

## Research Article

# Bacterial Foraging-Tabu Search Metaheuristics for Identification of Nonlinear Friction Model

**Nuapett Sarasiri, Kittiwong Suthamno, and Sarawut Sujitjorn**

*Power Electronics, Machines, and Control Research Group, Control and Automation Research Unit, School of Electrical Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand*

Correspondence should be addressed to Sarawut Sujitjorn, sarawut@sut.ac.th

Received 20 February 2012; Accepted 28 April 2012

Academic Editor: Hak-Keung Lam

Copyright © 2012 Nuapett Sarasiri et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes new metaheuristic algorithms for an identification problem of nonlinear friction model. The proposed cooperative algorithms are formed from the bacterial foraging optimization (BFO) algorithm and the tabu search (TS). The paper reports the search comparison studies of the BFO, the TS, the genetic algorithm (GA), and the proposed metaheuristics. Search performances are assessed by using surface optimization problems. The proposed algorithms show superiority among them. A real-world identification problem of the Stribeck friction model parameters is presented. Experimental setup and results are elaborated.

## 1. Introduction

AI-based algorithms have been successfully applied to solve optimal solutions of complex and NP-hard problems in engineering. For some examples, dated back to 1995, the genetic algorithm was proposed to solve line-balancing problem to minimize the cycle time of the line for a given number of workstations [1], similar approach was used to optimize the determination of an optimal control sequence in model-based predictive control [2], the simulated annealing and genetic algorithms were used to solve nonlinear controller parameter optimization for the diving and heading motions of a submarine model [3], and recently the particle swarm optimization algorithm was applied to optimize the parameters of a fuzzy control system that was used for the vibration control problem of a flexible structure [4], and so forth. Of a particular interest in control, current motion control technologies demand very high precision in positioning of an object, for example, in CNC machines, robots, and so forth. Difficulties caused by nonlinear friction arise under very low velocity motions, in which stick-slip phenomenon pronouncedly exhibits. Control techniques can

be used to compensate for the problem, and as such an accurate friction model is needed. To obtain the model it requires careful experimental setup, accurate measurement and appropriate identification algorithms. Due to strong nonlinearity of the friction model, the conventional regressive approach is inappropriate.

An identification task can be formulated as an optimization problem solvable by available optimization algorithms. Artificial intelligence- (AI-) based methods are efficient candidates of the present technology. There is a wide range of algorithms which has been applied to solve identification problems. For some instances, the genetic algorithm has been applied to various problems including the modelling of a laboratory scale process involving a coupled water tank system and the identification of a helicopter rotor speed controller [5] and the identification of induction machine parameters [6, 7]; the tabu search and simulated annealing algorithms were applied to identify the optimal parameter structure for groundwater models [8]; the adaptive tabu search was applied to harmonic identification for an active power filter [9]; the particle swarm optimization was applied to solve various problems including the identification of thermal power plant [10], the calculation of deformations in soil or rock in geotechnical engineering [11], and the identification of MIMO FIR systems [12]; the ant colony optimization algorithm was used to identify the aquifer parameters for the underground water control engineering [13]; the proportionate affine projection algorithm was applied to the identification of sparse impulse response [14]; recently, the bacterial foraging algorithm was applied for the radiofrequency identification (RFID) communication system [15].

Among those metaheuristics, evolutionary and bioinspired algorithms have gained major interests since they are not hard to understand, and programming according to the procedural lists is not so strict compared with conventional scientific programming. It also opens a new route to effectively obtain an optimal or suboptimal solution for a complex system. This paper proposes the use of the tabu search (TS) and the bacterial foraging optimization (BFO) algorithm in a cooperative manner. Due to the dominant explorative property of the BFO algorithm, the modified TS with the BFO built-in is able to start searching with an elite initial solution. While the sharp focusing property of the TS remains, the proposed algorithms can move towards the solution very rapidly. Section 2 of the paper explains the algorithms. The search performance was investigated using some well-known unconstrained optimization problems on Pentium IV, 2.4 GHz, 640 Mbytes SD-RAM. The search results are compared among the BFO, the TS, the genetic algorithm (GA), and the modified TS denoted as bacterial foraging-tabu search (BTS). Since the GA is well known, review of the algorithms is omitted herein. The comparative results of performance studies are presented in Section 3. The proposed algorithms are applied to identify the parameters of the nonlinear Stribeck friction model. Section 4 presents the experimental setup, results, and discussions. Conclusions follow in Section 5.

## **2. Bacterial Foraging-Tabu Search Metaheuristics**

### **2.1. Tabu Search**

Tabu search (TS) originated by Glover [16, 17] has become one of the most efficient metaheuristic methods. It incorporates two major strategies, namely, intensification and diversification, respectively. Successful applications of the TS have appeared in various fields, for example, food processing [18], optimal power flow [19], flow shop problem [20], and so forth. For some complex systems containing many local optima, the simplistic TS is

usually unable to release the search move from a local entrapment. This problem has been overcome by different modifications made to the TS. These include the reactive tabu search [21], probabilistic tabu search [22] and adaptive tabu search (ATS) [23]. The ATS has found various successful applications such as identification [9], control [24], and signal processing [25].

The ATS consists of two major additional strategies made to the conventional TS. These are back-tracking (BT) and adaptive search radius (AR) mechanisms, respectively. The former assists the TS to release itself from being locked by a local solution. It looks up the tabu list (TL), that is, short-term memory, for a visited elite solution, and uses this solution for starting a new search move. The later enhances the focusing characteristic of the TS. This strategy decreases the search radius gradually when the search comes close to a solution of high quality having a potential of being the optimal one. However, too short of the search radius could result in a slow search. Recommendations for selection of search parameters are in [23]. The ATS algorithm is presented as a part of the proposed algorithms in Section 2.3.

## **2.2. Bacterial Foraging Optimization**

In 2002, Passino developed a new bioinspired optimization algorithm called bacterial foraging optimization (BFO) [26, 27]. The algorithm imitates the foraging behavior of *E. coli* bacteria; the computer codes appear in [http://www2.ece.ohio-state.edu/~passino/ICbook/ic\\_code.html](http://www2.ece.ohio-state.edu/~passino/ICbook/ic_code.html). The BFO has been applied to various optimization problems including estimation of harmonics [28], active power filter design [29], transmission loss reduction [30], and optimal power flow [31]. Unfortunately, the BFO sometimes does not converge to a high-quality solution, particularly when applied to complicated problems. The difficulty has been resolved by some researchers [32, 33] who modified the chemotaxis step of the BFO to become an adaptive mechanism and, hence, the name adaptive bacterial foraging optimization, or ABFO. It incorporates an adaptive jumping step denoted as  $C(i)$  under a basic concept of a long step corresponding to a large deviation of the cost value from the targeted one, and vice versa. The ABFO algorithm consists of 4 main mechanisms as follows.

### *Chemotaxis*

This mechanism imitates the swimming movement of a bacterium. The position of a bacterium is denoted as  $\theta^i$ .

### *Swarming*

When one bacterium presents itself in an elite position, that is, a local hill or valley, it attracts the others. Simultaneously, each bacterium tries to repel the others nearby. The attractive and the repellent effects are modeled as weighted summation of exponential terms presenting the objective function,  $J_{CC}$ . The weighting factors are  $d_{\text{attract}}$ ,  $w_{\text{attract}}$ ,  $h_{\text{repellant}}$ , and  $w_{\text{repellant}}$  and can be chosen arbitrarily.

### *Reproduction*

The bacteria are classified during the computing process as healthy and unhealthy due to their cost values. Only the healthy ones reproduce by duplicating themselves at the same positions.

### Elimination and Dispersal

The mechanism allows the unhealthy bacteria to be discarded. The healthy ones are dispersed randomly over the search space with the probability  $P_{ed}$ .

The original BFO and the ABFO algorithms run iteratively and terminate on the maximum iteration criterion. The solutions obtained from search are stored in a memory and eventually sorted to find the optimal solution. From testing the ABFO algorithm, it demonstrates a strong explorative (or diversification) property. This property is commonly found in population-based algorithms, and the ABFO algorithm is one of them. In contrast, single-solution-based algorithms, such as the TS, have strong exploitative property [34–36]. Therefore, both algorithms complement each other in the senses that the ABFO is useful for provision of an elite initial solution to the TS, and the TS is an efficient tool to track down a global solution rapidly. The ABFO algorithm is presented as a part of the algorithms in the next section.

### 2.3. Bacterial Foraging-Tabu Search

As mentioned, the TS has a dominant focusing characteristic, while the ABFO is strong in explorative operation. Such properties can complement each other. Since the TS has straightforward procedures, and moves rapidly towards a local solution, the method forms the hunting steps for a satisfied solution to the problem. The two algorithms are combined to form new metaheuristics working in a cooperative manner. The new algorithms are referred to as bacterial foraging-tabu search or BTS in short. In this new algorithmic form, it is unnecessary to employ the reproduction mechanism of the ABFO part because ranking the available solutions to single out one with the minimum cost is an important step. This specific solution is transferred to the TS part as an initial solution. The procedural list of the BTS algorithms is as follows.

*Step 1.* Initialize search parameters:  $p$ , search space,  $S$ ,  $N_c$ ,  $N_s$ ,  $\alpha$ ,  $d_{\text{attract}}$ ,  $w_{\text{attract}}$ ,  $h_{\text{repellant}}$ ,  $w_{\text{repellant}}$ ,  $R$ ,  $N$ , TL,  $\text{count}_{\text{max}}$ , BT,  $n\_re\_back$ ,  $best\_neighbor1$ ,  $best\_error$ ,  $R_i$  and  $\epsilon_i$ .

*Step 2.* Randomly or heuristically select an initial solution  $\theta^i$  from the search space. Set  $\theta^i$  as the current solution.

