ARTICLE IN PRESS

Information Sciences xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Information Sciences

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

A modified Artificial Bee Colony algorithm for real-parameter optimization

Bahriye Akay *, Dervis Karaboga

Department of Computer Engineering, Erciyes University, 38039 Melikgazi, Kayseri, Turkey

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Swarm intelligence
Self-organization
Artificial Bee Colony algorithm
Real-parameter optimization

ABSTRACT

Swarm intelligence is a research field that models the collective intelligence in swarms of insects or animals. Many algorithms that simulates these models have been proposed in order to solve a wide range of problems. The Artificial Bee Colony algorithm is one of the most recent swarm intelligence based algorithms which simulates the foraging behaviour of honey bee colonies. In this work, modified versions of the Artificial Bee Colony algorithm are introduced and applied for efficiently solving real-parameter optimization problems.

1. Introduction

The collective intelligent behaviour of insect or animal groups in nature such as flocks of birds, colonies of ants, schools of fish, swarms of bees, and termites have attracted the attention of researchers. The aggregate behaviour of insects or animals is called swarm behaviour. Entomologists have studied this collective behaviour to model biological swarms, and engineers applied these models as a framework for solving complex real-world problems. This branch of artificial intelligence which deals with the collective behaviour of swarms through complex interaction of individuals without supervision, is referred to as swarm intelligence. Bonabeau defined swarm intelligence as “any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of the social insect colonies and other animal societies” [8]. Swarm intelligence has some advantages such as scalability, fault tolerance, adaptation, speed, modularity, autonomy, and parallelism [29].

The key components of swarm intelligence are self-organization and division of labour. In a self-organising system, each of the covered units may respond to local stimuli individually and act together to accomplish a global task via division of labour without a centralized supervision. The entire system can adapt to internal and external changes efficiently. Bonabeau et al. have characterized four basic properties on which self-organization relies: positive feedback, negative feedback, fluctuations and multiple interactions [8]. Positive feedback means that an individual recruits other individuals by some directive, such as dancing of bees in order to lead some other bees onto a specific food source site. Negative feedback avoids all individuals accumulating on the same task by counterbalancing the attraction negatively, such as abandoning the exhausted food source. Fluctuations are random behaviours of individuals in order to explore new states, such as random flights of scouts in a bee swarm. Multiple interactions are the basis of the tasks to be carried out by certain rules.

Bee swarms exhibit many intelligent behaviours in their tasks such as nest site building, marriage, foraging, navigation and task selection. There is an efficient task selection mechanism in a bee swarm that can be adaptively changed by the state of the hive and the environment. Foraging is another crucial task for bees. Forage selection depends on recruitment for and abandonment of food sources. There are three types of bees associated with the foraging task with respect to their selection mechanisms. Employed bees fly onto the sources which they are exploiting; onlooker bees choose the sources by watching

* Corresponding author. Tel.: +90 352 437 49 01x32578; fax: +90 352 437 57 84.
E-mail addresses: bahriye@erciyes.edu.tr (B. Akay), karaboga@erciyes.edu.tr (D. Karaboga).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.07.015

the dances performed by employed bees, and scouts choose sources randomly by means of some internal motivation or possible external clue. The exchange of information among bees is the most important occurrence in the formation of the collective knowledge. The most important part of the hive in terms of exchanging information is the dancing area. Communication among bees related to the quality of food sources takes place in the dancing area. Various dances are performed on the dancing area, such as waggle, round, tremble depending on the distance of the discovered source.

