Adaptive Backtracking Search Algorithm for Induction Magnetometer Optimization

Haibin Duan[1,2] and Qinan Luo[1]

[1]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China
[2]Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

Backtracking search algorithm (BSA) is a novel evolutionary algorithm (EA) for solving real-valued numerical optimization problems. In this paper, an adaptive BSA (ABSA) is proposed to solve the optimization problem of an induction magnetometer (IM). In the adaptive algorithm, the probabilities of crossover and mutation are varied depending on the fitness values of the solutions to refine the convergence performance. The proposed ABSA will also be compared with basic BSA and other widely used EA algorithms. Simulation results show that ABSA is better able to solving the IM optimization problems.

Index Terms—**Backtracking search algorithm (BSA), evolutionary computation, magnetometers, optimization.**

I. Introduction

EVOLUTIONARY algorithms (EA) [1], including covariance matrix adaptation evolution strategy [2], differential evolution (DE) algorithm [3], particle swarm optimization (PSO) [4], artificial bee colony (ABC) [5] algorithm, have been widely used for electromagnetic design problems [6]. Unlike classical optimization techniques, EAs are sufficiently flexible to solve different types of problems in virtue of their global exploration and local exploitation abilities. EA techniques are typically used to solve non-linear and non-differentiable problems. EAs radically differ from one another based on their strategies for generating and updating trial individuals. Different strategies have a considerable effect on problem-solving success and speed.

By following the inspirational previous works, many bioinspired evolutionary computation methods have been proposed, developed, and studied for scientific research and engineering applications. Backtracking search algorithm (BSA) is a novel EA, which was first proposed by Civicioglu [7] in 2013. This algorithm consists of three basic genetic operators, which are selection, mutation, and crossover to generate trial individuals. In contrast to many genetic algorithms, such as DE and its derivatives, BSAs random mutation strategy, and a non-uniform crossover strategy, enable it to solve numerical optimization problems successfully and rapidly.

BSA uses randomly generated populations in calculating the search-direction matrix, while PSO and ABC do not use previous generation populations. BSA is a dual-population algorithm, and the historical populations that include more efficient individuals are well used based on a randomly selected previous generation. More advanced generations relative to the historical populations are generated by BSAs crossover strategy. This facilitates BSAs generation of more efficient trial individuals.

Induction magnetometer (IM) is an effective tool for acquiring magnetic field data, which are required with a wide bandwidth [8]. The lengths and the weights of existing IMs are inconvenient for geophysicists to use in the wild. In this paper, we mainly focus on the optimization problem of reducing the mass and dimension of IMs.

This paper proposes an adaptive BSA to optimize the design parameters of an IM to minimize its weight. The adaptive strategy is employed to reinforce the convergence performance of BSA. It is observed that as the BSA converges, the fitness distance between each population will become smaller. Therefore, it would be sensible to incorporate the fitness distances between individuals into the adaptive updating of the probability of mutation.

The rest of this paper is organized as follows. Section II introduces the basic principles of BSA and the adaptive mechanism. The optimization problem of an IM is presented in Section III. The comparative experimental results are given in Section IV, which are carried out to compare the adaptive BSA (ABSA) with the BSA, PSO, ABC, and DE. Our concluding remarks are contained in Section V.

II. Adaptive BSA

A. BSA

As presented in [7], BSA is a population-based iterative EA designed to be a global minimizer, including five processes. BSA uses three basic genetic operators (selection, mutation, and crossover) to generate trial individuals. BSA has a random mutation strategy that uses only one direction individual for each target individual and randomly chooses the direction individual from individuals of a randomly chosen previous generation. BSA uses a non-uniform crossover strategy that is much more complex than traditional crossover strategies. The processes of BSA can be described as follows.

First, BSA initializes the population denoted as $P_{ij} \sim U(\mathrm{low}_j, \mathrm{up}_j)$, for $i = 1, 2, 3 \ldots, N$ and $j = 1, 2, 3, \ldots, D$, where N and D are the population size and the problem

Manuscript received December 9, 2013; revised January 22, 2014, March 25, 2014, and June 4, 2014; accepted July 20, 2014. Date of publication July 23, 2014; date of current version December 12, 2014. Corresponding author: H. Duan (e-mail: hbduan@buaa.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMAG.2014.2342192

dimension, respectively. Each dimension signifies one design parameter.

