Applied Soft Computing 11 (2011) 11–22

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Hybrid GA–BF based intelligent PID controller tuning for AVR system

Dong Hwa Kim [*]

Dept. of Instrumentation and Control Eng., Hanbat National University, 16-1 San Duckmyong-Dong Yuseong-Gu, Daejeon City 305-719, Republic of Korea

ARTICLE INFO

Article history:
Received 19 September 2005
Received in revised form 5 November 2007
Accepted 9 January 2009
Available online 24 October 2009

Keywords:
Genetic Algorithm
Bacterial Foraging optimization
Hybrid system
Optimal algorithm

ABSTRACT

This paper deals with hybrid system (GA–BF) based on the conventional GA (Genetic Algorithm) and BF (Bacterial Foraging) which is the social foraging behavior of bacteria. A variety of test function is introduced and simulated to illustrate the characteristics and performance by mutation, crossover, variation of step size, variation of chemotactic step, and variation of lifetime of bacteria in the proposed hybrid system GA–BF. The simulated results represent that the proposed method is highly satisfactory. This approach provides us with novel hybrid model based on foraging behavior and also with a possible new connection between evolutionary forces in social foraging and distributed nongradient optimization algorithm design for global optimization over noisy surfaces.

1. Introduction

In the last decade, evolutionary computation based approaches have received increased attention from the engineers dealing with problems which could not be solved using conventional problem solving techniques [1–7]. A typical task of a GA in this context is to find the best values of a predefined set of free parameters associated with either a process model or a control vector. One of the active areas of research in GA is system identification [12–16]. A recent survey of evolutionary algorithms for the evaluation of improved learning algorithm and control system engineering can be found in Ref. [12,16,18]. The general problem of evolutionary algorithm based engineering system design has been tackled in various ways. GA has also been used to optimize nonlinear system strategies. Among them, a large amount of research is focused on the design of fuzzy controllers using evolutionary algorithm approaches. GA could be used for developing the knowledge base about the controlled process in the form of linguistic rules and the fine-tuning of fuzzy membership function [18].

A possible solution to a specific problem can be encoded as an individual (or a chromosome), which consists of group of genes. Each individual represents a point in the search space and a possible solution to the problem can be formulated. A population consists of a finite number of individuals and each individual is decided by an evaluating mechanism to obtain its fitness value. Using this fitness value and genetic operators, a new population is generated iteratively which is referred to as a generation. The GA uses the basic reproduction operators such as crossover and mutation to produce the genetic composition of a population. The crossover operator produces two offsprings (new candidate solutions) by recombining the information from two parents. As mutation operation is a random alteration of some gene values in an individual, the allele of each gene is a candidate for mutation, and its function is determined by the mutation probability. Many efforts for the enhancement of traditional GAs have been proposed [21–23]. Among them, one category focuses on modifying the structure of the population or on the individual's role [15]. Some examples are distributed GA [15], cellular GA [19] and symbiotic GA. Another category is focused on modification/efficient control of the basic operations, such as crossover or mutation, of traditional GAs [19].

On the other hand, as natural selection tends to eliminate animals with poor foraging strategies through methods for locating, handling, and ingesting food and favor the propagation of genes of those animals that have successful foraging strategies, they are more likely to apply reproductive success to have an optimal solution [25,26]. After many generations, poor foraging strategies are either eliminated or shaped into good ones. Logically, such evolutionary principles have led scientists in the field of foraging theory to hypothesize that it is appropriate to model the activity of foraging as an optimization process. Since a foraging animal takes actions to maximize the energy obtained per unit time spent foraging, in the face of constraints presented by its own physiology, such as sensing and cognitive capabilities and environment (e.g., density of prey, risks from predators, physical

[*] Tel.: +82 42 821 1170; fax: +82 821 1164.
E-mail address: kimdh@hanbat.ac.kr.
URL: http://www.aialab.net

1568-4946/$ – see front matter © 2010 Published by Elsevier B.V.
doi:10.1016/j.asoc.2009.01.004

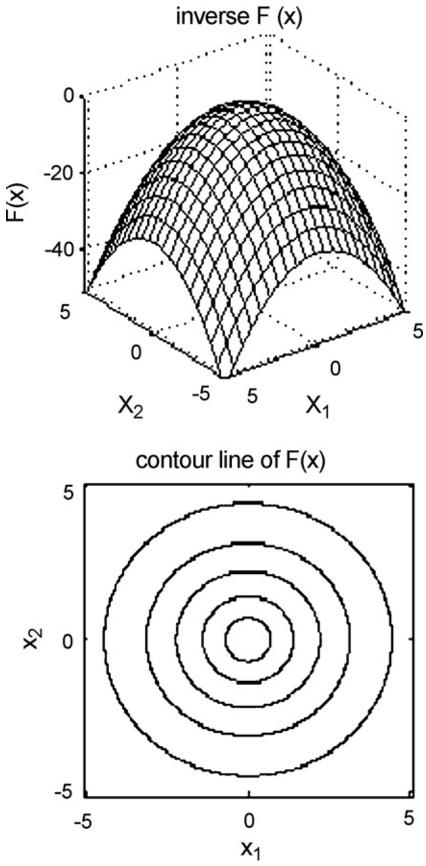

Fig. 1. Contour of test function f_1.

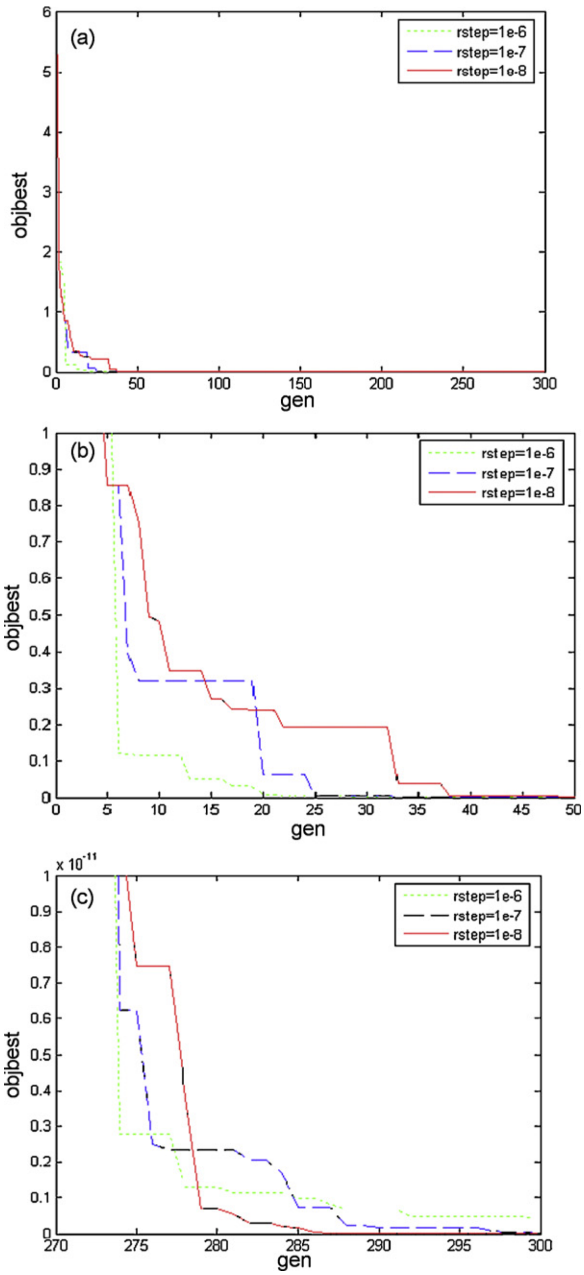

Fig. 2. (a) Characteristic of variables to variation of step size (generations = 1–30). (b) Characteristic of variables to variation of step size (generations = 1–50). (c) Characteristic of variables to variation of step size (generations = 270–300).

characteristics of the search area), evolution can provide optimization within these constraints and essentially apply to engineering field by what is sometimes referring to as an optimal foraging policy. That is, optimization models can provide for social foraging where groups of parameters communicate to cooperatively forage in engineering.

As the hybrid system, Refs. [32,33] suggest optimal solution, In Ref. [32], they use hybrid least square bacterial strategy, Ref. [33] GA–BF strategy for tuning including experiments.