Although the Ant Colony Optimization (ACO) algorithm [11] simulating the behaviour of ant colonies and Particle Swarm Optimization (PSO) algorithm [30] mimicking flocks of birds are the most popular intelligence based optimization algorithms, there are some algorithms presented in the literature based on the foraging behaviour of a bee swarm [55,36,57,58,18,42,15,12,9,3,46,35,31]. Tereshko developed a model of foraging behaviour of a honeybee colony based on reaction–diffusion equations [55]. Lucic and Teodorovic developed the Bee System based on the foraging behaviour of a bee colony for solving difficult combinatorial optimization problems [36]. Another algorithm is BeeAdHoc proposed by Wedde and Farooq, which is a routing algorithm for energy efficient routing in mobile ad hoc networks [57]. BeeAdHoc is also inspired by the foraging principles of honey bees. Yang presented a virtual bee algorithm (VBA) to solve numerical optimization problems [58]. Karaboga introduced a bee swarm algorithm called the Artificial Bee Colony (ABC) algorithm that simulates the foraging behaviour of bees [18] for multimodal and multi-dimensional numerical optimization problems. Pham et al. also described the Bees Algorithm which mimics the foraging behaviour of honey bees [42]. Ghosh and Marshall proposed a model of learning and collective decision-making in honey bees engaged in foraging [15]. They employed their model for a swarm of robots. Drias and Yahi introduced a meta-heuristic named Bees Swarm Optimization (BSO) based on the behaviour of real bees for solving maximum weight satisfiability problems [12]. Chong et al. described a bee colony optimization algorithm based on the foraging bahaviour and the waggle dance. The algorithm was applied to job shop scheduling [9]. Baig and Rashid presented Honey Bee Foraging (HBF) algorithm which simulates the foraging behaviour of the honey bees and performs swarm-based collective foraging in promising neighborhoods with individual scouting searches in other areas [3]. Quijano and Passino introduced a foraging model of honey bees for solving a class of optimal resource allocation problems [46]. Lu and Zhou developed Bee Collecting Pollen Algorithm (BCPA) by simulating the honeybees' pollen collecting behaviour for solving the travelling salesman problem [35]. Ko et al. proposed a self-adaptive grid computing protocol called HoneyAdapt which is based on adaptive bee foraging behaviour in nature [31].

In this work, some modifications to the standard ABC algorithm are introduced, and the performance of the modified ABC algorithm is investigated for real-parameter optimization on both basic and composite functions presented at the Congress of Evolutionary Computation 2005 (CEC05). Effects of the perturbation rate that controls the frequency of parameter change, the scaling factor (step size) that determines the magnitude of change in parameters while producing a neighboring solution, and the "*limit*" parameter on the performance of the ABC algorithm are investigated on real-parameter optimization.

The rest of the paper is organized as follows. In Section 2, the ABC Algorithm is described. In Section 3, the works related to the ABC algorithm are summarized and then the modifications to the basic ABC algorithm are introduced in Section 4. In Section 5, experiments are presented and the results are discussed and in Section 6, a thorough comparative analysis including the algorithms considered in this study is presented.

2. Artificial Bee Colony algorithm

In a real bee colony, some tasks are performed by specialized individuals. These specialized bees try to maximize the nectar amount stored in the hive using efficient division of labour and self-organization. The Artificial Bee Colony (ABC) algorithm, proposed by Karaboga in 2005 for real-parameter optimization, is a recently introduced optimization algorithm which simulates the foraging behaviour of a bee colony [18]. The minimal model of swarm-intelligent forage selection in a honey bee colony which the ABC algorithm simulates consists of three kinds of bees: employed bees, onlooker bees and scout bees. Half of the colony consists of employed bees, and the other half includes onlooker bees. Employed bees are responsible for exploiting the nectar sources explored before and giving information to the waiting bees (onlooker bees) in the hive about the quality of the food source sites which they are exploiting. Onlooker bees wait in the hive and decide on a food source to exploit based on the information shared by the employed bees. Scouts either randomly search the environment in order to find a new food source depending on an internal motivation or based on possible external clues [52].

This emergent intelligent behaviour in foraging bees can be summarized as follows:

1. At the initial phase of the foraging process, the bees start to explore the environment randomly in order to find a food source.
2. After finding a food source, the bee becomes an employed forager and starts to exploit the discovered source. The employed bee returns to the hive with the nectar and unloads the nectar. After unloading the nectar, she can go back to her discovered source site directly or she can share information about her source site by performing a dance on the dance area. If her source is exhausted, she becomes a scout and starts to randomly search for a new source.
3. Onlooker bees waiting in the hive watch the dances advertising the profitable sources and choose a source site depending on the frequency of a dance proportional to the quality of the source.