Then, BSAs Selection-I stage evaluates the population and determines the historical population oldP according to the obtained fitness value

$$\text{if } a < b \text{ then oldP} := P \,|a, b \sim U(0,1) \tag{1}$$

where := is the update operation. Equation (1) ensures that BSA designates a population belonging to a randomly selected previous generation as the historical population and remembers this historical population until it is changed.

During each generation, BSAs mutation process generates the initial form of the trial population using

$$\text{Mutant} = P + F \cdot (\text{oldP} - P) \tag{2}$$

where F is the amplitude control factor that controls the amplitude of the search-direction matrix. Because the historical population is used in the calculation of the search-direction matrix, the trial population is generated by taking partial advantage of its experiences from previous generations.

After the new mutant operation is finished, the crossover process generates the final form of the trial population T. The initial value of the trial population is Mutant, which has been set in the mutation process. Individuals with better fitness values for the optimization problem are used to evolve the target population. The first step of the crossover process calculates a binary integer-valued matrix (H) of size $N \cdot D$ that indicates the individuals of T to be manipulated using the relevant individuals of P. Then, the trial population T is updated as

$$\begin{aligned} H_{i,u(1:\lceil M\cdot \text{rnd}\cdot D\rceil)} &= 0 \,|u = \text{permuting}((1,2,3,\ldots,D)) \\ T_{n,m} &:= P_{n,m} \\ \text{if } H_{n,m} &= 1 \end{aligned} \tag{3}$$

where $n \in \{1, 2, 3, \ldots, N\}$ and $m \in \{1, 2, 3, \ldots, D\}$.

Two predefined strategies are randomly used in defining the integer-valued matrix, which is more complex than the processes used in DE. The first strategy uses mix rate M, and the other allows only one randomly chosen individual to mutate in each trial.

After one generation is finished, in BSAs Selection-II stage, the global minimizer is updated based on the best individual of T. The iteration goes until terminal requirement is met. Then, the global minimizer is the output as the optimal solution to the problem.

B. Adaptive BSA

The mutation operation in BSA introduces occasional changes of a random individual position with a specified mutation probability. The crossover operator, which is quite different from the crossover strategies used in other EAs, uses the mix rate parameter to control the number of elements of individuals that will mutate in a trial. However, the significance of amplitude control factor F and mix rate M in controlling BSA performance has not been acknowledged in BSA research.

In this paper, an adaptive mechanism is introduced to improve the performance of BSA by utilizing the global information and further improve the convergence performance of BSA [9]. The key idea of the ABSA is adapting the amplitude control factor F and mix rate M based on the fitness statistics of population at each generation.

The proposed algorithm is based on the existing adaptive strategy [3], but introduces a normalized fitness distance between the current individual and other individuals in the population to control the probability of mutation. It has been observed that the difference between the average fitness value and minimum fitness value of the population $f - f_{\min}$ is likely to be less for a population that has converged to optimum solution than that for a population scattered in the solution space. Therefore, the mix rate would be sensible to incorporate the fitness distances between individuals into the adaptive update principle. The values of F and M should be varied depending on the value of $f - f_{\min}$. The adaptive strategy for updating F and M can be described by the following:

$$M = \begin{cases} k_1(f_c - f_{\min})/(\bar{f} - f_{\min}), & f_c > \bar{f} \\ k_2, & f_c \le \bar{f} \end{cases} \tag{4}$$

$$F = \begin{cases} k_3(f_i - f_{\min})/(\bar{f} - f_{\min}), & f_i > \bar{f} \\ k_4, & f_i \le \bar{f} \end{cases} \tag{5}$$

where k_1, k_2, k_3, and k_4 have to be less than 1.0 to constrain F and M to the range of 0.0–1.0, f_c is the larger of the fitness values of the individuals selected for crossover, and f_i is the fitness of the ith individual to which the mutation is applied. In the basic BSA, the mix rate M and amplitude control factor F are generally set as 0.9 and 1.0, according to the range of M and F, k_1, k_2, k_3, and k_4 are set as 0.9, 0.9, 1.0, and 1.0.