First, this paper provides a brief literature overview of the area of Bacterial Foraging as it forms the biological foundation for this paper. Then, this paper also focuses on dealing with an enhanced optimal solution using a hybrid approach consisting of BA (Bacterial Foraging) and GA (Genetic Algorithm). Finally, we focus on evidence for the proposed hybrid system by simulating through various test functions.

2. Hybrid system consisting of GA and Bacteria Foraging

As mentioned above, foraging behavior of bacteria [25,26] can be found using, for instance, dynamic programming. Search and optimal foraging decision-making of animals can be used to engineering. To perform social foraging an animal needs communication capabilities, and it can gain advantages in that it can exploit essentially the sensing capabilities of the group, the group can gang-up on large prey, individuals can obtain protection from predators while in a group, and in a certain sense the group can forage with a type of collective intelligence.

2.1. Overview of chemotactic behavior of Escherichia coli

This paper considers the foraging behavior of *E. coli*, which is a common type of bacteria as mentioned in the previous comment [25,26]. Its behavior to move comes from a set of up to six rigid

Table 1
Parameter values to step size of bacteria.

Step size	*x*1	*x*2	*x*3	Optimal objective function	Average objective function
1.0E−5	3.87E−13	6.60E−13	2.92E−07	−5.43E−07	−8.98E−08
1.0E−6	2.85E−14	2.34E−13	−5.52E−08	1.50E−07	−5.45E−08
1.0E−7	5.01E−16	1.43E−15	−1.70E−08	−1.44E−08	−2.31E−09

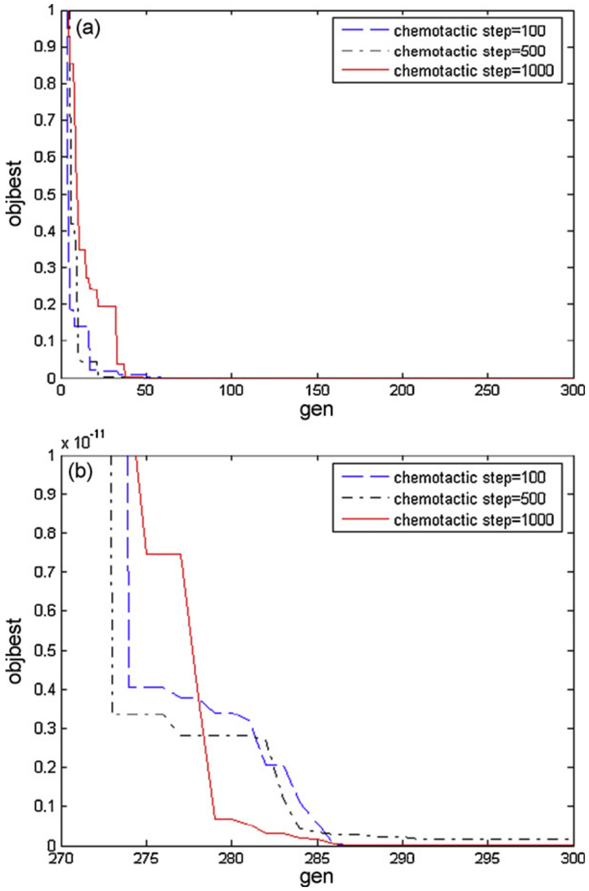

Fig. 3. (a) Relationship between objective function and generations in different chemotactic steps (generations = 0–300). (b) Relationship between objective function and generations in different chemotactic steps (generations = 270–300).

100–200 rps spinning flagella, each driven as a biological motor. An *E. coli* bacterium alternates through running and tumbling. Running speed is 10–20 μm/s, but it cannot swim straight.

For this paper, the chemotactic actions of bacteria can be summarized with the following descriptions:

- If in neutral medium, alternate tumbles and runs, its action has search.
- If it swims up a nutrient gradient (or out of noxious substances), swims longer (climbs up nutrient gradient or down noxious gradient), its behavior seeks increasingly favorable environments.
- If it swims down a nutrient gradient (or up noxious substance gradient), then search action avoids unfavorable environments.

So, it can climb up nutrient hills and at the same time avoid noxious substances. The sensors it needs for optimal resolution are receptor proteins which are very sensitive and high gain. That is, a small change in the concentration of nutrients can cause a significant change in behavior. This is probably the best-understood sensory and decision-making system in biology.

Mutations in *E. coli* affect the reproductive efficiency at different temperatures, and occur at a rate of about 10^{-7} per gene and per generation. *E. coli* occasionally engages in a conjugation that affects the characteristics of a population of bacteria. Since there are many types of taxes that are used in bacteria, such as aerotaxis attraction to oxygen, phototaxis by light, thermotaxis by temperature, magnetotaxis, bacteria can be affected by magnetic lines of flux and some bacteria can change their shape and number of flagella which is based on the medium to reconfigure in order to ensure efficient foraging in a variety of media. Bacteria can form intricate stable spatio-temporal patterns in certain semisolid nutrient substances and they can eat radially their way through a medium if placed together initially at its center. Moreover, under certain conditions, they will secrete cell-to-cell attractant signals so that they will group and protect one another. These bacteria can swarm.

2.2. Optimization function of bacterial swarm foraging for the hybrid system GA–BF

The main goal of the hybrid system GA–BF based on Bacterial Foraging is to apply in order to find the minimum of $P(\phi)$, $\phi \in R^n$, not in the gradient $\bigtriangledown P(\phi)$. Here, when ϕ is the position of a bacterium, and $J(\phi)$ is an attractant–repellant profile. That is, it means where nutrients and noxious substances are located, so $P < 0$, $P = 0$, $P > 0$ represent the presence of nutrients. A neutral medium, and the presence of noxious substances, respectively can be defined by

$$H(j,k,l) = \{\phi^i(j,k,l) | i = 1, 2, \ldots, N\}. \tag{1}$$

Equation represents the positions of each member in the population of the N bacteria at the jth chemotactic step, kth reproduction step, and lth elimination-dispersal event. Let $P(i, j, k, l)$ denote the cost at the location of the ith bacterium $\phi^i(j, k, l) \in R^n$, and

$$\phi^i = (j+1,k,l) = \phi^i(j,k,l) + C(i)\varphi(j), \tag{2}$$

so that $C(i) > 0$ is the size of the step taken in the random direction specified by the tumble. If at ϕ^i $(j + 1, k, l)$ the cost $J(i, j + 1, k, l)$ is better (lower than) at ϕ^i (j, k, l), then another chemotactic step of size $C(i)$ in this same direction will be taken and repeated up to a maximum number of steps *Ns*. *Ns* is the length of the lifetime of the bacteria measured by the number of chemotactic steps. Functions $P_c^i(\phi)$, $i = 1, 2, \ldots, S$, to model the cell-to-cell signaling via an attractant and a repellant are represented by [19,20,25,26]

$$\begin{aligned} P_c(\phi) &= \sum_{i=1}^{N} P_{cc}^i \\ &= \sum_{i=1}^{N}\left[-L_{attract}\exp\left(-\delta_{attract}\sum_{j=1}^{n}(\phi_j-\phi_j^i)^2\right)\right] \\ &\quad+\sum_{i=1}^{N}\left[-L_{re\,pellant}\exp\left(-\delta_{attract}\sum_{j=1}^{n}(\phi_j-\phi_j^i)^2\right)\right], \end{aligned} \tag{3}$$

where $\phi = [\phi_1,\ldots,\phi_p]^T$ is a point on the optimization domain, $L_{attract}$ is the depth of the attractant released by the cell and $\delta_{attract}$ is a measure of the width of the attractant signal. $K_{repellant} = L_{attract}$ is the height of the repellant effect magnitude, and $\delta_{attract}$ is a

Table 2
Variation of parameter values to chemotactic step.

Chemotactic step	*x*1	*x*2	*x*3	Optimal objective function	Average objective function
100	−9.32E−08	3.78E−07	−8.57E−09	1.52E−13	1.59E−13
500	2.97E−08	1.92E−08	2.32E−08	1.79E−15	3.26E−15
1000	−1.70E−08	−1.44E−08	−2.31E−09	5.01E−16	1.43E−15

Table 3
Initial condition of test function and variation of parameters obtained by simulation.