*Step 3.* Compute objective functions  $J(i, j)$  according to (2.1) ( $i = 1, 2, \dots, S$ ). Set  $J_{\text{last}} = J(i, j)$ ;  $j = 1, 2, \dots, N_c$ ,

$$\begin{aligned}
 J(i, j) &= J(i, j) + J_{\text{CC}}(\theta^i(j), P(i, j)) \\
 J_{\text{CC}}(\theta^i, P(i, j)) &= \sum_{i=1}^S J_{\text{CC}}^i(\theta, \theta^i(j)) \\
 &= \sum_{i=1}^S \left[ -d_{\text{attract}} \exp\left(-w_{\text{attract}} \sum_{m=1}^p (\theta_m - \theta_m^i)^2\right) \right] \\
 &\quad + \sum_{i=1}^S \left[ h_{\text{repellent}} \exp\left(-w_{\text{repellent}} \sum_{m=1}^p (\theta_m - \theta_m^i)^2\right) \right].
 \end{aligned} \tag{2.1}$$

*Step 4.* Generate randomly  $([-1, 1])$  the elements of the random vector  $\Delta_p(i) \in \mathbb{R}^p$ , then compute the adaptive step size,  $C(i, j)$  using (2.2), and update the solution  $\theta^i(j+1)$  according to (2.3). Compute the objective function for  $j = j + 1$  according to (2.1). Set  $m = 0$ ,

$$C(i, j) = \frac{|J(i, j)|}{|J(i, j)| + \alpha} = \frac{1}{1 + \alpha/|J(i, j)|}, \quad (2.2)$$

$$\theta^i(j+1) = \theta^i(j) + C(i, j) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}. \quad (2.3)$$

*Step 5.* If  $J(i, j+1) < J_{\text{last}}$  then  $J_{\text{last}} = J(i, j+1)$ ; use the direction of the same random vector  $\Delta(i)$  to compute  $\theta^i(j+1)$  and  $J(i, j+1)$ . Update  $m$  and repeat Step 5 until  $m > N_s$ .

*Step 6.* If  $j \leq N_c$ , go to Step 3.

*Step 7.* Do minimum sorting of the objective functions  $J$ . Define  $best\_theta$  as the solution with the minimum  $J$ . Set  $S_0 = best\_theta$ .

*Step 8.* Generate a neighbourhood around  $S_0$  within an initial search radius  $R$ . Set  $N$  solutions as the members of the set  $S_1(r)$ .

*Step 9.* Evaluate the objective function of each member belonging to  $S_1(r)$ . Define  $S_1 = best\_neighbor1$  as a solution with the minimum cost,  $J_1$ .

*Step 10.* If  $J_1 < J_0$ , store  $S_0$  in the TL, assign  $S_0 = S_1$ , otherwise, store  $S_1$  in the TL.

*Step 11.* Invoke the BT when a solution deadlock occurs (the current solution has been repeated many times as defined by  $n = 1, 2, \dots, BT$ ) (Algorithm 1).

*Step 12.* If the termination criterion based on the  $J$  values is met or  $count > count_{\text{max}}$  ( $count = 1, 2, \dots, count_{\text{max}}$ ) exit with the global solution.

*Step 13.* Invoke the AR when the current solution is relatively close to a local minimum (Algorithm 2).

*Step 14.* Updated count. If  $count \leq count_{\text{max}}$  then go to Step 8.

Referring to Step 12, one termination criterion is the maximum number of iterations. There are other approaches the users may employ, that is, maximum CPU time, maximum iteration with or without improvement in solution quality, and a solution of sufficient quality [35]. Another termination criterion is the sufficient solution quality concept, which is represented by the cost  $J$ . The value of the cost  $J$  depends on application. For engineering problems, the cost  $J$  can be set from design specifications, component tolerances, and so forth. The maximum number of iterations ( $count_{\text{max}}$ ) can be determined from the ratio of search boundary to search radius of the TS. Some presearch trials are useful to determine an adjustment to the value of  $count_{\text{max}}$ . The above algorithms are general enough for various optimization problems. Specific alterations for the problem of friction model identification will be discussed in Section 4.

```

BT: if  $n \geq BT$ 
     $n = n + 1$ 
     $best\_error = RANK(TL)$ 
    look back in the  $TL$ , then retrieve the  $n.re.back^{th}$  solution from the  $TL$ .
else
     $n = 0$ 
    define  $S_0 = best\_neighbor$ 
     $best\_error = best\_error$ 
end if

```

Algorithm 1

```

AR: if  $best\_error < \varepsilon_1$ 
     $R = R_1$  where  $R < R_1$ .
end
    if  $best\_error < \varepsilon_2$ 
     $R = R_2$  where  $R_2 < R_1$  and  $\varepsilon_2 < \varepsilon_1$ .
end
    ...
    if  $best\_error < \varepsilon_n$ 
     $R = R_n$  where  $R_n < R_{n-1}$  and  $\varepsilon_n < \varepsilon_{n-1}$ .
end

```

Algorithm 2

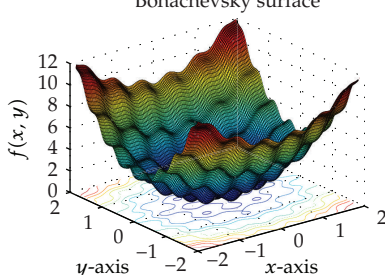
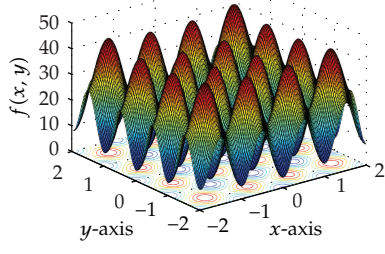
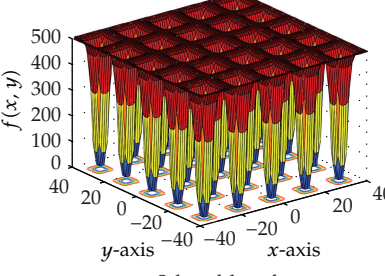
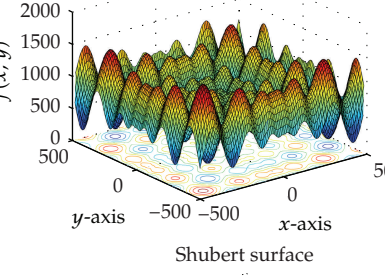
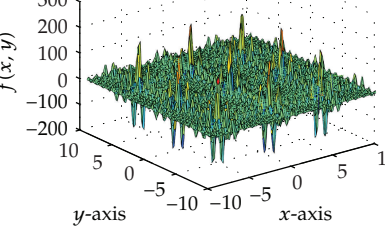
### 3. Search Performance

This section presents the performance comparison studies among the following algorithms: adaptive tabu search (ATS), adaptive bacterial foraging optimization (ABFO), bacterial foraging-tabu search metaheuristics (BTS), and genetic algorithm (GA). Review of the GA is omitted since the algorithm is well known. Good sources that readers may refer to are [37–39]. Each of these algorithms performs search on several test functions for 50 trials, and the results are averaged. Each search trial begins the search with different initial solutions, while search parameters are kept the same for all trials.

This approach is commonly referred to as multiple-points-single-strategy (MPSS) in metaheuristic contexts. The test functions adapted are well-known unconstrained problems for testing optimization algorithms. These include Bohachevsky function (BF), Rastrigin function (RF), Shekel's fox-holes function (SF), Schwefel function (SchF), and Shubert function (ShuF), respectively. Table 1 summarizes these test functions in which  $J_{\min}$  is the minimum cost required to terminate the search. Search parameter settings for the ATS follow [23], the for ABFO follow [26], and for the GA follow MATLAB-GA Toolbox [39]. Tables 2 and 3 summarize the search parameters of the ATS and the ABFO, respectively. These parameters are adapted for the BTS with  $N_c = 20$ , in particular.

Table 4 summarizes the average results over 50 trials. There are 2 groups of data denoted as average search time and average search rounds, respectively. Since the ATS, ABFO, and GA have different algorithmic approaches, comparisons of their average search rounds are not meaningful. On the contrary, the ABFO is combined to the ATS in order to

**Table 1:** Summary of the unconstrained problems used for performance test.

Test functions	Equations	Surfaces in 3D
BF	$f(x, y) = x^2 + 2y^2 - 0.3 \cos(3\pi x) - 0.4 \cos(4\pi y) + 0.7$ global minimum of 0 at (0,0), search space: [2 2; -2 -2] and $J_{\min} \leq 1 \times 10^{-9}$ .	Surfaces in 3D Bohachevsky surface 
RF	$f(x, y) = x^2 + y^2 - 10 \cos(2\pi x) - 10 \cos(2\pi y) + 20$ global minimum of 0 at (0,0), search space: [2 2; -2 -2] and $J_{\min} \leq 1 \times 10^{-8}$ .	Rastrigin surface 
SF	$f(x_1, x_2) = [1/500 + \sum_{j=1}^{25} (1/(j + \sum_{i=1}^2 (x_i - a_{ij})^6))]^{-1}$ when $a_{ij} = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 & 32 & 32 \end{pmatrix}$ . global minimum of 0.998 at (-32, -32), search space: [40 40; -40 -40] and $J_{\min} \leq 0.9990$ .	Shekel foxholes surface 
SchF	$f(x) = 418.9829n - \sum_{i=1}^n (x_i \sin \sqrt{ x_i }); n = 2$ global minimum of 0 at (420.9687, 420.9687), search space: [500 500; -500 -500] and $J_{\min} \leq 1 \times 10^{-1}$ .	Schwefel surface 
ShuF	$f(x_1, x_2) = \sum_{i=1}^5 i \cos((i+1)x_1 + i) * \sum_{i=1}^5 i \cos((i+1)x_2 + i)$ when $-10 \leq x_1, x_2 \leq 10$ . global minima of equal values -186.7309 at 18 different locations, search space: [10 10; -10 -10] and $J_{\min} \leq -186.73$ .	Shubert surface 