In (4) and (5), high-fitness solutions are protected, while solutions with subaverage fitnesses are totally disrupted. This adaptive strategy can speed up the convergence rate of BSA.

The detailed processes of ABSA are described as follows.

Step 1: Initialization. Number of population size and maximum number of iterations are, respectively, assigned as N and $Nc_{\max}$. The initial population is randomly generated.

Step 2: Evaluate the population and conduct Selection-I to determines the historical population.

Step 3: Conduct the mutation operation and generate new trial population based on the search-direction matrix.

Step 4: Conduct crossover operation and select the best individual among the group consisting of the historical population, the newly generated trial population, and the two individuals produced by the crossover operation. Replace the historical population with the selected one.

Step 5: Evaluate the final trial population, and conduct Selection-II to update the global minimizer.

Step 6: Conduct adaptive strategy, update F and M according to (4) and (5).

Step 7: If the current number of iterations Nc is less than $Nc_{\max}$, go back to Step 3. Otherwise the algorithm is terminated and the global minimizer is the output as the solution to the optimization problem.

The flow chart of ABSA is shown in Fig. 1.

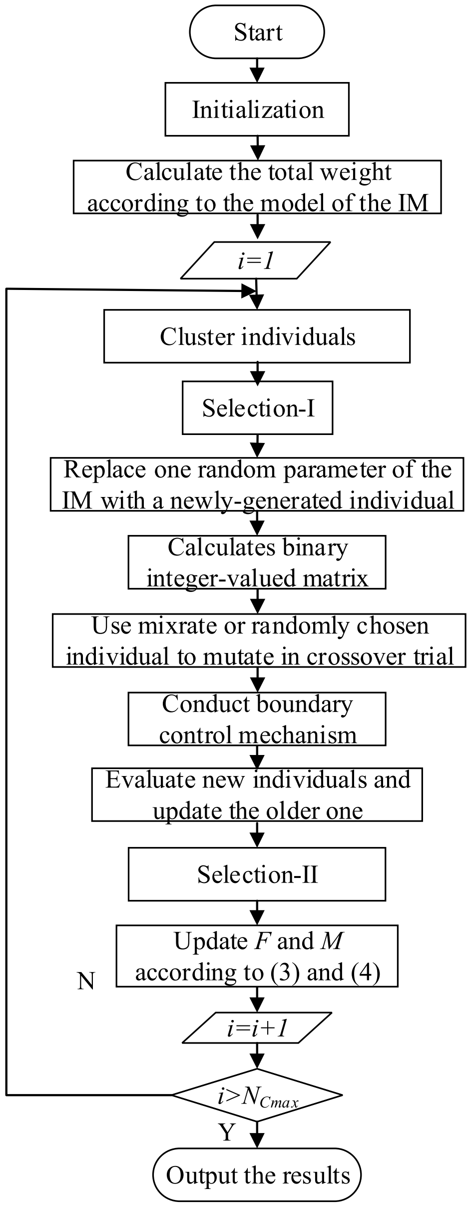

Fig. 1. Detailed flow chart of ABSA.

Numerical tests are conducted to examine the relative success of ABSA and the comparison algorithms in solving the numerical optimization. The effectiveness of these algorithms is investigated here using eight benchmark functions, which are shown in Table I. The basic statistics of the 30-solutions obtained by ABSA and the other algorithms are presented in Table II.

The results obtained from the tests reveal that our proposed ABSA is statistically more successful than all the other comparative algorithms.