Test function	Range		Genetic Algorithm parameters				Bacteria Foraging parameters			
	$x_i^{(L)}$	$x_i^{(U)}$		G	Mu	Cr	CS	Step size	Ns	S
$f_1(x)=\sum_{i=1}^{3} x_i^2$	−5.12	5.11	20	300	0.9	0.1	1000	1E−007	3	10
$f_2(x)=100(x_1^2-x_2)^2+(1-x_1)^2$	−2.048	2.047	20	600	0.9	0.1	1000	1E−007	3	10
$f_3=\sum_{i=1}^{5}[x_i]$	−5.12	5.12	20	180	0.9	0.1	1000	1E−007	3	10
$f_4=\sum_{i=1}^{30} ix_i^4+N(0,1)$	−1.28	1.27	20	300	0.9	0.1	1000	1E−007	3	10

measure of the width of the repellant. The expression of $P_c(\phi)$ means that its value does not depend on the nutrient concentration at position ϕ. That is, a bacterium with high nutrient concentration secretes stronger attractant than one with low nutrient concentration. Models use the function $P_{ar}(\phi)$ to represent the environment-dependent cell-to-cell signaling as

$$P_{ar}(\phi)=\exp(T-P(\phi))P_c(\phi) \tag{4}$$

where T is a tunable parameter. By considering minimization of $P(i, j, k, l)+P_{ar}(\phi^i(j, k, l))$, the cells try to find nutrients, avoid noxious substances, and at the same time try to move toward other cells, but not too close to them. The function $P_{ar}(\phi^i(j, k, l))$ implies that, with M being constant, the $P(\phi)$ is smaller, $P_{ar}(\phi)$ is larger and thus the stronger attraction, which is intuitively reasonable. In tuning the parameter M, it is normally found that, when M is very large, $P_{ar}(\phi)$ is much larger than $J(\phi)$, and thus the profile of the search space is dominated by the chemical attractant secreted by *E. coli*.

On the other hand, if T is very small, then $P_{ar}(\phi)$ is much smaller than $P(\phi)$, and it is the effect of the nutrients that dominates. In $P_{ar}(\phi)$, the scaling factor of $P_c(\phi)$ is given as in exponential form.

This paper describes the method in the form of an algorithm to search optimal value of parameters.

[step 1] Initialize parameters $n, N, N_C, Ns, N_{re}, N_{ed}, P_{ed}, C(i)\ (i=1, 2, \ldots, N), \phi^i$.

where,

n: dimension of the search space,
N: the number of bacteria in the population,
N_C: chemotactic steps,
N_{re}: the number of reproduction steps,
N_{ed}: the number of elimination-dispersal events,
P_{ed}: elimination-dispersal with probability,
$C(i)$: the size of the step taken in the random direction specified by the tumble.

[step 2] Elimination-dispersal loop: $l=l+1$

[step 3] Reproduction loop: $k=k+1$

[step 4] Chemotaxis loop: $j=j+1$

[substep a] For $i=1, 2, \ldots, N$, take a chemotactic step for bacterium i as follows.

[substep b] Compute fitness function, ITSE (i, j, k, l).

[substep c] Let $\text{ITSE}_{last}=\text{ITSE}\ (i, j, k, l)$ to save this value since we may find a better cost via a run.

[substep d] Tumble: generate a random vector $\Delta(i)\in R^n$ with each element $\Delta_m(i), m=1, 2, \ldots, p$, a random number on $[-1, 1]$.

[substep e] Move: let

$$\phi^i=(j+1, k, l)=\phi^i(j, k, l)+C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

This results in a step of size $C(i)$ in the direction of the tumble for bacterium i.

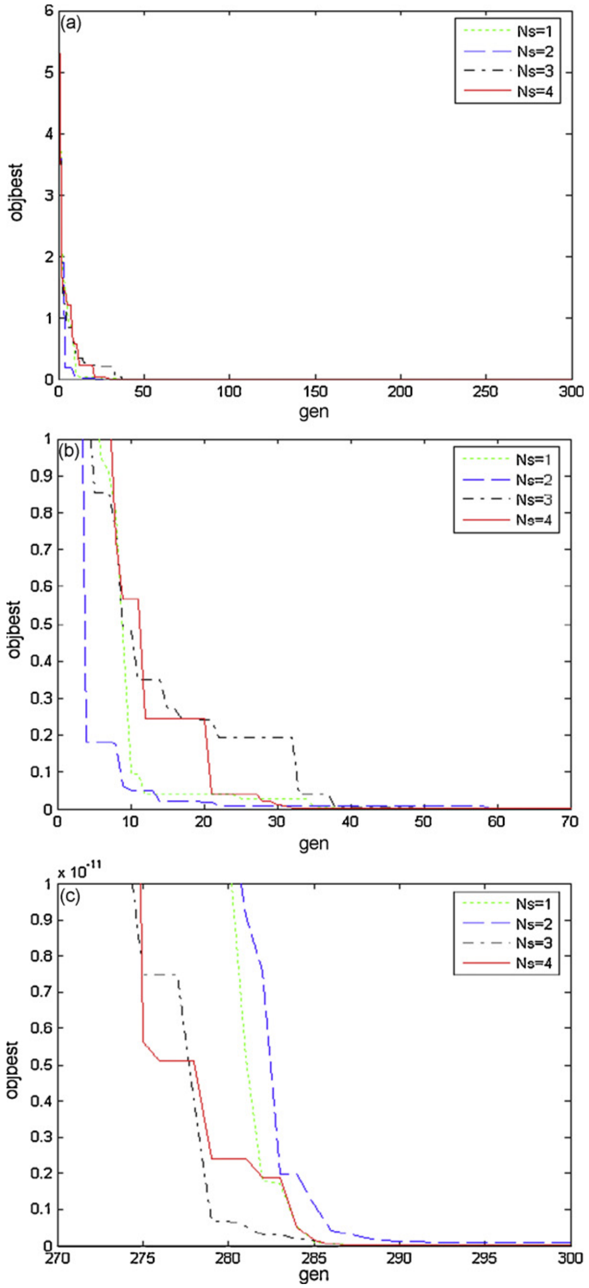

Fig. 4. (a) Characteristic between objective function and generators to different lifetime *Ns* of bacteria in the GA–BF (generators = 0–300). (b) Characteristic between objective function and generators to different lifetime *Ns* of bacteria in the GA–BF (generations = 1–70). (c) Characteristic between objective function and generators to different lifetime *Ns* of bacteria in the GA–BF (generations = 270–300).