**Table 2:** ATS parameters.

Test functions	$N$	$\text{Count}_{\max}$	$R$	$\text{BT}, n_{re\_back}$	AR		
					Stage I	Stage II	Stage III
					BF	30	10,000
RF	30	10,000	0.2	5	$R = 2 \times 10^{-3}$	$R = 2 \times 10^{-4}$	—
SF	30	10,000	0.8	5	$J < 5, R = 0.5$	$J < 2, R = 0.1$	—
SchF	30	10,000	50	5	$J < 100, R = 25$	$J < 10, R = 10$	$J < 1, R = 1$
ShuF	30	10,000	1.0	5	$J < 1, R = 1 \times 10^{-2}$	$J < -1, R = 1 \times 10^{-3}$	—

**Table 3:** ABFO parameters.

Test functions	ABFO parameters										
	$S$	$N_c$	$N_S$	$N_{re}$	$N_{ed}$	$P_{ed}$	$\alpha$	$d_{\text{attract}}$	$h_{\text{repellant}}$	$w_{\text{attract}}$	$w_{\text{repellant}}$
BF	30	20	4	4	2	0.25	10	0.1	0.1	0.2	1
RF	30	200	4	4	2	0.25	100	0.1	0.1	0.2	1
SF	30	500	4	4	2	0.25	0.01	0.1	0.1	0.2	10
SchF	30	2000	4	4	2	0.25	1	0.1	0.1	0.2	10
ShuF	30	1000	4	4	2	0.25	1	0.1	0.1	0.2	10

reduce search rounds and search time. So, it is meaningful to compare the search rounds consumed by the ATS and the BTS. In average, the proposed BTS consumes search rounds of 63.45% less than the ATS does. Referring to the search time data in Table 4, the BTS spends search time of 37.15% less than the ATS does, and 68.31% less than the ABFO does as averages. Moreover, the BTS consumes 58.21% less search time than the GA does. In terms of the number of local entrapment, the BTS encounters the entrapment of 73.43% less than the ATS does, and produces high-quality solution to the problem. Convergence curves are shown in Figure 1 for comparison purposes.

Table 5 summarizes the solutions obtained from different methods. It can be noticed that the ABFO provides solutions with the best quality in an exchange of a considerably long search time (see Table 4). The proposed BTS provides solutions with second to the best quality within the shortest search time (see Table 4). Note that the solutions found by the BTS meet the criterion of minimum cost. This outstanding performance of the BTS is achieved due to the explorative characteristic of the ABFO, the exploitative characteristic of the TS, and the deadlock releasing property of the ATS.

Figure 2 shows a representation of bacteria movements on the search space of the test functions. Noticeably, large areas of search spaces are explored by the bacteria. In other words, the BTS can provide a high-quality solution rapidly because the ABFO provides an elite initial solution to the ATS, and the ATS effectively releases the search from a deadlock or local entrapment while rapidly focuses the search to the solution.

The BTS has been applied to a constrained parametric search problem, that is, an identification of the nonlinear friction model. In the next section, experimental setup, and identification results are presented.



**Table 4:** Summary of the results (averaged over 50 trials).

Test functions	Average search time (seconds)				Average search rounds			
	ATS	ABFO	BTS	GA	ATS	ABFO ( $N_C$ )	BTS	GA
BF	11.66	5.75	6.83	49.12	616.48	20	151.20	1177.18
RF	14.60	48.63	5.81	11.78	868.28	200	323.30	225.54
SF	4.18	146.67	3.69	8.53	139.36	500	25.70	141.28
SchF	408.58	728.87	172.52	868.32	6889.42	2000	1469.14	354.44
ShuF	3.28	182.83	2.80	3.39	68.06	1000	55.28	34.24

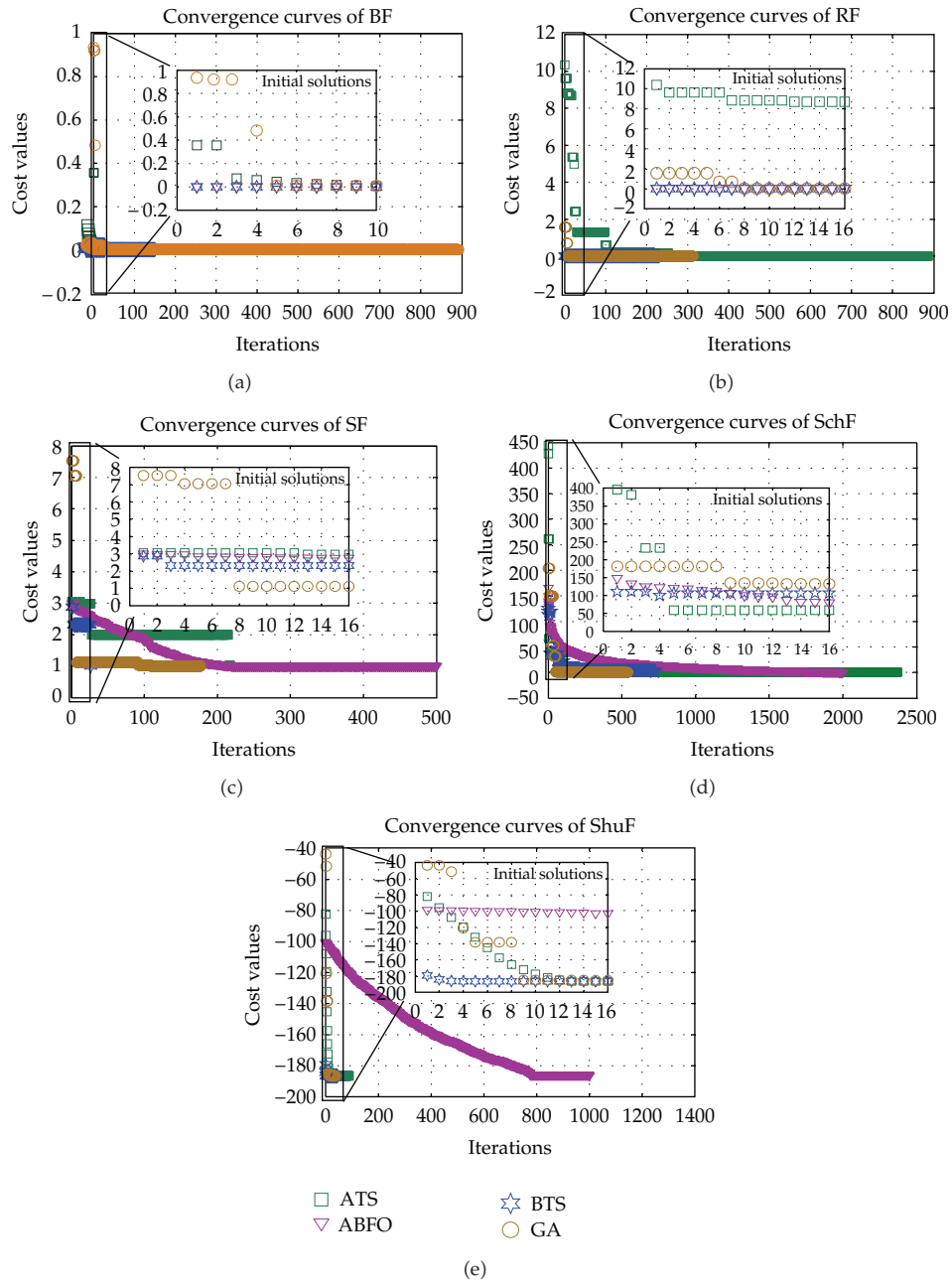
**Table 5:** Solutions obtained from different approaches.

Objective function	ATS	ABFO	BTS	GA
BF:				
Average	$4.5090e-10$	$1.0833e-10$	$5.30112e-10$	$6.0074e-08$
Min	$7.3184e-11$	$2.2204e-16$	$3.569e-12$	$1.1830e-11$
Max	$9.4151e-10$	$8.8759e-10$	$9.97229e-10$	$2.9855e-06$
Std.	$2.7476e-10$	$2.0742e-10$	$3.13515e-10$	$4.2216e-07$
RF:				
Average	$5.3478e-09$	$1.1534e-10$	$5.2460e-09$	$4.9642e-09$
Min	$2.9051e-10$	$3.5527e-15$	$5.9587e-10$	$1.1486e-10$
Max	$9.3771e-09$	$6.7094e-10$	$9.6620e-09$	$9.5235e-09$
Std.	$2.4328e-09$	$1.4942e-10$	$2.6721e-09$	$2.6740e-09$
SF:				
Average	0.9982	0.9981	0.9983	0.9983
Min	0.9980	0.9980	0.9980	0.9980
Max	0.9989	0.9988	0.9990	0.9990
Std.	0.0002	0.0002	0.0003	0.0003
SchF:				
Average	0.0456	0.0139	0.0273	0.0439
Min	0.0006	$2.5455e-05$	0.0002	0.0008
Max	0.0998	0.0768	0.0944	0.0982
Std.	0.0278	0.0240	0.0341	0.0281
ShuF:				
Average	-186.7305	-186.7305	-186.7305	-186.7304
Min	-186.7309	-186.7309	-186.7309	-186.7309
Max	-186.7300	-186.7300	-186.7301	-186.7300
Std.	0.0003	0.0003	0.0003	0.0002

