization: an artificial immune system with social learning," in *Proceedings of the 8th International Conference on Computational Intelligence and Security (CIS '12)*, Guangzhou, China, November 2012.

[26] Z. H. Li and C. H. He, "The AIS-HSL optimization: an artificial immune system with heuristic social learning," in *Proceedings of the 2012 IET International Conference on Information Science and Control Engineering*, pp. 306–310, Institution of Engineering and Technology, Shenzhen, China, December 2012.

[27] P. N. Suganthan, N. Hansen, J. J. Liang et al., "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Tech. Rep., Nanyang Technological University, Singapore, 2005.