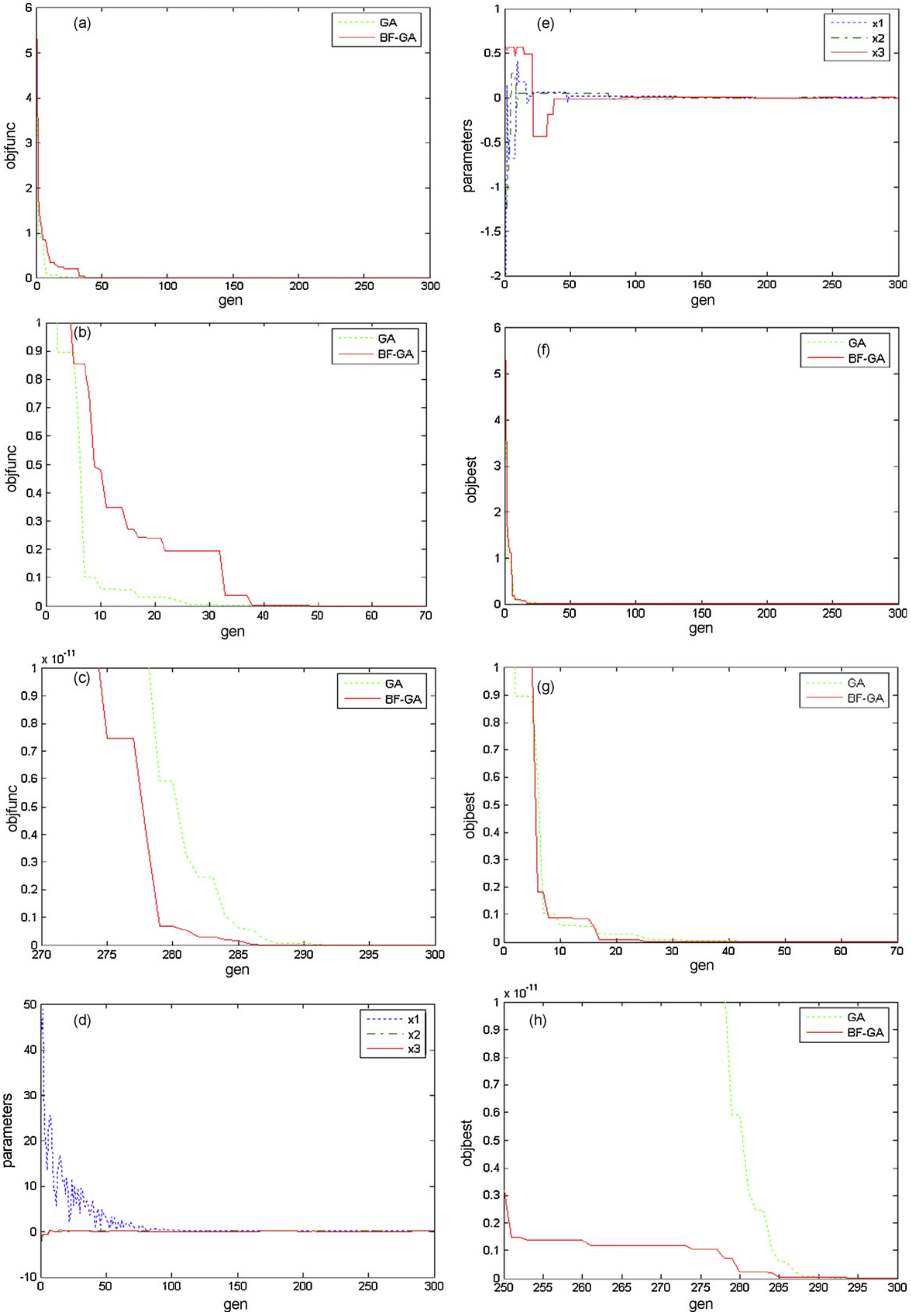

Fig. 5. (a) Characteristics of GA and GA–BF by test function f_1 (generators = 0–300). (b) Characteristics of GA and GA–BF by test function. f_1 (generators = 1–70). (c) Characteristics of GA and GA–BF by test function f_1 (generations = 270–300). (d) Process of optimal search in GA using test function f_1. (e) Process of optimal search in GA–BF using test function f_1. (f) Characteristic of GA and GA by test function f_1 BF (step size = 1×10^{-5}, generations = 0–300). (g) Characteristic of GA and GA–BF by test function f_1 (step size = 1×10^{-5}, generation = 1–70). (h) Characteristic of GA and GA–BF by test function f_1 (step size = 1×10^{-5}, generation: 270–300). (i) Process for optimal solution on GA and GA–BF using test function f_1 (step size = 1×10^{-5}).

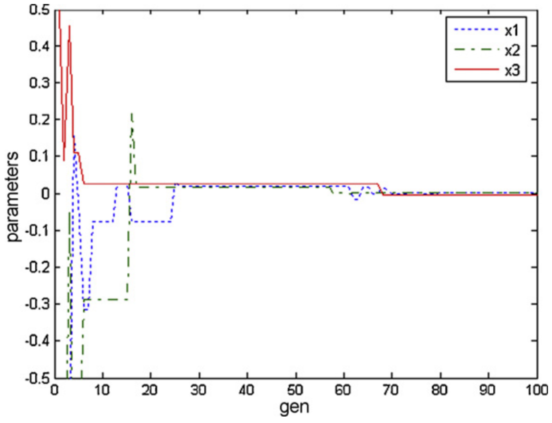

Fig. 5. (*Continued*).

[substep f] Compute ITSE ($i, j+1, k, l$).

[substep g] Swim

(i) Let $m = 0$ (counter for swim length),

(ii)While $m < N_s$ (if have not climbed down too long).

- Let $m = m + 1$.
- If ITSE ($i, j+1, k, l$) < ITSE$_{last}$ (if doing better), let ITSE$_{last}$ = ITSE ($i, j+1, k, l$) and let

$$\phi^i(j+1,k,l) = \phi^i(j+1,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

and use this ϕ^i ($j+1, k, l$) to compute the new ITSE ($i, j+1, k, l$) as we did in [substep f]

- Else, let $m = N_s$. This is the end of the while statement.

[substep h] Go to next bacterium (i, 1) if $i \neq N$ (i.e., go to [substep b] to process the next bacterium),

[step 5] If $j < N_C$, go to step 3. In this case, continue chemotaxis, since the life of the bacteria is not over.

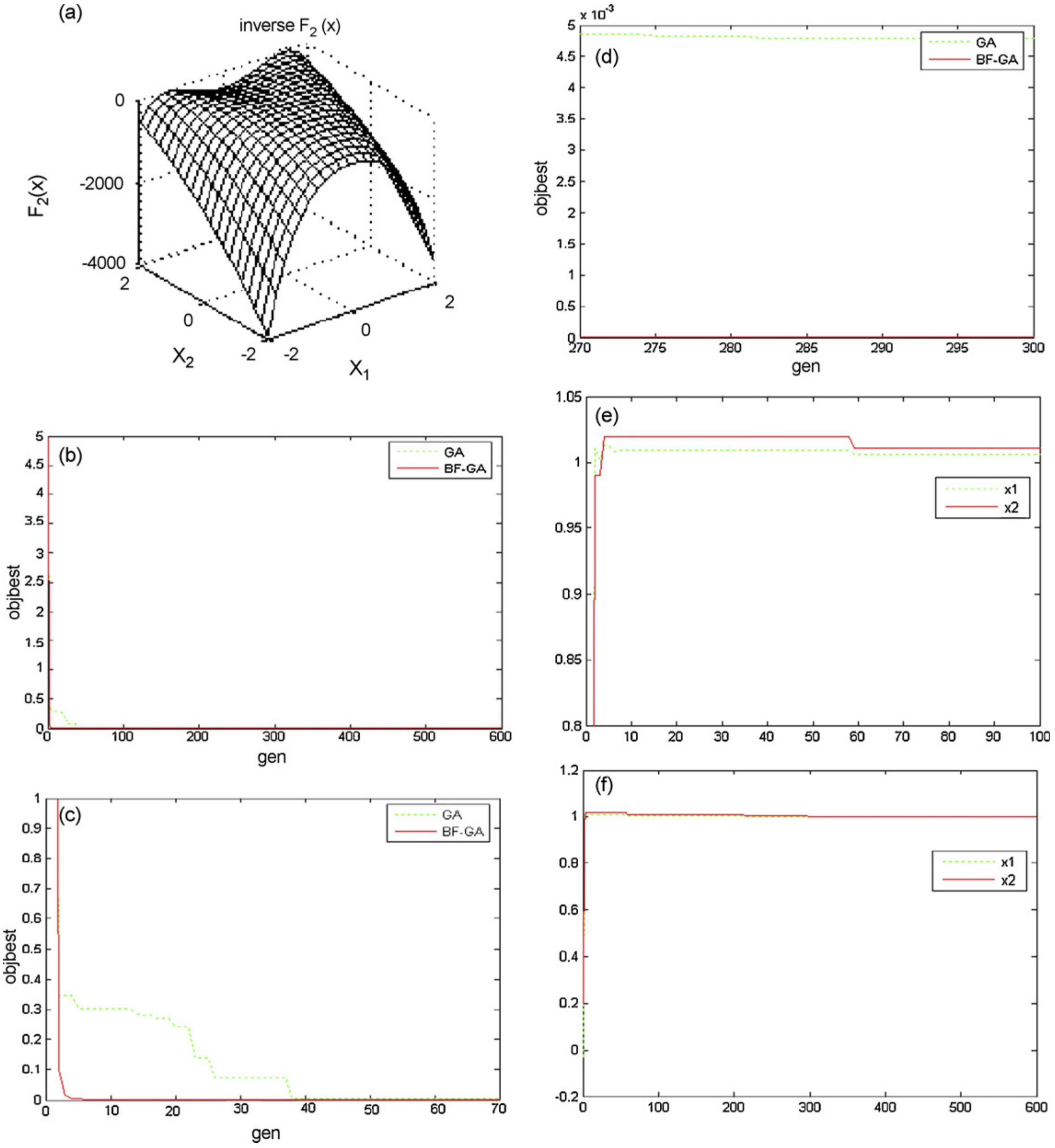

Fig. 6. (a) Contour of test function (f_2) at $x = [1\ 1]^T$. (b) Characteristic of GA and GA–BF by test function f_2 (generations = 0–600). (c) Characteristic of GA and GA–BF by test function f_2 (generations = 1–70). (d) Characteristic of GA and GA–BF by test function f_2 (generation: 270–300). (e) Optimal process of GA by test function, f_2. (f) Optimal process of GA–BF by test function, f_2.