## 4. Identification Results

### 4.1. Experimental Setup

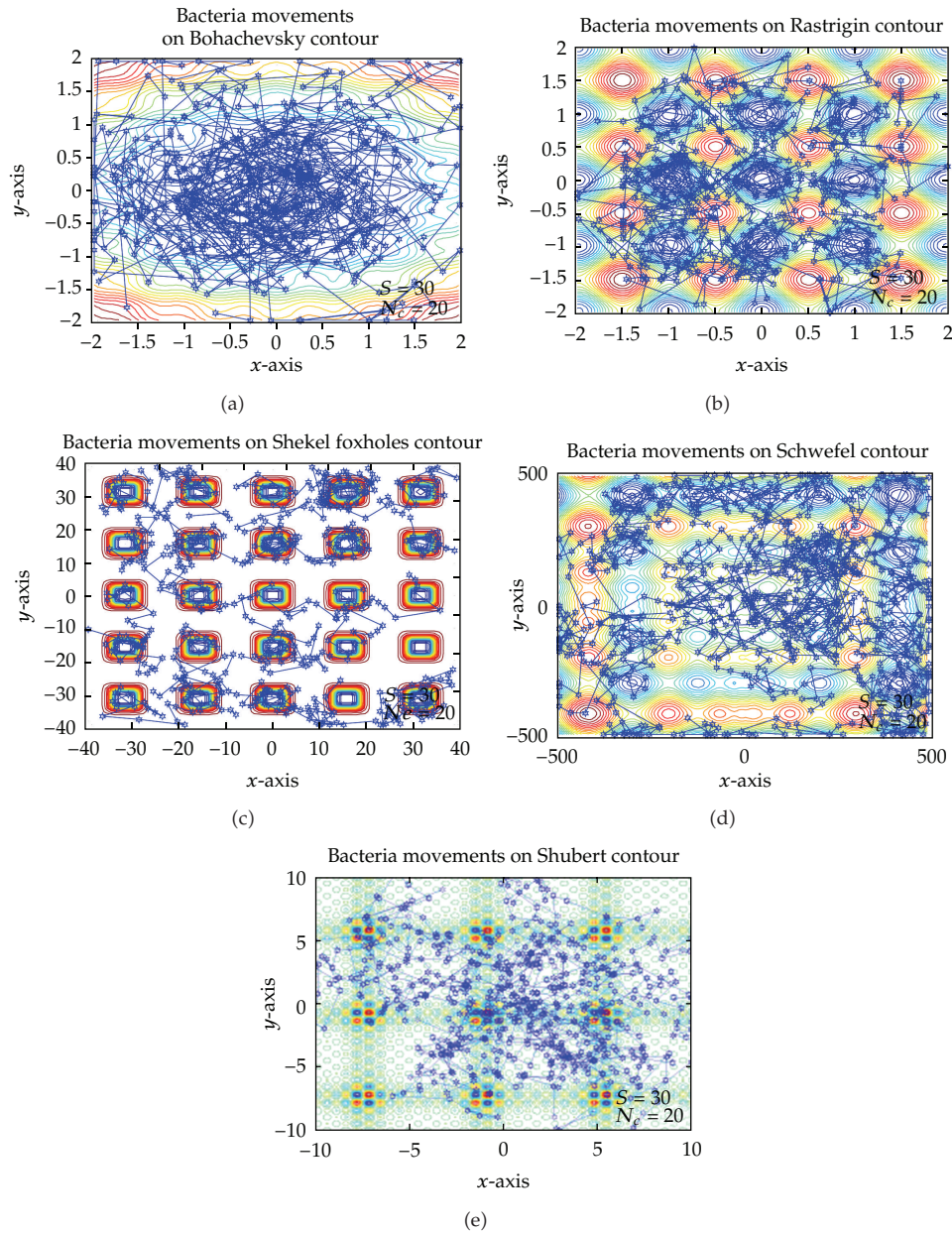
A closed loop position control system is a necessary test bed for monitoring stick-slip phenomenon. The diagram in Figure 3 represents the experimental setup. The linear slide bed



**Figure 1:** Convergence curves—(a) BF, (b) RF, (c) SF, (d) SchF, and (e) ShuF.

is the controlled plant consisting of a dc motor, a threaded rod, a reflector, and an ultrasonic transducer (UC3000-UIE2). The effective moving range of the reflector is 0–400 mm with the home position at the middle. In test mode, the motion control circuit performs an initial test move of the reflector for the whole range and, eventually, places the reflector at the home position.

For the reflector to follow a ramp command, a closed loop position control has been built. The hardware components consist of a PC as a P-controller, a 12-bit ADC, a 2Q-drive



**Figure 2:** Bacterial search movements—(a) BF, (b) RF, (c) SF, (d) SchF, and (e) ShuF.

circuit, a current sensor, an ultrasonic transducer, a 2nd-order differentiator producing a speed signal from a position signal, and a few signal conditioning circuits including zero-span circuits and a bipolar voltage generator, respectively, and a dc power supply. In control mode, the motion follows an up-down ramp command directing the reflector to move rightward (positive direction, ramp-up command) and leftward (negative direction, ramp-down command). The reflector moves in the range of 50–350 mm in the control mode. A desired speed can be set via the keyboard of the PC functioning as a P-controller.

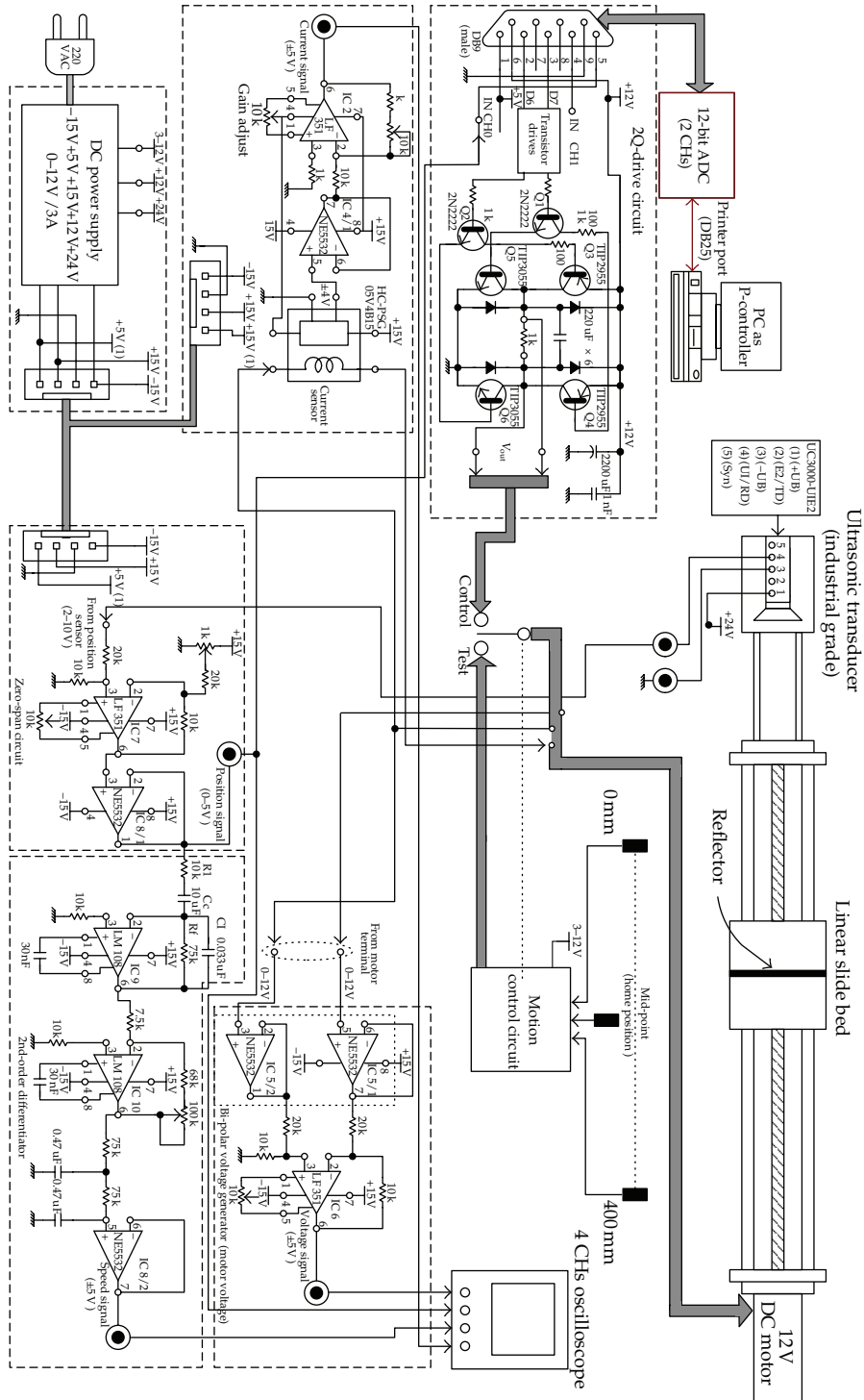


Figure 3: Circuit diagram representing the experimental setup.

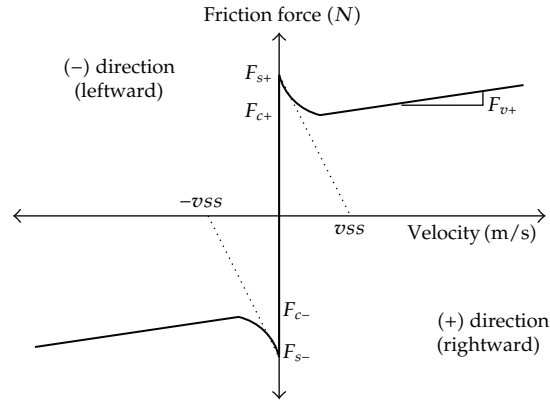


Figure 4: Stribeck friction curve.