Please cite this article in press as: B. Akay, D. Karaboga, A modified Artificial Bee Colony algorithm for real-parameter optimization, Inform. Sci. (2010), doi:10.1016/j.ins.2010.07.015

In the ABC algorithm proposed by Karaboga, the position of a food source represents a possible solution to the optimization problem, and the nectar amount of a food source corresponds to the profitability (fitness) of the associated solution. Each food source is exploited by only one employed bee. In other words, the number of employed bees is equal to the number of food sources existing around the hive (number of solutions in the population). The employed bee whose food source has been abandoned becomes a scout.

Using the analogy between emergent intelligence in foraging of bees and the ABC algorithm, the units of the basic ABC algorithm can be explained as follows:

2.1. Producing initial food source sites

If the search space is considered to be the environment of the hive that contains the food source sites, the algorithm starts with randomly producing food source sites that correspond to the solutions in the search space. Initial food sources are produced randomly within the range of the boundaries of the parameters.

$$x_{ij} = x_j^{min} + \mathrm{rand}(0,1)(x_j^{max} - x_j^{min}), \tag{1}$$

where $i = 1 \ldots SN$, $j = 1 \ldots D$. SN is the number of food sources and D is the number of optimization parameters. In addition, counters which store the numbers of trials of solutions are reset to 0 in this phase.

After initialization, the population of the food sources (solutions) is subjected to repeat cycles of the search processes of the employed bees, the onlooker bees and the scout bees. Termination criteria for the ABC algorithm might be reaching a maximum cycle number (MCN) or meeting an error tolerance (ϵ).

2.2. Sending employed bees to the food source sites

As mentioned earlier, each employed bee is associated with only one food source site. Hence, the number of food source sites is equal to the number of employed bees. An employed bee produces a modification on the position of the food source (solution) in her memory depending on local information (visual information) and finds a neighboring food source, and then evaluates its quality. In ABC, finding a neighboring food source is defined by (2)

$$\upsilon_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}). \tag{2}$$

Within the neigbourhood of every food source site represented by x_i, a food source υ_i is determined by changing one parameter of x_i. In Eq. (2), j is a random integer in the range $[1,D]$ and $k \in \{1,2,\ldots SN\}$ is a randomly chosen index that has to be different from $i.\phi_{ij}$ is a uniformly distributed real random number in the range $[-1,1]$.

As can be seen from Eq. (2), as the difference between the parameters of the $x_{i,j}$ and $x_{k,j}$ decreases, the perturbation on the position $x_{i,j}$ decreases. Thus, as the search approaches to the optimal solution in the search space, the step length is adaptively reduced.

If a parameter value produced by this operation exceeds its predetermined boundaries, the parameter can be set to an acceptable value. In this work, the value of the parameter exceeding its boundary is set to its boundaries. If $x_i > x_i^{max}$ then $x_i = x_i^{max}$; If $x_i < x_i^{min}$ then $x_i = x_i^{min}$.

After producing υ_i within the boundaries, a fitness value for a minimization problem can be assigned to the solution υ_i by (3).

$$\mathrm{fitness}_i = \left\{ \begin{array}{ll} 1/(1+f_i) & \text{if } f_i \geqslant 0 \\ 1 + abs(f_i) & \text{if } f_i < 0 \end{array} \right\}, \tag{3}$$

where f_i is the cost value of the solution υ_i. For maximization problems, the cost function can be directly used as a fitness function. A greedy selection is applied between x_i and υ_i; then the better one is selected depending on fitness values representing the nectar amount of the food sources at x_i and υ_i. If the source at υ_i is superior to that of x_i in terms of profitability, the employed bee memorizes the new position and forgets the old one. Otherwise the previous position is kept in memory. If x_i cannot be improved, its counter holding the number of trials is incremented by 1, otherwise, the counter is reset to 0.