Table 4
Parameter variation of GA and GA–BF by test function f_1.

Chemotatic step	$x1$	$x2$	$x3$	Optimal objective function	Average objective function
GA	7.22E−08	5.07E−08	−9.43E−09	7.87E−15	8.03E−15
GA–BF	−1.70E−08	−1.44E−08	−2.31E−09	5.01E−16	1.43E−15

Table 5
Parameter variation of GA and GA–BF by test function f_2.

Chemotactic step	$x1$	$x2$	Optimal objective value	Average objective value
GA	0.001967	0.001967	1.0443267	1.0907699
BF–GA	5.12E−09	5.17E−09	0.9999285	0.9998567

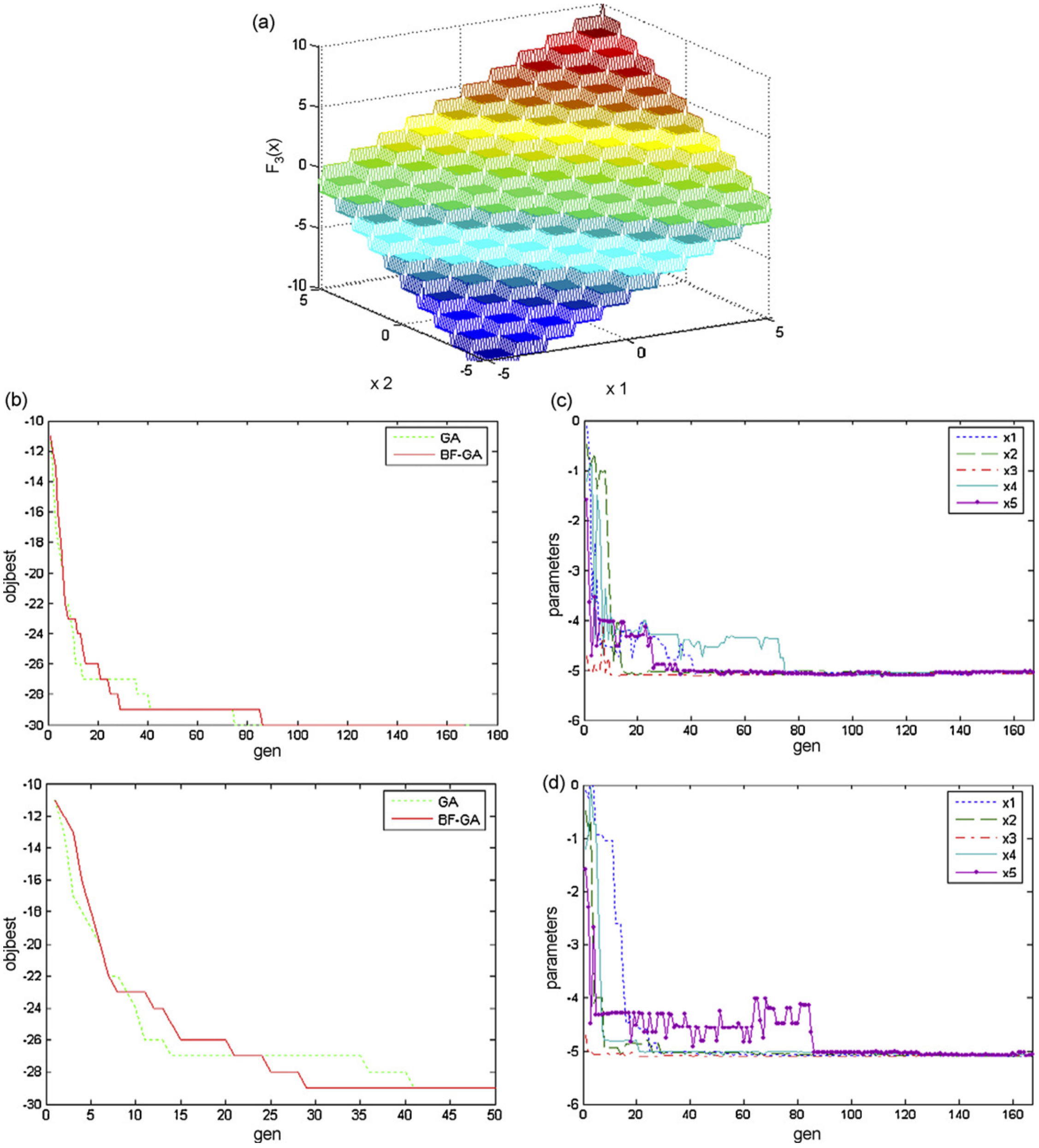

Fig. 7. (a) Contour map of test function f_3. (a) Characteristic of GA and GA–BF by test function f_3. (b) Characteristic of GA and GA–BF by test function f_3 (generations = 1–70). (c) Optimal process of GA on test function f_3. (d) Optimal process of GA–BF on test function f_3.

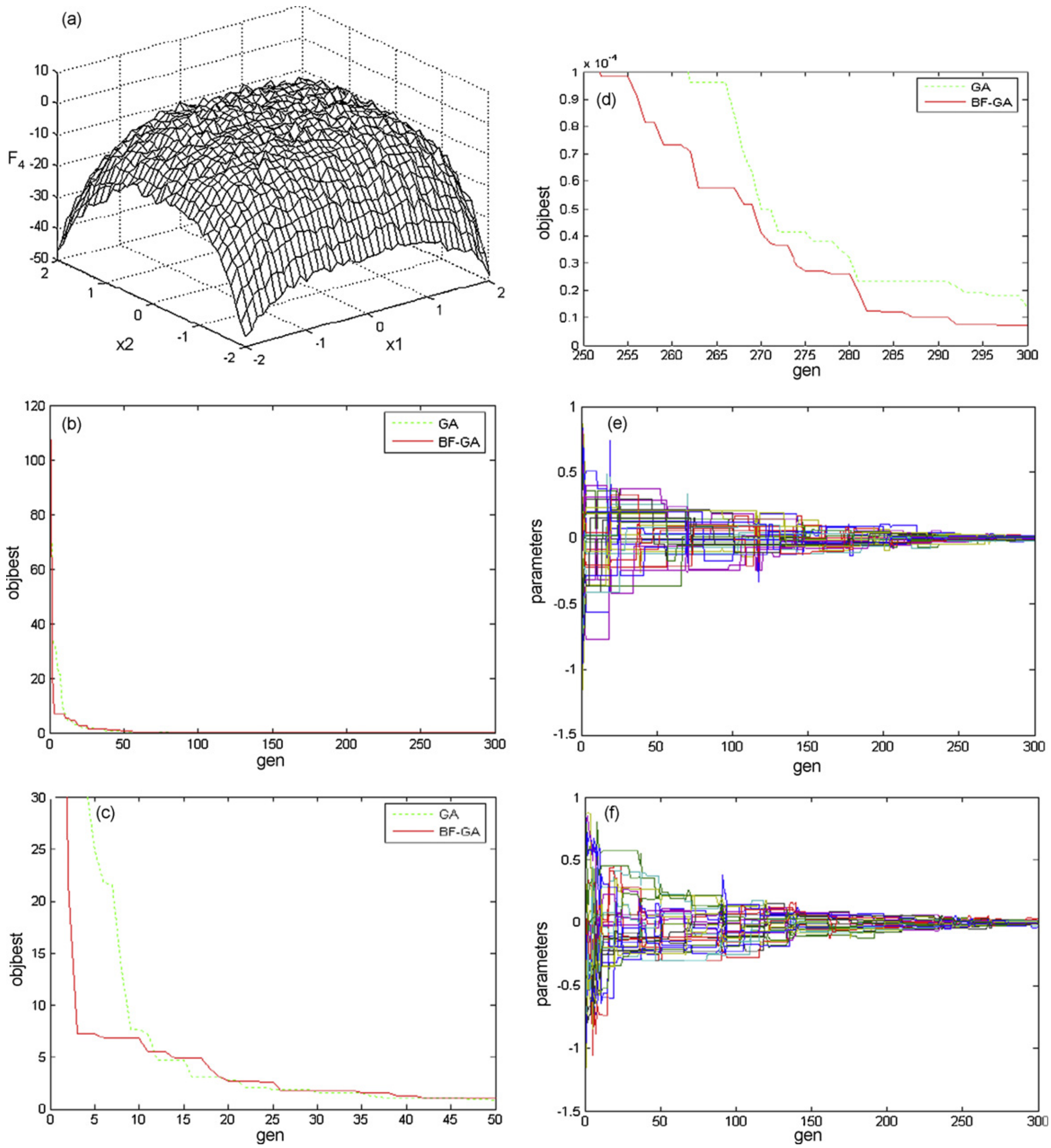

Fig. 8. (a) Contour map of test function f_4. (b) Characteristic of GA and GA–BF by test function f_4. (c) Characteristic of GA and GA–BF by test function f_4 (generations = 1–70). (d) Characteristic of GA and GA–BF by test function f_4 (generations = 270–300). (e) Optimal process of GA on test function f_4. (f) Optimal process of GA–BF on test function f_4.