TABLE I

BENCHMARK PROBLEMS USED IN TEST (DIM: DIMENSION, LOW, UP: LIMITS OF SEARCH SPACE, M: MULTIMODAL, N: NON-SEPARABLE, U: UNIMODAL, S: SEPARABLE, E: EXPANDED, H: HYBRID, C: COMPOSITION)

Problem	Name	Type	Low	Up	Dim
F1	Foxholes	MS	-65.536	65.536	2
F2	Penalized	MN	-50	50	30
F3	Beale	UN	-4.5	4.5	5
F4	Sphere2	US	-100	100	30
F5	Shifted sphere	U	-100	100	10
F6	Shifted Rosenbrock's	M	-100	100	10
F7	Expanded rotated extended Scaffes	E	-100	100	10
F8	Rotated hybrid comp. Fn 1 with noise	HC	-5	5	10

TABLE II

BASIC STATISTICS OF THE 30-SOLUTIONS OBTAINED BY PSO, ABC, DE, AND ABSA IN TEST (MEAN: MEAN-SOLUTION, STD: STANDARD-DEVIATION OF MEAN-SOLUTION, BEST: THE BEST-SOLUTION, AND RUNTIME: MEAN-RUN TIME IN SECONDS)

Problem	Statistics	PSO	ABC	DE	ABSA
F1	Mean	1.3316	0.9980	1.0641	**0.9980**
	Std	0.9455	0.0000	0.3622	**0.0000**
	Best	0.9980	0.9980	0.9980	**0.9980**
	Runtime	72.527	64.976	51.101	**35.846**
F2	Mean	0.1278	0.0000	0.0034	**0.0000**
	Std	0.2772	0.0000	0.0189	**0.0000**
	Best	0.0000	0.0000	0.0000	**0.0000**
	Runtime	139.555	84.416	**9.492**	16.462
F3	Mean	0.0000	0.0000	0.0000	**0.0000**
	Std	0.0000	0.0000	0.0000	**0.0000**
	Best	0.0000	0.0000	0.0000	**0.0000**
	Runtime	32.409	22.367	1.279	**0.846**
F4	Mean	0.0000	0.0000	0.0000	**0.0000**
	Std	0.0000	0.0000	0.0000	**0.0000**
	Best	0.0000	0.0000	0.0000	**0.0000**
	Runtime	159.904	21.924	**1.424**	2.846
F5	Mean	-450.0000	-450.0000	-450.0000	**-450.0000**
	Std	0.0000	0.0000	0.0000	**0.0000**
	Best	-450.0000	-450.0000	-450.0000	**-450.0000**
	Runtime	212.862	113.623	118.477	**108.496**
F6	Mean	393.4960	391.2531	231.3986	**390.1328**
	Std	16.0224	3.7254	247.2968	**0.0000**
	Best	390.0000	390.0101	-140.0000	**390.0000**
	Runtime	1178.079	159.762	**153.715**	284.646
F7	Mean	**-298.2836**	-296.9323	-296.8840	-297.5360
	Std	0.5587	**0.2251**	0.4330	0.4085
	Best	**-299.6022**	-297.4660	-297.8412	-298.3869
	Runtime	2517.138	**262.533**	334.888	1835.449
F8	Mean	217.3338	265.0370	228.7309	**228.3770**
	Std	20.6686	12.4034	12.3683	**8.7087**
	Best	120.0000	241.9810	181.6800	**204.6479**
	Runtime	8208.697	**2159.392**	5873.112	7166.546

III. PROBLEM FORMULATION

In this paper, we focus on the optimization of IM to achieve the required performance with the least mass. To analyze the optimization, some reasonable hypotheses are made. The apparent permeability of an IM is determined by the size of the core (d and l) and the initial permeability of the core material due to the demagnetization effect. As is shown in Table I, this optimization problem has four design variables and one objective variable, and our object is to minimize the weight cost.

The measuring principle of the IM is based on Faraday's law, and the induction voltage (e) of the coil can be expressed as

$$e(t) = -N\frac{d\Phi(t)}{dt} = -\mu_{\text{app}} NS \frac{dB(t)}{dt} \tag{6}$$

where Φ is the magnetic flux in the core, μ_{app} is the apparent permeability of the core, N is the turns of the coil, S is the core cross section, and B is the magnetic flux density.