2.3. Calculating probability values involved in probabilistic selection

After all employed bees complete their searches, they share their information related to the nectar amounts and the positions of their sources with the onlooker bees on the dance area. This is the multiple interaction feature of the artificial bees of ABC. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source site with a probability related to its nectar amount. This probabilistic selection depends on the fitness values of the solutions in the population. A fitness-based selection scheme might be a roulette wheel, ranking based, stochastic universal sampling, tournament selection or another selection scheme. In basic ABC, roulette wheel selection scheme in which each slice is proportional in size to the fitness value is employed (4):

$$p_i = \frac{\mathrm{fitness}_i}{\sum_{i=1}^{SN} \mathrm{fitness}_i}. \tag{4}$$

In this probabilistic selection scheme, as the nectar amount of food sources (the fitness of solutions) increases, the number of onlookers visiting them increases, too. This is the positive feedback feature of ABC.

2.4. Food source site selection by onlookers based on the information provided by employed bees

In the ABC algorithm, a random real number within the range [0, 1] is generated for each source. If the probability value (p_i in Eq. (4)) associated with that source is greater than this random number then the onlooker bee produces a modification on

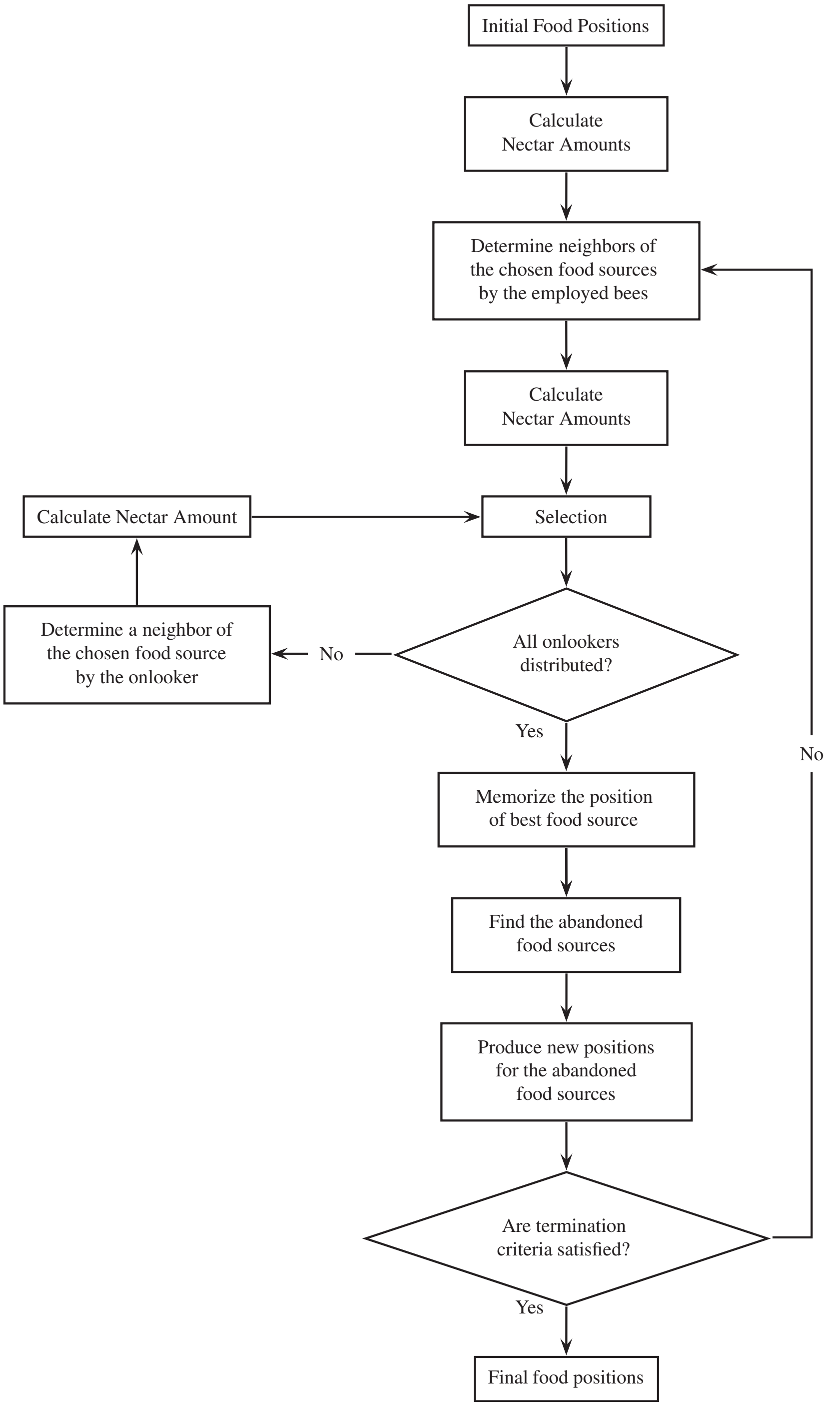


Fig. 1. Flowchart of the Artificial Bee Colony algorithm.