[step 6] Reproduction:

[substep a] For the given k and l, and for each $i = 1, 2, \ldots, N$, let

$$\mathrm{ITSE}^{i}_{health} = \sum_{j=1}^{N_C+1} \mathrm{ITSE}(i, j, k, l)$$

be the health of bacterium i (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters $C(i)$ in order of ascending cost ITSE_{health} (higher cost means lower health).

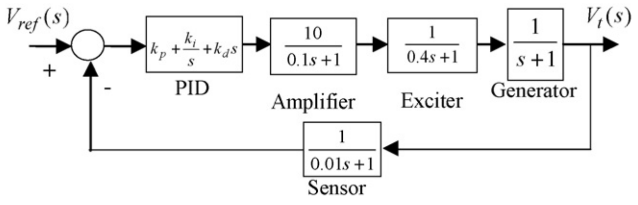

Fig. 9. Block diagram of an AVR system with a PID controller.

[substep b] The S_r bacteria with the highest ITSE_{health} values die and the other S_r bacteria split with the best values (this process is performed by the copies that are made are placed at the same location as their parent).

[step 7] If $k < N_{re}$, go to [step 3]. In this case, we have not reached the number of specified reproduction steps, so we start the next generation in the chemotactic loop.

[step 8] Elimination-dispersal: for $i = 1, 2, \ldots, N$, with probability P_{ed}, eliminate and disperse each bacterium, and this results in keeping the number of bacteria in the population constant.

To do this, if you eliminate a bacterium, simply disperse one to a random location on the optimization domain. If $l < N_{ed}$, then go to [step 2]; otherwise end.

Table 6
Parameter variation of GA and GA–BF by test function f_3.

Method	*x*1	*x*2	*x*3	*x*4	*x*5	Optimal objective value	Average objective value
GA	−5.024811	−5.015523	−5.059941	−5.03529	−5.03527	−30	−29.4
BF–GA	−5.111186	−5.097807	−5.089435	−5.06529	−5.06891	−30	−29.95

Table 7
Parameter variation of GA and GA–BF by test function f_4.

Method	*x*1	*x*2	*x*3	*x*4	*x*5	Optimal objective value	Average objective value
GA	−5.024811	−5.015523	−5.059941	−5.03529	−5.03527	−30	−29.4
BF–GA	−5.111186	−5.097807	−5.089435	−5.06529	−5.06891	−30	−29.95

Table 8
Range of PID parameters for learning of GA–BF.

PID parameters	Range	
	Min	Max
k_p	0	1.5
k_i	0	1
k_d	0	1

Table 9
Simulation parameters of BF–GA.

Parameters	Values
Step size	0.08
Ns	4
P_c	0.9
P_m	0.65

3. Simulation for comparison of the conventional GA and the hybrid GA–BF

This paper illustrates characteristics between the proposed GA–BF (Genetic Algorithms–Bacteria Foraging) and the conventional SGA (Simple Genetic Algorithm) using test function, De Jong [31].

3.1. Mutation operation in GA–BF

For illustrating the characteristic of mutation in GA–BF dynamic mutation is used [31]. Genetic, x_j of calculator based on this mutation is defined as

$$x_j = \begin{cases} \tilde{x}_j + \Delta(k, x_j^{(U)} - \tilde{x}_j), & \tau = 0, \\ \tilde{x}_j - \Delta(k, \tilde{x}_j - x_j^{(L)}), & \tau = 1. \end{cases} \tag{4'}$$

where, random constant, τ becomes 0 or 1 and $\Delta(k, y)$ is given as

$$\Delta(k, y) = y \cdot \eta \cdot \left(1 - \frac{k}{T}\right)^A. \tag{5}$$

Here, η has 0 or 1 randomly and T is maximum generation for computation. A is defined by user.

3.2. Crossover operation in GA–BF

In this paper, modified simple crossover is used for representing the characteristic of BF–GA [31]. Chromosome in this approach is calculated as follows:

$$\tilde{x}_j^u = \lambda \bar{x}_j^v + (1 - \lambda)\bar{x}_j^u \tag{6a}$$

$$\tilde{x}_j^v = \lambda \tilde{x}_j^u + (1 - \lambda)\tilde{x}_j^v \tag{6b}$$

where, $\bar{x}_j^u$, $\bar{x}_j^v$ mean parent's generations and $\tilde{x}_j^u$, $\tilde{x}_j^v$ mean offspring's generations. j is chromosome of jth and λ is multiplier.

3.3. Performance to variation of step size

For comparison of performance to variation of step size, test function f_1 is used as follows:

$$f_1(x) = \sum_{i=1}^{3} x_i^2. \tag{7}$$

Range for test is given as $-5.12 \le x_1, x_2, x_3\ 5.11 \le$. Step size means moving distance per step of bacteria (Fig. 1).

Fig. 2(a) shows the characteristic to variation of step size and Fig. 2(b) and (c) depicts characteristic to variation of step size on generations from 1 to 50 and from 270 to 300, respectively. From Fig. 2, the bigger the step size, the fast the convergence (Table 1).

Table 10
Best solution using BF–GA controller with the different β values.

β	Number of generation	k_p	k_i	k_d	*Mo* (%)	*ess*	t_s	t_r	Evaluation value
0.5	200	0.68233	0.6138	0.26782	1.94	0.0171	0.3770	0.2522	0.3614
1	200	0.68002	0.52212	0.24401	1.97	0.0067	0.4010	0.2684	0.1487
1.5	200	0.67278	0.47869	0.22987	1.97	0.0014	0.4180	0.2795	0.07562

Table 11
Comparison of the evaluation value between both methods (β = 1.5, generation = 200).

β	Number of generation	k_p	k_i	k_d	*Mo* (%)	*ess*	t_s	t_r	Evaluation value
1.5	GA	0.8282	0.7143	0.3010	6.7122	0.0112	0.5950	0.2156	0.0135
	PSO	0.6445	0.5043	0.2348	0.8399	0.0084	0.4300	0.2827	0.0073
	GA–PSO	0.6794	0.6167	0.2681	1.8540	0.0178	0.8000	0.2526	0.0071
	BF–GA	0.6728	0.4787	0.2299	1.97	0.0014	0.4180	0.2795	0.0756

3.4. Characteristic to chemotactic step ofGA–BF

Fig. 3 and Table 2 show relationship between objective function and the number of generations in different chemotactic steps. The chemotactic step is smaller, the objective function is faster than convergence with small generator. Fig. 4 shows characteristics between objective function and generators to different lifetime *Ns* of bacteria in the hybrid system, GA–BF. Table 3 shows initial condition of test function and variation of parameters obtained by simulation, and parameter values are obtained using range (searching range), chemotactic step, total number of chemotactic reactions of bacteria, step size, basic unit for the movement of bacteria, *Ns*: the number of critical reaction, *S*: the number of bacteria, *G*: generations, *Mu*: mutation, and *Cr*: crossover.

3.5. Characteristic to lifetime of GA–BF

Fig. 4.

3.5.1. Characteristics of GA–BF to test functions

(1) Test function: f_1

Test function $f_1(x) = \sum_{i=1}^{3} x_i^2$ mentioned above (Fig. 1) is used for comparing the characteristic of the conventional GA and the proposed hybrid system GA–BF.

Fig. 5(a) shows characteristics of GA and GA–BF on total generations by this test function and Fig. 5(b) shows characteristics of GA and GA–BF on generations from 1 to 70. Fig. 5(c) represents characteristics of GA and GA–BF on generations from 270 to 300. Fig. 5(b) shows that the GA can search optimal solution earlier as 10 generations as than the hybrid system GA–BF at the beginning step, but Fig. 5(c) represents that the GA–BF is earlier than GA at final step.