#### 4.2. Nonlinear Friction Model

When two solid materials translating over one another at very low velocity, a stick-slip phenomenon occurs. This phenomenon is caused by nonlinear friction characteristics also known as Stribeck's effect [40]. An effective model describing the friction can be represented by the curve in Figure 4, and is referred to as complex friction model or Stribeck model [41–43]. When an applied force to a mass cannot overcome the static friction, which is represented by either  $F_{S+}$  or  $F_{S-}$  depending on the direction of motion, the mass cannot move.

This situation is referred to as stick mode, and described by the stick-friction force

$$F_{\text{stick}}(F_{\text{in}}) = \begin{cases} F_{S+}, & F_{\text{in}} > F_{S+} \\ F_{\text{in}}, & F_{S-} \leq F_{\text{in}} \leq F_{S+} \\ F_{S-}, & F_{\text{in}} < F_{S-}. \end{cases} \quad (4.1)$$

Once the applied force is greater than the static friction, the mass begins moving. After a certain period of time, the mass keeps up a higher velocity during which it encounters both Coulomb and viscous frictions. This situation is known as slip mode and described by the slip-friction force

$$F_{\text{slip}}(v) = \left( F_C + (F_S - F_C) \cdot e^{-(v/v_{ss})} \right) \cdot \text{sgn}(v) + F_v \cdot v. \quad (4.2)$$

To cover the whole velocity range, the friction force can be expressed in a compact form as

$$F_f(v, F_{\text{in}}) = \begin{cases} F_{\text{stick}}(F_{\text{in}}), & |v| = 0 \\ F_{\text{slip}}(v), & |v| \neq 0. \end{cases} \quad (4.3)$$

### 4.3. Objective Function Implementation

During the search process to identify the friction model parameters, an objective function ( $J$ ) has to be evaluated repeatedly. To calculate the objective function, it is assumed that the translational dynamic can be represented by the mass-spring model as follows:

$$m\ddot{x} = k_{\text{spring}}(x_i - x_d) - F_f(v, F_{\text{in}}) + F_{\text{ex}}, \quad (4.4)$$

in which the parameters  $F_S$ ,  $F_C$ ,  $F_V$ ,  $v_{SS}$ , and  $k_{\text{spring}}$  are to be identified. Below is the procedural list for objective function calculation.

*Step 1.* Calculate an average displacement  $x_i = (\sum_{j=1}^{50} x_{\text{test}}(j))/50$ .

For ramping-up motion, calculate an approximated force  $F_{\text{motor}} = \hat{k}(x_i - x_d) \approx 3.5(x_i - x_d)$ .

For ramping-down motion, calculate an approximated force  $F_{\text{motor}} = \hat{k}(x_i - x_d) \approx 2.8(x_i - x_d)$ .

An approximated force  $F_{\text{motor}} = \hat{k}(x_i - x_d) \approx 2.8(x_i - x_d)$ .

*Step 2.* If  $(F_{\text{motor}} \geq F_{\text{fm}})$  then  $(F_{f\_motor} = F_{\text{fm}})$ .

If  $(F_{\text{motor}} \leq -F_{\text{fm}})$  then  $(F_{f\_motor} = -F_{\text{fm}})$ .

If  $(-F_{\text{fm}} < F_{\text{motor}} < F_{\text{fm}})$  then  $(F_{f\_motor} = F_{\text{motor}})$ .

*Step 3.* Calculate  $F_{m\_motor}(v) = K_{\text{kt}}(J_m(dv/dt) + D_m v)$ .

If  $(F_{m\_motor} > 30 \text{ N})$  then  $(F_{m\_motor} = 30 \text{ N})$ .

If  $(F_{m\_motor} < -30 \text{ N})$  then  $(F_{m\_motor} = -30 \text{ N})$ .

*Step 4.* Calculate the following forces:

externally applied force— $F_{\text{ex}} = F_{\text{motor}} - F_{f\_motor}(F_{\text{motor}}) - F_{m\_motor}(v)$ ,

spring force— $F_{\text{sp}} = k_{\text{spring}}(x_i - x_d)$ ,

internally applied force— $F_{\text{in}} = F_{\text{ex}} + F_{\text{sp}}$ ,

stick friction force—if  $(F_{\text{in}} > F_S)$  then  $(F_{\text{stick}} = F_S)$ , if  $(F_{\text{in}} < -F_S)$  then  $(F_{\text{stick}} = -F_S)$ ,

if  $(-F_S \leq F_{\text{in}} \leq F_S)$  then  $(F_{\text{stick}} = F_{\text{in}})$ , and slip friction force— $F_{\text{slip}}(v) = (F_C + (F_S - F_C) \cdot e^{-(v/v_{SS})}) \cdot \text{sgn}(v) + F_V \cdot v$ .

*Step 5.* Calculate velocity and displacement of the mass:

$P = \int [F_{\text{in}} - F_f(v, F_{\text{in}})] dt$ .

If  $(-dp < P < dp)$  then  $(v = 0)$  otherwise  $(v = P/m)$ .

$x_d = \int v dt$ .

*Step 6.* Calculate the objective function:

$$J = \sqrt{\frac{\sum_{i=1}^n (x_{\text{test}} - x_d)^2}{n}}. \quad (4.5)$$

*Step 7.* Return to main search.

#### 4.4. Algorithm Implementation

Regarding this identification problem, the mass ( $m$ ) is known,  $m = 10.90$  kg. There are 5 parameters to be searched, that is,  $F_S$ ,  $F_C$ ,  $F_V$ ,  $v_{SS}$ , and  $k_{\text{spring}}$ , respectively. The termination criterion is either  $J < 4.5$  or  $\text{count}_{\text{max}} = 1,000$ . The procedural list below presents the implemented algorithm for this identification problem.

*Step 1.* Initialization: search parameters:  $\{p = 5, S = 30, N_c = 20, N_S = 4, \alpha = 1 \times 10^3, d_{\text{attract}} = 0.1, w_{\text{attract}} = 0.2, h_{\text{repellant}} = 0.1, w_{\text{repellant}} = 0.1, R = 0.15, N = 40, \text{count}_{\text{max}} = 1000, \text{BT} = 5 \text{ and } n_{\text{re\_back}} = 5\}$ , search spaces:  $\{F_S = [100-180], F_C = [40-80], F_V = [0.4-1.0], v_{SS} = [1-7], \text{ and } k_{\text{spring}} = [0.1-1.5]\}$ .

*Step 2.* Randomly assign real values to the parameters to be searched for  $S$  sets. Calculate the corresponding objective functions. Select the solution set having the best objective function, and store it in the variable  $\theta^i$ .

*Step 3.* Calculate the objective functions,  $J(i, j)$ , with  $J_{cc}$  taken into account according to (2.1). Assign  $J_{\text{last}} = J(i, j)$ .

*Step 4.* Random the value of  $\Delta(i)$  in  $[-1, 1]$ . Use (2.2) to calculate  $C(i, j)$ . Calculate the next parameters,  $\theta^i(j + 1)$ , according to (2.3). Calculate the objective functions  $J(i, j + 1)$ .

*Step 5.* Evaluate the objective functions: if  $(J(i, j + 1) \leq J_{\text{last}})$  then  $(J_{\text{last}} = J(i, j + 1))$ , otherwise  $J_{\text{last}}$  remains unchanged. Update  $J(i, j + 1)$  by using  $\Delta(i)$  until iteration count =  $N_S$ .

*Step 6.* Repeat Steps 3 to 5 for  $N_c$  times.

*Step 7.* Evaluate the objective functions,  $J$ . Assign  $\text{best\_}\theta = \text{minimum value of } J \text{ just found}$ . Assign the current best solutions as the initial solutions,  $S_0$ , and their corresponding objective functions as the initial  $J_0$  values.

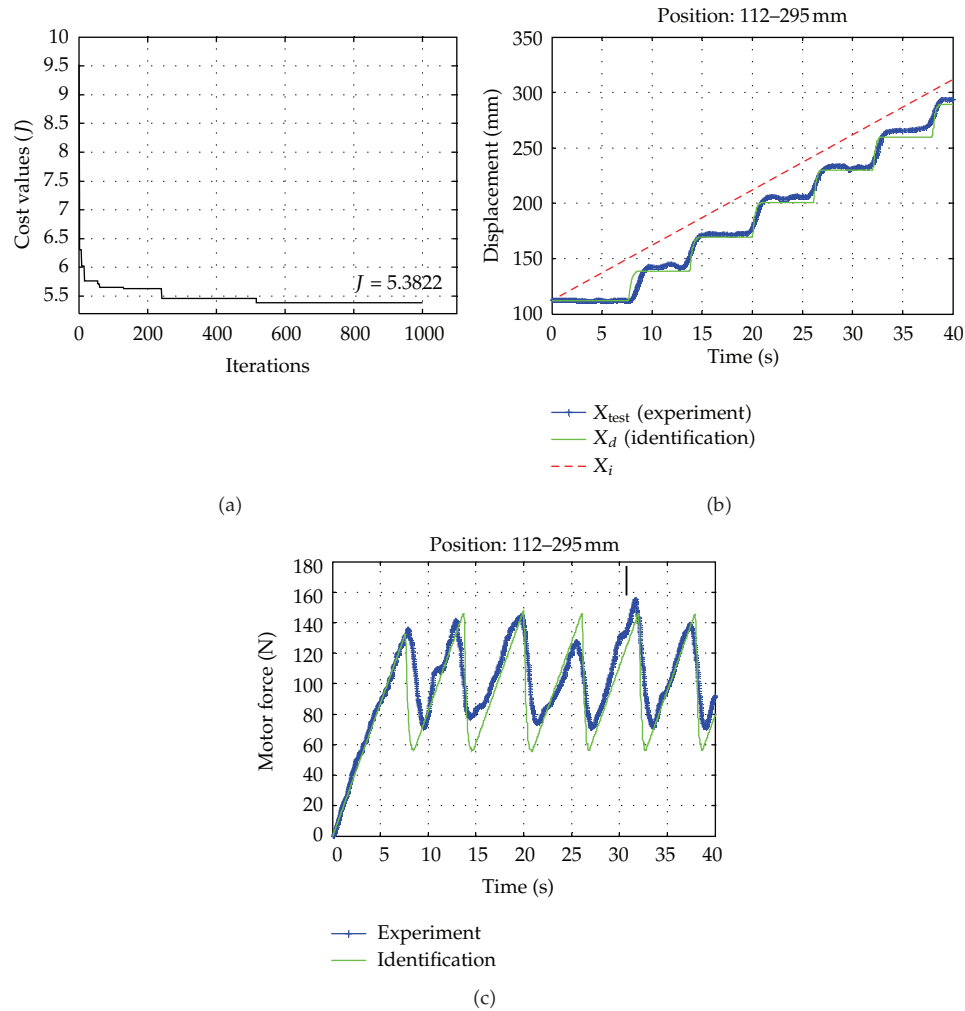
*Step 8.* In the neighborhood of  $S_0$  with the search radius  $R$ , create randomly  $N$  sets of solutions and store them in the set  $S_1(r)$ . Calculate the objective functions for all solutions according to the procedures described in Section 4.3.