Please cite this article in press as: B. Akay, D. Karaboga, A modified Artificial Bee Colony algorithm for real-parameter optimization, Inform. Sci. (2010), doi:10.1016/j.ins.2010.07.015

the position of this food source site by using Eq. (2) as in the case of the employed bee. After the source is evaluated, greedy selection is applied and the onlooker bee either memorizes the new position by forgetting the old one or keeps the old one. If solution x_i cannot be improved, its counter holding trials is incremented by 1, otherwise, the counter is reset to 0. This process is repeated until all onlookers are distributed onto food source sites.

2.5. Abandonment criteria: Limit and scout production

In a cycle, after all employed bees and onlooker bees complete their searches, the algorithm checks to see if there is any exhausted source to be abandoned. In order to decide if a source is to be abandoned, the counters which have been updated during search are used. If the value of the counter is greater than the control parameter of the ABC algorithm, known as the "*limit*", then the source associated with this counter is assumed to be exhausted and is abandoned. The food source abandoned by its bee is replaced with a new food source discovered by the scout, which represents the negative feedback mechanism and fluctuation property in the self-organization of ABC. This is simulated by producing a site position randomly and replacing it with the abandoned one. Assume that the abandoned source is x_i, then the scout randomly discovers a new food source to be replaced with x_i. This operation can be defined as in (1). In basic ABC, it is assumed that only one source can be exhausted in each cycle, and only one employed bee can be a scout. If more than one counter exceeds the "*limit*" value, one of the maximum ones might be chosen programmatically.

All these units and interactions between them are shown as a flowchart on Fig. 1.

3. Previous work on the ABC algorithm

The ABC algorithm was first applied to numerical optimization [18]. Performance of the ABC algorithm was compared to those of the Genetic Algorithm (GA), Particle Swarm Inspired Evolutionary Algorithm (PS-EA) [5,24]; and to those of Differential Evolution (DE), PSO and Evolutionary Algorithm (EA) on a limited number of basic test problems [25,21]. The effect of region scaling on algorithms including ABC, DE and PSO algorithms was studied in [20]. The ABC algorithm was extended for constrained optimization problems in [23] and was applied to train neural networks [19,22], to medical pattern classification and clustering problems [26,37], to solve TSP problems [13]. Fenglei et al. also studied the control mechanism of local optimal solution in order to improve the global search ability of the algorithm [13]. Singh used the Artificial Bee Colony algorithm for the leaf-constrained minimum spanning tree (LCMST) problem called ABC-LCMST and compared the approach against GA, ACO and tabu search (TS) [53]. In [53], it is reported that ABC-LCMST outperforms the other approaches in terms of the best and average solution qualities and computational time. Rao et al. applied the ABC algorithm to network reconfiguration problem in a radial distribution system in order to minimize the real power loss, improve voltage profile and balance feeder load subject to the radial network structure in which all loads must be energized. The results obtained by the ABC algorithm were better than the other methods compared in the study in terms of quality of the solution and computation efficiency [47]. Bendes and Ozkan used the ABC algorithm for solving direct linear transformation (DLT) which is one of the camera calibration methods by establishing a relation between 3D object coordinate and 2D image plane linearly. Results produced by the ABC algorithm were compared against those of the DE algorithm [7]. Karaboga used the ABC algorithm in the signal processing area for designing digital IIR filters [27]. Qingxian and Haijun proposed a modification in the initialization scheme by making the initial group symmetrical, and the Boltzmann selection mechanism was employed instead of roulette wheel selection for improving the convergence ability of the ABC algorithm [44]. Hemamalini and Simon proposed an economic load dispatch with valve-point effect by using the ABC algorithm [16]. Quan and Shi integrated a search iteration operator based on the fixed point theorem of contractive mapping in Banach spaces with the ABC algorithm in order to improve convergence rate [45]. Pawar et al. applied the ABC algorithm to some problems in mechanical engineering including multi-objective optimization of electro–chemical machining process parameters, optimization of process parameters of the abrasive flow machining process and the milling process [38–40]. In order to maximize the exploitation capacity of the onlooker stage, Tsai et al. introduced the Newtonian law of universal gravitation in the onlooker phase of the basic ABC algorithm in which onlookers are selected based on a roulette wheel (Interactive ABC, IABC) [56]. Baykasoglu et al. incorporated the ABC algorithm with shift neighborhood searches and greedy randomized adaptive search heuristic and applied it to the generalized assignment problem [6].