Fig. 6(d) and (e) show the process of optimal search to test function f_1 in GA and GA–BF, respectively. Fig. 6(f), Fig. 5(g), Fig. 5(h), and Fig. 5(i) illustrate characteristic of GA and GA–BF with different step size (step size = 1 × 10^{-5}) by test function, f_1. Table 4 shows parameters obtained by simulation to test function f_1.

(2) Test function: f_2

Test function $f_2(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$ is used for the comparison of performance of GA and GA–FA. Fig. 6(a) shows contour of this function at $x = [1\ 1]^T$. Fig. 6(b)–(f) represent characteristics of the conventional GA and the proposed hybrid system GA–BF by test function f_2 (generation: 270–300), as the previous test function f_1.

From these figures, GA–BF can have solution earlier than GA at all generations (initial generations and final generations) and we can find that it is possible for the GA–BF to search more satisfactory solution. Table 5 shows parameter values of GA and GA–BF obtained by test function f_2

(3) Test function: f_3

Test function $f_3 = \sum_{i=1}^{5} [x_i]$ is applied to compare performance to GA and GA–BF.

This function has minimum −30 at x = [−5.12, −5.12, −5.12, −5.12, −5.12,]. Fig. 7(a) is contour map of this function and Fig. 7(b)–(d) represent the result simulated by this test function as the test function f_1, f_2.

(4) Test function: f_4

$f_4 = \sum_{i=1}^{30} ix_i^4 + N(0,1)$ is used to compare the conventional GA and the proposed system GA–BF and Fig. 8(a) shows contour map of this function. In Fig. 8(b)–(c), GA and GA–BF represent the similar response but Fig. 8(d) shows the proposed system is the faster response to the conventional GA.

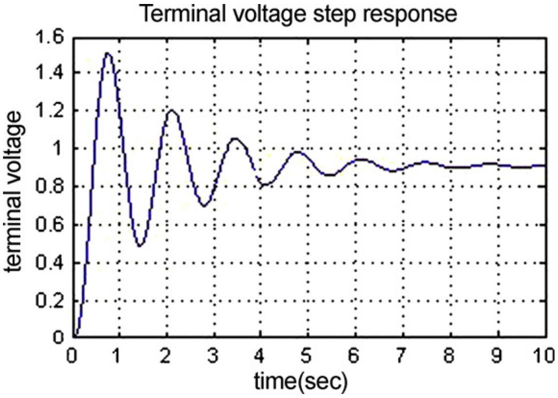

Fig. 10. Step response of terminal voltage in an AVR system without controller.

4. Intelligent tuning of PID controller for AVR using hybrid system GA–BF

The transfer function of PID controller in the AVR system is given by

$$G(s) = k_p + \frac{k_i}{s} + k_d s, \tag{8}$$

and block diagram of the AVR system is shown in Fig. 9. The performance index of control response is defined by

$$\begin{aligned}\min F(k_p, k_i, d_d) &= \frac{e^{-\beta} \cdot t_s/\max(t)}{(1-e^{-\beta}) \cdot |1 - t_r/\max(t)|} + e^{-\beta} \cdot Mo + ess \\ &= \frac{e^{-\beta} \cdot (t_s + \alpha_2 \cdot |1 - t_r/\max(t) \cdot Mo|)}{(1-e^{-\beta}) \cdot |1 - t_r/\max(t)|} \\ &= \frac{e^{-\beta} \cdot (t_s/\max(t) + \alpha \cdot Mo)}{\alpha} + ess\end{aligned} \tag{9}$$

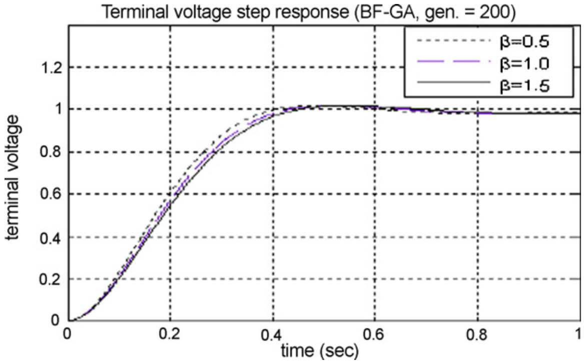

Fig. 11. Terminal voltage step response of an AVR system by BF–GA.

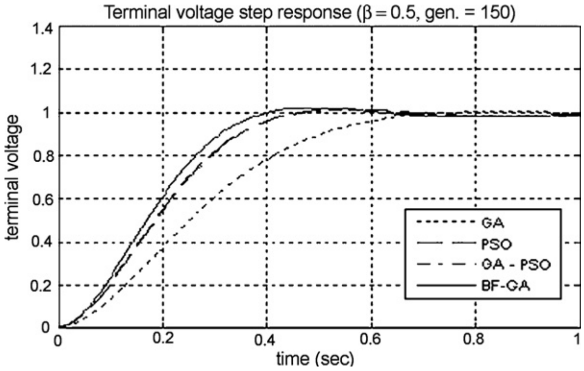

Fig. 12. Terminal voltage step response of an AVR system with different controllers (β = 0.5, generations = 200).

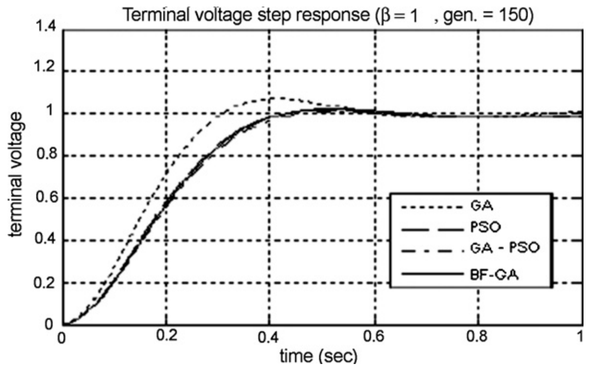

Fig. 13. Terminal voltage step response of an AVR system with different controllers (β = 1.0, generations = 200).

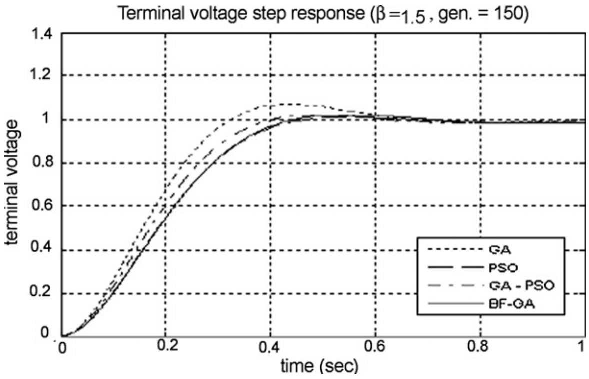

Fig. 14. Terminal voltage step response of an AVR system with different controllers (β = 1.5, generations = 200).

$\alpha = (1 - e^{-\beta})\cdot|1 - t_r/\max\,(t)|, k_p,\ k_i,\ k_d$: parameter of PID controller, β, weighting factor; *Mo*: overshoot, t_s: settling time (2%), *ess:* steady-state error, t: desired settling time.

In Eq. (9), if the weighting factor, β increases, rising time of response curve is small, and when β decreases, rising time is big. Performance criterion is defined as Mo = 50.61%, ess = 0.0909, t_r = 0.2693 (s) t_s = 6.9834 (s) (Tables 6 and 7).

Initial values of PID controller and GA–BF for simulation are shown as Tables 8 and 9, respectively.

Fig. 10 is the response of terminal voltage to step input when there is controller in total control system. Figs. 11–14 represent results obtained by GA, PSO, GA–PSO, and GA–BF (BF–GA) to variation of β at generations 200 in Eq. (9). Results are satisfactory. Figs. 15–17 illustrate, in an AVR system shown in Fig. 9, search process for optimal parameters to variation of β (=0.5, 1.0, and 1.5) by BF–GA. Table 10 shows the best solution using BF–GA controller with different β values and Table 11 is comparison of the values evaluated between both methods (β = 1.5, generation = 200).