In the first place, the demagnetizing factor N_B can be calculated as [10]

$$N_B = \frac{1}{\frac{l^2}{d^2}-1}\left\{\frac{\frac{l}{d}}{2\left(\frac{l^2}{d^2}-1\right)^{\frac{1}{2}}}\ln\left(\frac{\frac{l}{d}+\sqrt{\frac{l^2}{d^2}-1}}{\frac{l}{d}-\sqrt{\frac{l^2}{d^2}-1}}\right)-1\right\}. \tag{7}$$

Then, the apparent permeability of the core μ_{app} can be expressed as [11]

$$\mu_{\text{app}} = \frac{\mu_r}{1+N_B(\mu_r-1)}. \tag{8}$$

The effective core cross section S represents the area of the core material without the adhesive gap, which can be expressed as

$$S = \pi \times \left(\frac{d}{2}\right)^2 \times \eta. \tag{9}$$

The relation between the resistance and the self-inductance of the coils can be evaluated using the parameters listed in Table III [12]

$$L_{\text{pc}} = \frac{N^2\mu_0\mu_{\text{app}} \times \pi \times \left(\frac{d}{2}\right)^2}{l}\eta \tag{10}$$

$$R_{\text{sc}} = \frac{4\rho N_t}{d_w^2}\left[\frac{N_t \times (d_w+t_w)^2}{\kappa l} + d + 2t_{\text{coil}}\right]. \tag{11}$$

The equivalent input magnetic noise level is a function of variables d, l, d_w, and N_t

$$B_{\text{nte}}(d,l,d_w,N_t) = \frac{\sqrt{e_w^2 + (i_w \times |R_{\text{sc}} + j2\pi L_{\text{pc}}|)^2 + 4K_bT_cR_{\text{sc}}}}{|2\pi\,\mu_{\text{app}}N_tS|}. \tag{12}$$

The total weight of the IM mainly consists of three parts: the core W_{core}, the coil W_{coil}, and the package W_{package}

$$W_{\text{total}}(d,l,d_w,N_t) = W_{\text{core}} + W_{\text{coil}} + W_{\text{package}}. \tag{13}$$

The weight of package is supposed to be 1 kg and the other two parts can be evaluated as

$$W_{\text{core}} = \rho_c\pi\left(\frac{d}{2}\right)^2 l\eta \tag{14}$$

$$W_{\text{coil}} = \frac{1}{2}\rho_w N_t \pi^2 d_w^2\left[\frac{N_t(d_w+t_w)^2}{2\kappa l} + \frac{d}{2} + t_{\text{coil}}\right]. \tag{15}$$

The geometrical parameters of the IM are shown in Fig. 2.

Based on these analyses, the optimization problem of IM weight can be expressed as the following general optimization problem:

$$\begin{aligned} &\text{maximize } W_{\text{total}} = f(d,l,d_w,N_t) \\ &\text{with } \text{B}_{\text{nte}}(d,l,d_w,N_t) = \text{Nte}. \end{aligned} \tag{16}$$

TABLE III

OPTIMIZATION PARAMETERS

	Variables	Description	Value
Objective variable	W_{total}	The total weight of the IM	-
Design variables	N_t	Number of turns	[0,100000]
	d_w (m)	Diameter of the winding wire	[0,1]
	d(m)	Diameter of the core	[0,1]
	l(m)	Length of the core	[0,1]
Fixed variables	η	Ratio of the weight of the magnetic material to the weight of the core	0.9
	t_{coil} (mm)	Thickness of the bobbin tube	1.5
	κ	Ratio of the length of the coil to the length of the core	0.9
	t_w (mm)	Thickness of the enamel coating of the wire	0.03
	μ_0 (V·s/(A·m))	Vacuum permeability	1.257e-6
	e_w (nV/√Hz)	Voltage noise level of the circuit	3
	i_w (pA/√Hz)	Current noise level of the circuit	0.4
	K_b ($m^2kg\ s^{-2}K^{-1}$)	Boltzmann constant	1.3806503e-23
	T_c (K)	Absolute temperature	300
	ρ_w (kg/m^3)	Density of the wire	8.9e3
	ρ (Ω·m)	Resistivity of the wire	1.75e-8
	$w_{package}$ (kg)	Weight of the wire	1
	δ	Admissible relative error for frequency response of the IM	1%
	H_{max} (nT)	Maximum geomagnetic field	55000