*Step 9.* Based on the objective functions, do minimum sorting for the solutions in  $S_1(r)$ . Assign  $\text{best\_neighbor1} = \text{solutions with minimum objective functions and their objective functions} = J_1$ .

*Step 10.* If  $(J_1 < J_0)$  then (store  $S_0$  in the 2nd–6th columns of TL, and store  $J_0$  in the 7th column of TL), otherwise (store  $S_1$  and  $J_1$  in the TL).

*Step 11* (backtracking mechanism). If the frequency of solution cycling occurrence is equal to BT, do minimum sorting for the previous solutions stored in the TL, retrieve the 5th backward solution set, and assign it as the initial solution set for the next search move.

*Step 12.* If  $(J < 2 \text{ or } \text{count} = \text{count}_{\text{max}})$  then (terminate the search, exit and render the best solutions).



**Figure 5:** Identification results of ramp-up command at 5 mm/s—(a) convergence curve, (b) displacement, and (c) force exerted by motor. (Note: positions in the range of 112–295 mm).

*Step 13* (Adaptive search radius mechanism). If ( $best\_error < 15$ ) then ( $R = 0.0375$ ).

    If ( $best\_error < 8$ ) then ( $R = 0.0095$ ).

    If ( $best\_error < 3$ ) then ( $R = 0.0025$ ).

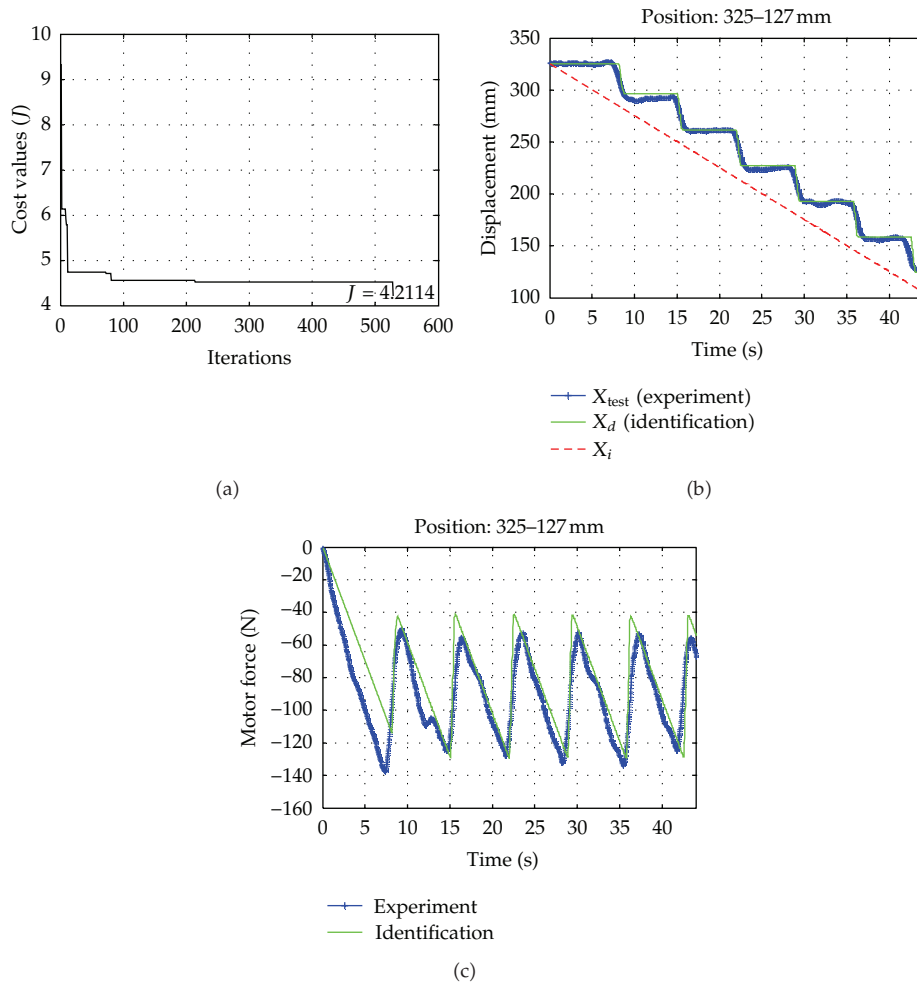
*Step 14.* Go to Step 8 until computing expires.

## 4.5. Results and Discussions

### 4.5.1. Identification

Due to the strong nonlinearity in friction force, it is necessary to identify two sets of model parameters corresponding to rightward and leftward motions. Referring to Figure 5, the illustrated graphs correspond to the rightward motion, that is, ramp-up command of 5 mm/s,

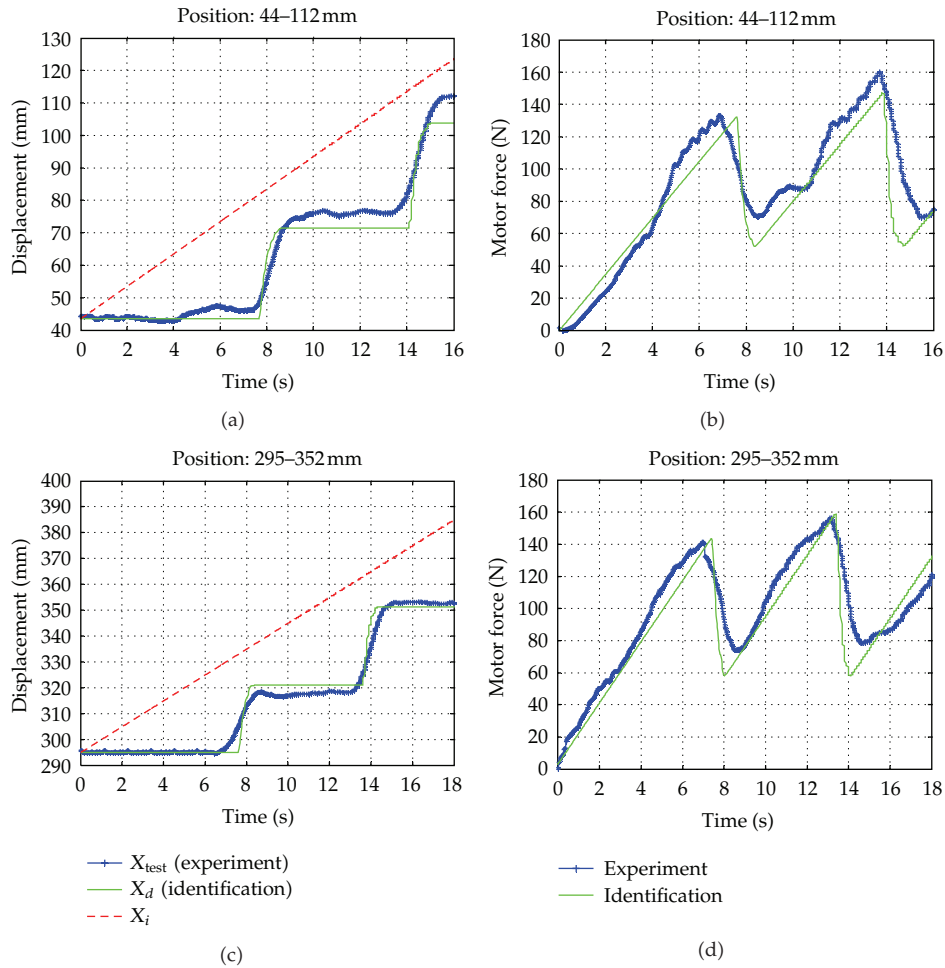




**Figure 6:** Identification results of ramp-down command at  $-5$  mm/s—(a) convergence curve, (b) displacement, and (c) force exerted by motor. (Note: positions in the range of 325–127 mm).

that uses the data in the range of 112–295 mm for identification. The convergence curve in Figure 5(a) indicates that the search terminated by  $\text{count}_{\max} = 1000$ . The cost of the returned results is  $J = 5.3822$ . The obtained parameters are as follows:  $\{F_S = 144.5463$  N,  $F_C = 47.8514$  N,  $F_V = 0.9583$  Ns/mm,  $v_{ss} = 1.0964$  mm/s,  $k_{\text{spring}} = 0.96798$  N/mm}. The experimental data and the model plots for displacement and force exerted by motor are illustrated in Figures 5(b) and 5(c), respectively. Good agreement between the experiment and the model can be observed.