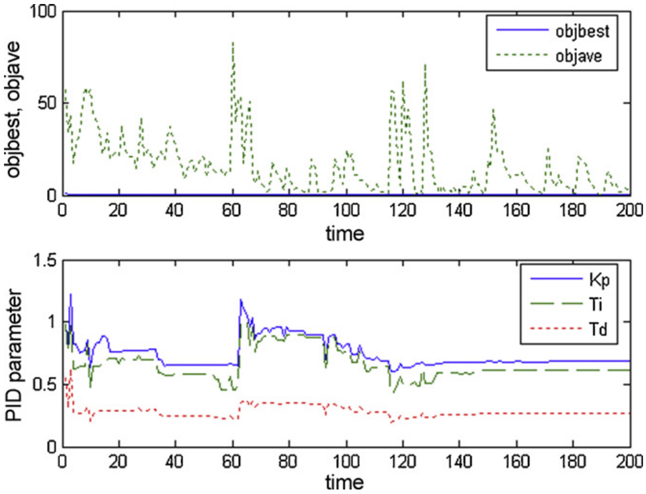

Fig. 15. Search process for optimal parameters in an AVR system by BF–GA (β = 0.5).

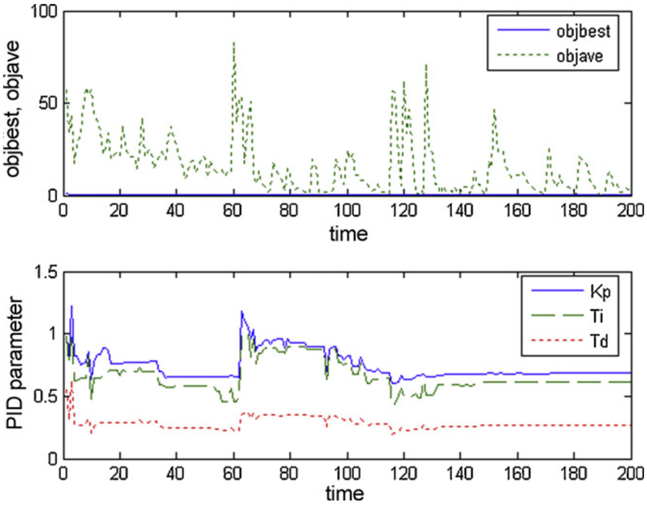

Fig. 16. Search process for optimal parameters in an AVR system by BF–GA (β = 1.0).

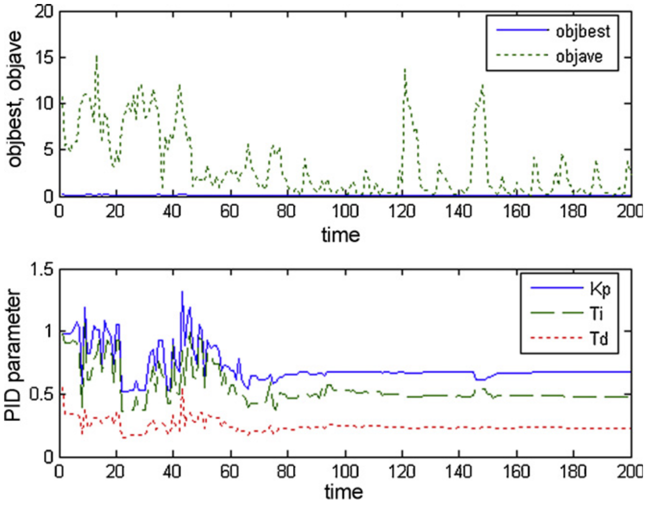

Fig. 17. Search process for optimal parameters in AVR system by BF–GA (β = 1.5).

5. Conclusion

Many recent approaches of evolutionary algorithms for the evaluation of improved learning algorithm and control engineering have been studied. The general problem of evolutionary algorithm based engineering system design has been tackled in various ways because of learning time and local or suboptimal solution. GA has also been used to optimize nonlinear system strategies but it might be locally optimized. Among optimization, a large amount of research is focused on the design of fuzzy controllers using evolutionary algorithm approaches. GA could be used for developing the knowledge-based learning about the controlled process in the form of linguistic rules and the fine-tuning of fuzzy membership function. However, it may also have problem with local optimization or suboptimal solution.

This paper suggests the hybrid system consisting of GA (Genetic Algorithm) and BF (Bacterial Foraging) and proved the characteristic of that system using various test functions. Also, it is introduced into tuning for PID controller of AVR system and compared with GA, PSO, GA–PSO.

This approach proposed in this has the potential to be useful in practical optimization problems (e.g., engineering design, online distributed optimization in distributed computing and cooperative control) as models of social foraging are also distributed nongradient optimization methods. It can also be used in a wide variety of fruitful research directions and ways to improve the models (e.g., modeling more dynamics of cell motion). Moreover, other species of bacteria or biological based computing approach could be studied but it remains to be seen how practically useful the optimization algorithms are for engineering optimization problems, because they depend on the theoretical properties of the algorithm, theoretical and empirical comparisons to other methods, and extensive evaluation on many benchmark problems and real-world problems.

References

[1] C.L. Lin, H.-W. Su, Intelligent control theory in guidance and control system design: an overview, Proc. Natul. Sci., Counc. ROC(A) 24 (1) (2000) 15–30.

[2] P.J. Fleming, R.C. Purshouse, Evolutionary algorithms in control system engineering: a survey, Control Eng. Pract. 10 (2002) 1223–1241.

[3] M. Dotoli, G. Maione, D. Naso, E.B. Turchiano, Genetic identification of dynamical systems with static nonlinearities, in: Proceedings of the IEEE SMCia/01, Mountain Workshop Soft Computing Industrial Applications, Blacksburg, VA, June 25–27, (2001), pp. 65–70.

[4] G.J. Gray, D.J. Murray-Smith, Y. Li, K.C. Sharman, T. Weinbrenner, Nonlinear model structure identification using genetic programming, Control Eng. Pract. (6) (1998) 1341–1352.

[5] K. Kristinnson, G.A. Dumont, System identification and control using genetic algorithms, IEEE Trans. System, Man, Cybern. 22 (September–October) (1992) 1033–1046.

[6] B. Maione, D. Naso, B. Turchiano, GARA: a genetic algorithm with resolution adaptation for solving system identification problems, in: Proceedings of the European Control Conference, Porto, Portugal, September 4–7, (2001), pp. 3570–3575.

[7] C.M. Fonseca, P.J. Fleming, Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part I: a unified formulation;" "—Part II: application example", IEEE Trans. System, Man, Cybern. A: Syst. Humans 28 (January (1)) (1998) 26–47.

[12] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, New York, 1999.

[13] J. Arabas, Z. Michalewicz, J. Mulawka, GAVaPS—a genetic algorithm with varying population size, in: Proceedings of the IEEE International Conference on Evolutionary Computation, Orlando, (1994), pp. 73–78.

[14] R. Tanese, Distributed genetic algorithm, in: Proceedings of the International Conference on Genetic Algorithms, 1989, pp. 434–439.

[15] R.J. Collins, D.R. Jefferson, Selection in massively parallel genetic algorithms, in: Proc. Int. Conf. Genetic.

[16] S. Tsutsui, D.E. Goldberg, Simplex crossover and linkage identification: single-stage evolution vs. multi-stage evolution, in: Proceedings of the IEEE International Conference on Evolutionary Computation, HI, (2002), pp. 974–979.

[18] H. Yoshida, K. Kawata, Y. Fukuyama, A particle swarm optimization for reactive power and voltage control considering voltage security assessment, IEEE Trans. Power Syst. 15 (November) (2000) 1232–1239.

[19] C.-F. Juang, A hybrid of genetic algorithm and particle swarm optimization for recurrent network design, Systems, Man Cybern., Part B: IEEE Trans. 34 (April (2)) (2004) 997–1006.

[20] D.W. Stephens, J.R. Krebs, Foraging Theory, Princeton University Press, Princeton, NJ, 1986.

[21] J. Alcock, Animal Behavior: An Evolutionary Approach, Sinauer Associates, Sunderland, Massachusetts, 1998.

[22] W.J. Bell, Searching Behavior: The Behavioral Ecology of Finding Resources, Chapman and Hall, London, England, 1991.

[23] D. Grunbaum, Schooling as a strategy for taxis in a noisy environment, Evol. Ecol. 12 (1998) 503–522.

[25] K.M. Passino, Biomimicry of Bacterial foraging for Distributed Optimization, University Press, Princeton, NJ, 2001.

[26] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst. Mag. (2002).

[31] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, Berlin, Heideberg, 1996.

[32] S. Mishra, A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation, IEEE Trans. Evol. Comput. 9 (February (1)) (2005) 61–73.

[33] D.H. Kim, Genetic algorithm combined with particle swarm optimization/bacterial foraging and its application to PID controller tuning, Thesis, Tokyo Institute of Technology, Japan, April 2006.