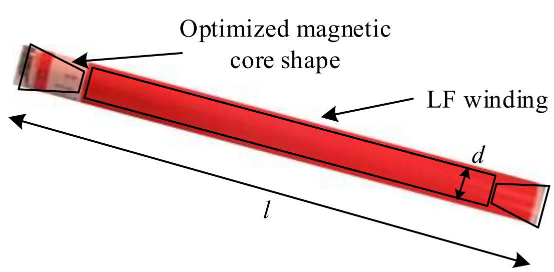

Fig. 2. Geometrical parameters of the IM.

IV. EXPERIMENTAL RESULTS

To investigate the feasibility and effectiveness of the ABSA for solving the optimization of the IM, a series of comparative simulations is conducted. The control parameters of BSA and ABSA are shown in Tables IV and V, respectively. The parameters of other EAs, including PSO, ABC, and DE, are adopted from [13]–[16],

TABLE IV
CONTROL PARAMETERS OF BSA

Parameter	Description	Value
m	Number of population	30
Nc_{max}	Maximum times of iteration	20
M	Mix rate	0.9
F	Amplitude control factor	1

TABLE V
CONTROL PARAMETERS OF ABSA

Parameter	Description	Value
m	Number of population	30
Nc_{max}	Maximum times of iteration	20
M	Mix rate	[0, 1]
F	Amplitude control factor	[0, 1]

TABLE VI
CONTROL PARAMETERS OF PSO

Parameter	Description	Value
m	Number of particles	30
Nc_{max}	Maximum times of iteration	20
c_1	Self best factor	2
c_2	Global best factor	2

TABLE VII
CONTROL PARAMETERS OF ABC

Parameter	Description	Value
m	Number of particles	30
Nc_{max}	Maximum times of iteration	20
n	Number of food sources	15
l	Food sources trials limit	5

TABLE VIII
CONTROL PARAMETERS OF DE

Parameter	Description	Value
m	Number of particles	30
Nc_{max}	Maximum times of iteration	20
F	Differential weight	0.7
CR	Crossover probability	0.9

TABLE IX
STATISTIC SIMULATION RESULTS

Methods	W_{total}		CPU time(s)	
	average	standard deviation	average	standard deviation
ABSA	1.4771	6.7×10^{-5}	1.92	0.13
BSA	1.4780	4.5×10^{-5}	1.55	0.11
PSO	1.4782	4.7×10^{-3}	2.76	0.93
ABC	1.4787	7.4×10^{-4}	3.61	0.24
DE	1.4794	5.4×10^{-5}	2.24	0.18

TABLE X
BEST SOLUTIONS OF ABSA, BSA, PSO, ABC, AND DE

	W_{total}	d_w (m)	d (m)	l (m)	N_t	CPU time (s)
ABSA	1.4769	0.0004	0.14 e-3	0.9	70358	1.86
BSA	1.4777	0.0004	0.15 e-3	0.8	70360	1.48
PSO	1.4774	0.0004	0.14 e-3	0.9	70350	2.64
ABC	1.4782	0.0004	0.14 e-3	0.8	70344	3.45
DE	1.4791	0.0004	0.13 e-3	0.9	70254	2.12

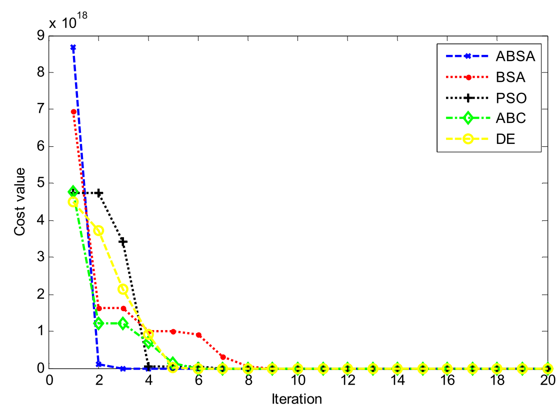

Fig. 3. Evolutionary process comparison of ABSA, BSA, PSO, ABC, and DE ($Nc = 20$).

which are shown in Tables VI–VIII. The population size and maximum number of iterations of the comparison EAs are set equal to those of BSA and ABSA.