For the leftward motion, that is, ramp-down command of  $-5$  mm/s, the graphical displays of identification results are shown in Figure 6. The data used for identification are in the range of 325–127 mm. As indicated by the convergence curve in Figure 6(a), the search terminated at the 528th iteration and returned the solutions with the cost  $J = 4.2114$ . The obtained parameters are as follows:  $\{F_S = -152.2804$  N,  $F_C = -40.4153$  N,  $F_V = -0.9757$  Ns/mm,  $v_{ss} = -3.21475$  mm/s,  $k_{\text{spring}} = 0.55384$  N/mm}. The experimental data and the model plots in Figures 6(b) and 6(c) show a good agreement.

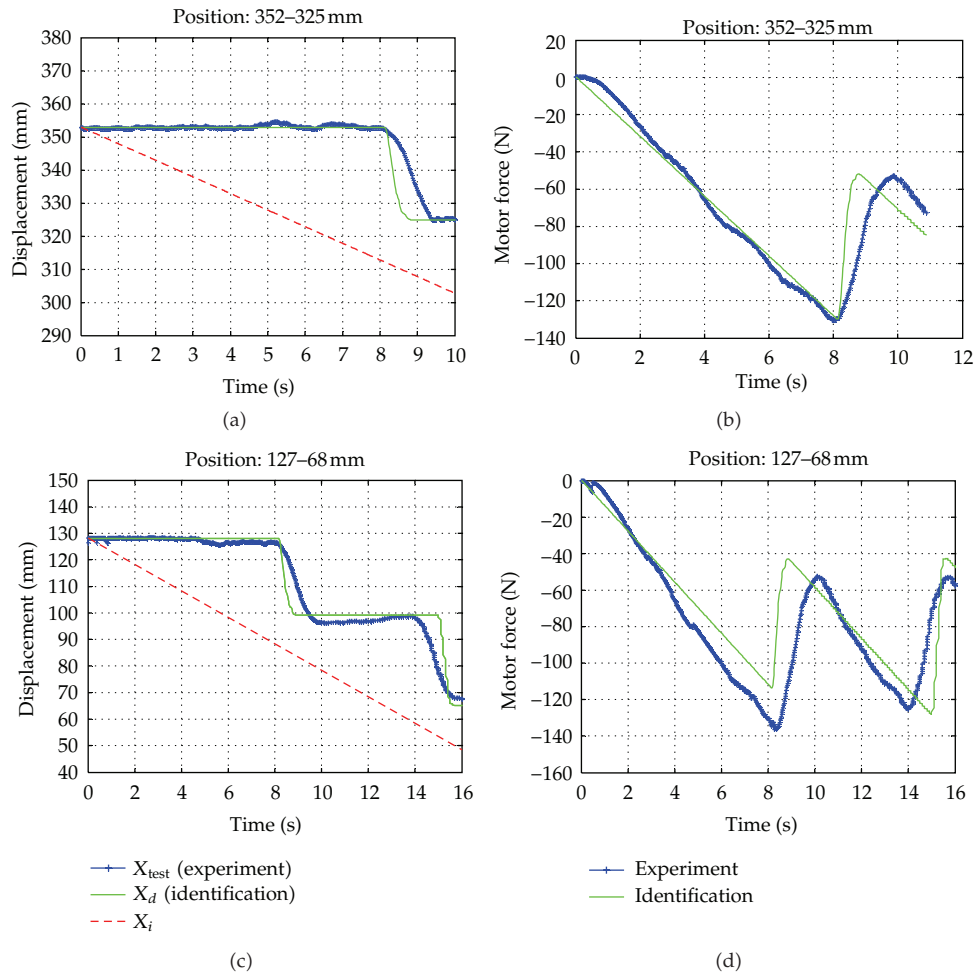


**Figure 7:** Validation results of ramp-up command—(a) displacement (44–112 mm), (b) force exerted by motor (44–112 mm), (c) displacement (295–352 mm), and (d) force exerted by motor (295–352 mm).

#### 4.5.2. Validation

Model validation was conducted for both directions of motion. Figure 7 illustrates the experimental data and the model plots for the rightward direction covering two ranges, that is, 44–112 mm and 295–352 mm. Figures 7(a) and 7(b) display the plots of the displacement and the force exerted by motor for 44–112 mm range. Similarly, the results for 295–352 mm range are shown in Figures 7(c) and 7(d). For the leftward direction covering 352–325 mm and 127–68 mm ranges, similar graphical displays are illustrated in Figures 8(a) and 8(b). Very good agreement among the practical and the theoretical results can be observed.

Furthermore, the friction curves based on model plots are shown against the experimental data in Figure 9. Very good agreement between the two can be observed. Therefore, the identified models are very good representations of the nonlinear friction forces.



**Figure 8:** Validation results of ramp-down command—(a) displacement (352–325-mm), (b) force exerted by motor (352–325 mm), (c) displacement (127–68 mm), and (d) force exerted by motor (127–68 mm).

## 5. Conclusions

This paper has proposed new metaheuristics denoted as bacterial foraging-tabu search (BTS), which are formed from the adaptive bacterial foraging optimization algorithm (ABFO) and the adaptive tabu search (ATS). The paper has elaborated the search performance assessment among the ABFO, ATS, GA, and BTS. The proposed BTS algorithms provide superior search performances as the presentation appears in Section 3. The algorithms have been applied to identify 5 parameters of the Stribeck friction model. An experimental bed of a closed-loop position control of a linear slide bed was constructed at the laboratory. The system setup is described in Section 4. Several test runs of ramp command following control were conducted for the slide bed to follow  $\pm 5$  mm/s commands such that the slide bed pronouncedly exhibited stick-slip. The experimental data were split into 2 groups for identification and validation purposes. Section 4 also elaborates important issues of objective function and algorithm implementations as well as identification results. As a result of model validation, very satisfactory model parameters have been identified by the proposed metaheuristics.

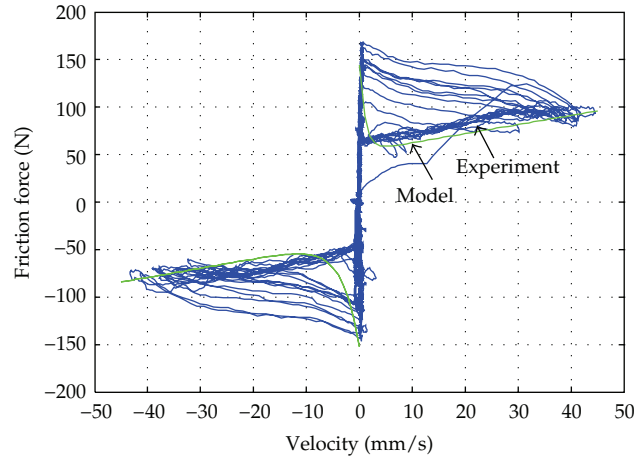


Figure 9: Plots of friction force curves (ramp command of  $\pm 5$  mm/s).

## Nomenclature

### (A) Algorithms

$\text{count}_{\max}$ :	Maximum iteration
$d_{\text{attract}}$ :	Coefficient representing the depth of attractant released
$h_{\text{repellant}}$ :	Coefficient representing the height of the repellant effect
$n_{\text{re\_back}}$ :	$k$ th backtracking solution retrieved from the TL
$p$ :	Number of parameters to be optimized
$w_{\text{attract}}$ :	Coefficient representing the width of the attractant signal
$w_{\text{repellant}}$ :	Coefficient representing the width of the repellant by the cell
AR:	Adaptive radius
BT:	Frequency of solution cycling
$C(i, j)$ :	Step size taken in random direction specified by the tumble
$J(i, j)$ :	Cost value of $i$ th bacterium
$N$ :	Number of the neighbourhood
$N_c$ :	Number of iterations to be carried out in a chemotactic loop
$N_{\text{ed}}$ :	Maximum number of elimination and dispersal events
$N_{\text{re}}$ :	Number of reproduction loop
$N_s$ :	Swimming length after which tumbling of bacteria in a chemotactic loop
$P_{\text{ed}}$ :	Probability with which the elimination and dispersal continues
$R$ :	Search radius
$S$ :	Number of bacteria in the population
$S_r$ :	A half of number of bacteria ( $S/2$ )
TL:	Tabu list
$\alpha$ :	A positive constant
$\Delta(i)$ :	Random vector on $[-1, 1]$
$\theta^i$ :	Position of $i$ th bacterium.

*(B) Test Functions*

BF: Bohachevsky function  
 RF: Rastrigin function  
 SchF: Schwefel function  
 ShuF: Shubert function  
 SF: Shekel's fox-holes function.

*(C) Identification Problem*

$a$ : Gear ratio = 5.9  
 $i$ : Motor current (A)  
 $k_{\text{spring}}$ : Gravity constant (N/mm)  
 $l$ : Ball screw lead = 5 mm  
 $m$ : Mass (kg)  
 $n$ : Number of data  
 $v$ : Velocity (mm/s)  
 $v_{\text{SS}}$ : Crossover velocity (mm/s)  
 $x_i$ : Displacement of spring (mm)  
 $x_d$ : Displacement of mass (mm)  
 $D_m$ : Viscous friction coefficient =  $2.60 \times 10^{-6}$  Nm/rad/sec  
 $F_{\text{ex}}$ : External input force (N)  
 $F_f$ : Friction force (N)  
 $F_{f,\text{motor}}$ : Friction force of motor (N)  
 $F_{\text{in}}$ : Internal input force (N)  
 $F_{m,\text{motor}}$ : Force equivalent to the inertia of motor (Nm)  
 $J_m$ : Inertia of motor =  $4.17 \times 10^{-6}$  Kg·m<sup>2</sup>  
 $K_{kt}$ : Inertia to force conversion factor =  $K_n \times K_v = (2\pi a/l)^2 \eta_b \eta_c$   
 $K_n$ : Linear to angular velocity conversion factor =  $2\pi a/l$  (rad/m)  
 $K_t$ : Torque constant of motor =  $18.2 \times 10^{-3}$  N/A  
 $K_v$ : Torque to ball screw force conversion factor =  $2\pi \eta_b \eta_c a/l$   
 $F_C$ : Coulomb friction (N)  
 $F_S$ : Static friction (N)  
 $F_V$ : Viscous friction (Ns/mm)  
 $P$ : Moment (Nm)  
 $X_{\text{test}}$ : Displacement from measured (mm)  
 $\pm dv$ : Velocity band around zero velocity =  $\pm 0.1$  mm/s  
 $\pm dp$ : Notation for the term  $\pm dv \times m$   
 $\eta_b$ : Gear box efficiency = 0.81  
 $\eta_c$ : Ball screw efficiency = 0.925  
 $\hat{k}$ : Proportional controller gain.