Comparison of the simulation results between the ABSA and the basic BSA is illustrated in Tables IX and X. The statistic result of 50 trials is presented in Table IX. Table X shows the best solutions of the BSA and other three EAs.

From Table IX, we can conclude that the average value and stability of ABSA are slightly better than PSO, ABC, and DE. Among the five, BSA has the lowest standard deviation value, and ABSA is very close to it. PSO, ABC, and DE are not likely to precisely find the optimal solution. As a result, ABSA, which can reliably find solutions close enough to optimum, is a feasible approach to solve the IM optimization problem.

The first row in Table X presents the global optimum obtained by these comparison algorithms. All the algorithms are well capable of obtaining an appreciable approximation of the global optimum. When the results are examined, ABSA and PSO are statistically identical, and ABSA has provided statistically better solutions than the other comparison algorithms. In addition, we can conclude that the optimal value of ABSA is slightly better than the other algorithms. The simulation results also demonstrate that BSA is very likely to trap in local optima. Moreover, an examination of the data obtained from the tests shows that ABSA is generally faster than most of the comparison algorithms. As a result, ABSA, which can reliably find a solution close enough to optimum, is a feasible approach to the optimization problem of IM.

Fig. 3 shows the evolution curves of all the above algorithms. It can be observed that, using the adaptive strategy, the iterations for convergence can be reduced greatly, and the

TABLE XI
SIMULATION RESULTS FOR A BRUSHLESS DC WHEEL MOTOR OPTIMIZATION

Methods	Efficiency		CPU time(s)	
	average	standard deviation	average	standard deviation
PSO	94.94%	4.7×10^{-3}	3.06	0.93
ABC	95.26%	5.4×10^{-5}	3.87	**0.44**
DE	95.24%	6.4×10^{-4}	**2.86**	0.34
ABSA	**95.32%**	$\mathbf{4.7\times10^{-5}}$	3.81	0.51

TABLE XII
BEST SOLUTIONS OF PSO, ABC, DE, AND ABSA FOR A BRUSHLESS DC WHEEL MOTOR OPTIMIZATION

	η(%)	D_s (mm)	B_d (T)	δ (A/mm^2)	B_e (T)	B_{cs} (T)	CPU time (s)
PSO	95.18	150.0	1.8	2.0	0.572	1.6	5.91
ABC	95.28	201.3	1.8	2.070	0.648	0.876	3.73
DE	95.24	201.3	1.8	2068	0.648	0.881	**2.54**
ABSA	**95.32**	**201.3**	**1.8**	**2.057**	**0.649**	**0.884**	3.71

curve of ABSA is always above those of BSA, PSO, ABC, and DE.

The optimization for an analytical model of a brushless dc wheel motor has also been chosen as another benchmark for further test [17]. This benchmark problem has five design variables and one efficiency value. The efficiency is determined by these variables, and the object is to maximize the efficiency. The simulation results are shown in Tables XI and XII.

From the simulation results, one can conclude that both these evolution algorithms can obtain an appreciable approximation of the global optimum, yet the average value and stability of ABSA are better than those of the others.

V. CONCLUSION

BSA is a newly-proposed evolutionary optimization global search algorithm, which has a clear structure that enables it to benefit from previous generation populations as it searches for solutions with better fitness values. In each generation, BSA produces very efficient trial populations, because the mutation produces both large amplitude values essential for a global search and the small amplitude values necessary for a local search, and the complex crossover operation ensures creation of new trial individuals. BSA has already proven competent in solving some constrained benchmark problems.

This paper developed an adaptive BSA with an adaptive strategy for determination of the global or near-global optimum solution of an electromagnetic optimization problem. The amplitude control factor and mix rate are varied depending on the fitness values of the solutions. Simulation results on optimizing the IM variables verified the feasibility and effectiveness of our proposed ABSA in comparison with basic BSA, PSO, ABC, and DE algorithms. ABSA can successfully solve the complicated optimization problems and exhibit better convergence, which is especially promising as an EA.