**Acknowledgments**

The authors are thankful to the Royal Golden Jubilee Ph.D. Program under Grant PHD/0091/2551 for the research grants as well as some partial fundings available from Suranaree University of Technology.

## References

- [1] T. Watanabe, Y. Hashimoto, I. Nishikawa, and H. Tokumaru, "Line balancing using a genetic evolution model," *Control Engineering Practice*, vol. 3, no. 1, pp. 69–76, 1995.
- [2] C. Onnen, R. Babuška, U. Kaymak, J. M. Sousa, H. B. Verbruggen, and R. Isermann, "Genetic algorithms for optimization in predictive control," *Control Engineering Practice*, vol. 5, no. 10, pp. 1363–1372, 1997.
- [3] E. W. McGookin and D. J. Murray-Smith, "Submarine manoeuvring controller's optimisation using simulated annealing and genetic algorithms," *Control Engineering Practice*, vol. 14, no. 1, pp. 1–15, 2006.
- [4] M. Marinaki, Y. Marinakis, and G. E. Stavroulakis, "Fuzzy control optimized by PSO for vibration suppression of beams," *Control Engineering Practice*, vol. 18, no. 6, pp. 618–629, 2010.
- [5] J. G. Gray, D. J. Murray-Smith, Y. Li, K. C. Sharman, and T. Weinbrenner, "Nonlinear model structure identification using genetic programming," *Control Engineering Practice*, vol. 6, no. 11, pp. 1341–1352, 1998.
- [6] B. Abdelhadi, A. Benoudjit, and N. Nait-Said, "Application of genetic algorithm with a novel adaptive scheme for the identification of induction machine parameters," *IEEE Transactions on Energy Conversion*, vol. 20, no. 2, pp. 284–291, 2005.
- [7] F. Alonge, F. D'Ippolito, and F. M. Raimondi, "Least squares and genetic algorithms for parameter identification of induction motors," *Control Engineering Practice*, vol. 9, no. 6, pp. 647–657, 2001.
- [8] C. Zheng and P. Wang, "Parameter structure identification using tabu search and simulated annealing," *Advances in Water Resources*, vol. 19, no. 4, pp. 215–224, 1996.
- [9] T. Kulworawanichpong, K.-L. Areerak, K.-N. Areerak, and S. Sujitjorn, "Harmonic identification for active power filters via adaptive tabu search method," *Lecture Notes in Computer Science*, vol. 3215, part 3, pp. 687–694, 2004.
- [10] Y. Liu and X. He, "Modeling identification of power plant thermal process based on PSO algorithm," in *Proceedings of the American Control Conference (ACC 05)*, pp. 4484–4489, Portland, Ore, USA, June 2005.
- [11] J. Meier, W. Schaedler, L. Borgatti, A. Corsini, and T. Schanz, "Inverse parameter identification technique using PSO algorithm applied to geotechnical modeling," *Journal of Artificial Evolution and Applications*, vol. 2008, Article ID 574613, 14 pages, 2008.
- [12] V. Khanagha, A. Khanagha, and V. T. Vakili, "Modified particle swarm optimization for blind deconvolution and identification of multi channel FIR Filters," *Eurasip Journal on Advances in Signal Processing*, vol. 2008, Article ID 280635, 6 pages, 2010.
- [13] X. He and J. J. Liu, "Aquifer parameter identification with ant colony optimization algorithm," in *Proceedings of the International Workshop on Intelligent Systems and Applications (ISA '09)*, pp. 1–4, May 2009.
- [14] L. Liu, M. Fukumoto, S. Saiki, and S. Zhang, "A variable step-size proportionate affine projection algorithm for identification of sparse impulse response," *Eurasip Journal on Advances in Signal Processing*, vol. 2009, Article ID 150914, 10 pages, 2009.
- [15] H. Chen, Y. Zhu, and K. Hu, "Multi-colony bacteria foraging optimization with cell-to-cell communication for RFID network planning," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 539–547, 2010.
- [16] F. Glover, "Tabu search—part I," *ORSA Journal on Computing*, vol. 1, pp. 190–206, 1989.
- [17] F. Glover, "Tabu search—part II," *ORSA Journal on Computing*, vol. 2, pp. 4–32, 1990.
- [18] G. Zhang, W. Habenicht, and W. E. L. Spieß, "Improving the structure of deep frozen and chilled food chain with tabu search procedure," *Journal of Food Engineering*, vol. 60, no. 1, pp. 67–79, 2003.
- [19] T. Kulworawanichpong and S. Sujitjorn, "Optimal power flow using tabu search," *IEEE Power Engineering Review*, vol. 22, no. 6, pp. 37–40, 2002.
- [20] E. Nowicki and C. Smutnicki, "A fast tabu search algorithm for the permutation flow-shop problem," *European Journal of Operational Research*, vol. 91, no. 1, pp. 160–175, 1996.
- [21] R. Battiti and G. Tecchioli, "The reactive tabu search," *ORSA Journal Computing*, vol. 6, no. 2, pp. 126–140, 1994.
- [22] Y. A. Kochetov and E. N. Goncharov, "Probabilistic tabu search algorithm for the multi-stage uncapacitated facility location problem," in *Operations Research Proceedings*, pp. 65–70, Springer, 2001.

- [23] S. Sujitjorn, T. Kulworawanichpong, D. Puangdownreong, and K.-N. Areerak, "Adaptive tabu search and applications in engineering design," in *Integrated Intelligent Systems for Engineering Design*, X. F. Zha and R. J. Howlett, Eds., pp. 233–257, IOS Press, Amsterdam, The Netherlands, 2006.
- [24] S. Sujitjorn and S. Khwan-on, "Learning control via neuro-tabu-fuzzy controller," *Lecture Notes in Computer Science*, vol. 4251, pp. 833–840, 2006.
- [25] N. Sriyingyong and K. Attakitmongcol, "Wavelet-based audio watermarking using adaptive tabu search," in *Proceedings of the 1st International Symposium on Wireless Pervasive Computing*, pp. 1–5, January 2006.
- [26] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.
- [27] Y. Liu and K. M. Passino, "Biomimicry of social foraging bacteria for distributed optimization: models, principles, and emergent behaviors," *Journal of Optimization Theory and Applications*, vol. 115, no. 3, pp. 603–628, 2002.
- [28] S. Mishra, "A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 1, pp. 61–73, 2005.
- [29] S. Mishra and C. N. Bhende, "Bacterial foraging technique-based optimized active power filter for load compensation," *IEEE Transactions on Power Delivery*, vol. 22, no. 1, pp. 457–465, 2007.
- [30] M. Tripathy, S. Mishra, L. L. Lai, and Q. P. Zhang, "Transmission loss reduction based on FACTS and bacteria foraging algorithm," *Lecture Notes in Computer Science*, vol. 4193, pp. 222–231, 2006.
- [31] W. J. Tang, M. S. Li, Q. H. Wu, and J. R. Saunders, "Bacterial foraging algorithm for optimal power flow in dynamic environments," *IEEE Transactions on Circuits and Systems*, vol. 55, no. 8, pp. 2433–2442, 2008.
- [32] T. Datta, I. S. Misra, B. B. Mangaraj, and S. Imtiaj, "Improved adaptive bacteria foraging algorithm in optimization of antenna array for faster convergence," *Progress in Electromagnetics Research*, vol. 1, pp. 143–157, 2008.
- [33] R. Majhi, G. Panda, B. Majhi, and G. Sahoo, "Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques," *Expert Systems with Applications*, vol. 36, no. 6, pp. 10097–10104, 2009.
- [34] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, 2003.
- [35] E. Alba, *Parallel Metaheuristics*, Wiley-Interscience, New Jersey, NJ, USA, 2005.
- [36] E. G. Talbi, *Metaheuristics*, John Wiley & Sons, New Jersey, NJ, USA, 2009.
- [37] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, 1975.
- [38] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.
- [39] MathWorks, "Genetic Algorithm and Direct Search Toolbox: for Use with MATLAB," User's Guide, Version 1, MathWorks, Natick, Mass, USA, 2005.
- [40] B. Armstrong-Hélouvry, "Stick slip and control in low-speed motion," *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 38, no. 10, pp. 1483–1496, 1993.
- [41] B. Armstrong-Hélouvry, P. Dupont, and C. Cadudas de Wit, "A survey of model, analysis tools and compensation methods for the control of machines with friction," *Automatica*, vol. 30, no. 7, pp. 1083–1138, 1994.
- [42] C. Canudas de Wit, H. Olsson, K. J. Åström, and P. Lischinsky, "A new model for control of systems with friction," *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 40, no. 3, pp. 419–425, 1995.
- [43] H. Du and S. S. Nair, "Modeling and compensation of low-velocity friction with bounds," *IEEE Transactions on Control Systems Technology*, vol. 7, no. 1, pp. 110–121, 1999.