ACKNOWLEDGMENT

This work was supported in part by the National Key Basic Research Program of China (973 Project) under Grant 2014CB046401 and Grant 2013CB035503; in part by the Natural Science Foundation of China under Grant 61333004, Grant 61273054, and Grant 60975072; in part by the Top-Notch Young Talents Program of China and National Magnetic Confinement Fusion Research Program of China under Grant 2012GB102006; and in part by the Aeronautical Foundation of China under Grant 20135851042.

REFERENCES

[1] K. Deb, A. Pratap, and S. Agarwal, “A fast and elitist multiobjective genetic algorithm: Nsga-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[2] C. Igel, N. Hansen, and S. Roth, “Covariance matrix adaptation for multi-objective optimization,” *Evol. Comput.*, vol. 15, no. 1, pp. 1–28, Mar. 2007.
[3] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.
[4] U. Baumgartner, C. Magele, and W. Renhart, “Pareto optimality and particle swarm optimization,” *IEEE Trans. Magn.*, vol. 40, no. 2, pp. 1172–1175, Mar. 2004.
[5] L. S. dos Coelho and P. Alotto, “Gaussian artificial bee colony algorithm approach applied to Loney’s solenoid benchmark problem,” *IEEE Trans. Magn.*, vol. 47, no. 5, pp. 1326–1329, May 2011.
[6] H. Duan, S. Li, and Y. Shi, “Predator-prey brain storm optimization for DC brushless motor,” *IEEE Trans. Magn.*, vol. 49, no. 10, pp. 5336–5340, Oct. 2013.
[7] P. Civicioglu, “Backtracking search optimization algorithm for numerical optimization problems,” *Appl. Math. Comput.*, vol. 219, no. 15, pp. 8121–8144, Apr. 2013.
[8] B. Yan, W. Zhu, L. Liu, K. Liu, and G. Fang, “An optimization method for induction magnetometer of 0.1 mHz to 1 kHz,” *IEEE Trans. Magn.*, vol. 49, no. 10, pp. 5294–5300, Oct. 2013.
[9] J. Zhang and A. C. Sanderson, “JADE: Adaptive differential evolution with optional external archive,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.
[10] J. A. Osborn, “Demagnetizing factors of the general ellipsoid,” *Phys. Rev.*, vol. 67, nos. 11–12, pp. 351–357, Jun. 1945.
[11] S. Tumanski, “Induction coil sensors—A review,” *Meas. Sci. Technol.*, vol. 18, no. 3, pp. 31–46, Jan. 2007.
[12] C. Coillot, J. Moutoussamy, P. Leroy, G. Chanteur, and A. Roux, “Improvements of the design of search coil magnetometer for space experiments,” *Sensor Lett.*, vol. 5, no. 1, pp. 1–4, Mar. 2007.
[13] J. K. Kishore, L. M. Patnaik, V. Mani, and V. K. Agrawal, “Application of genetic programming for multicategory pattern classification,” *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 242–258, Sep. 2000.
[14] G. Crevecoeur, P. Sergeant, L. Dupré, and R. Van de Walle, “A two-level genetic algorithm for electromagnetic optimization,” *IEEE Trans. Magn.*, vol. 46, no. 7, pp. 2585–2595, Jul. 2010.
[15] C. Xu, H. Duan, and F. Liu, “Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning,” *Aerosp. Sci. Technol.*, vol. 14, no. 8, pp. 535–541, Dec. 2010.
[16] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, “Comprehensive learning particle swarm optimizer for global optimization of multimodal functions,” *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.
[17] S. Brisset and P. Brochet, “Analytical model for the optimal design of a brushless DC wheel motor,” *COMPEL, Int. J. Comput. Math. Elect. Electron. Eng.*, vol. 24, no. 3, pp. 829–848, 2005.
[18] F. Moussouni and S. Brisset. (2008, Jul.). *A Benchmark for a Mono and Multi Objective Optimization of the Brushless DC Wheel Motor*. [Online]. Available: http://l2ep.univ-lille1.fr/come/benchmark-wheelmotor.htm