4. Modified Artificial Bee Colony algorithm

The basic version of the Artificial Bee Colony algorithm has only one control parameter "*limit*" apart from the common control parameters of the population-based algorithms such as population size or colony size (*SN*) and maximum generation number or maximum cycle number (*MCN*). The basic version of the ABC algorithm is very efficient for multimodal and multidimensional basic functions. However, the convergence rate of the algorithm is poorer when working with constrained problems, composite functions and some non-separable functions.

This issue arises from the stochastic variation process in which new solutions are produced from the parent solutions. In this process, some search parameters such as perturbation frequency or magnitude of the perturbation are important since they affect the distribution of new solutions. In order to improve the convergence rate, some modifications have been introduced in the perturbation process of the basic ABC algorithm.

4.1. Frequency of the perturbation

One of the modifications in the ABC algorithm is controlling the frequency of perturbation. In the basic version of ABC, this frequency is fixed. In basic ABC, while producing a new solution, υ_i, changing only one parameter of the parent solution x_i results in a slow convergence rate. In order to overcome this issue, the ABC algorithm is modified by introducing a control parameter, modification rate (MR). By means of this modification, for each parameter x_{ij}, an uniformly distributed random

Table 1
Basic unimodal and multimodal test functions employed in the first part of the experiments.

f	$f(x^*)$	Search range	Initialization range	Formulae
Sphere	$f(\vec{0})=0$	$[-100,100]^D$	$[-100,50]^D$	$f(x)=\sum_{i=1}^{D}x_i^2$
Rosenbrock	$f(\vec{1})=0$	$[-2.048,2.048]^D$	$[-2.048,2.048]^D$	$f(x)=\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$
Ackley	$f(\vec{0})=0$	$[-32.768,32.768]^D$	$[-32.768,16]^D$	$f(x)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+20+e$
Griewank	$f(\vec{0})=0$	$[-600,600]^D$	$[-600,200]^D$	$f(x)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$
Weierstrass	$f(\vec{0})=0$	$[-0.5,0.5]^D$	$[-0.5,0.2]^D$	$f(x)=\sum_{i=1}^{D}\left(\sum_{k=0}^{k\max}\left[a^k\cos\left(2\pi b^k(x_i+0.5)\right)\right]\right)$ $-D\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k 0.5)],\ a=0.5, b=3, k\max=20$
Rastrigin	$f(\vec{0})=0$	$[-5.12,5.12]^D$	$[-5.12,2]^D$	$f(x)=\sum_{i=1}^{D}[x_i^2-10\cos(2\pi x_i)+10]$
Noncontinuous Rastrigin	$f(\vec{0})=0$	$[-5.12,5.12]^D$	$[-5.12,2]^D$	$f(x)=\sum_{i=1}^{D}(y_i^2-10\cos(2\pi y_i)+10)$ $y_i=\begin{cases}x_i & \lvert x_i\rvert<\frac{1}{2}\\ \frac{round(2x_i)}{2} & \lvert x_i\rvert\geqslant\frac{1}{2}\end{cases}$
Schwefel	$f(\vec{4}20.96)=0$	$[-500,500]^D$	$[-500,500]^D$	$f(x)=418.9829xD-\sum_{i=1}^{D}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$

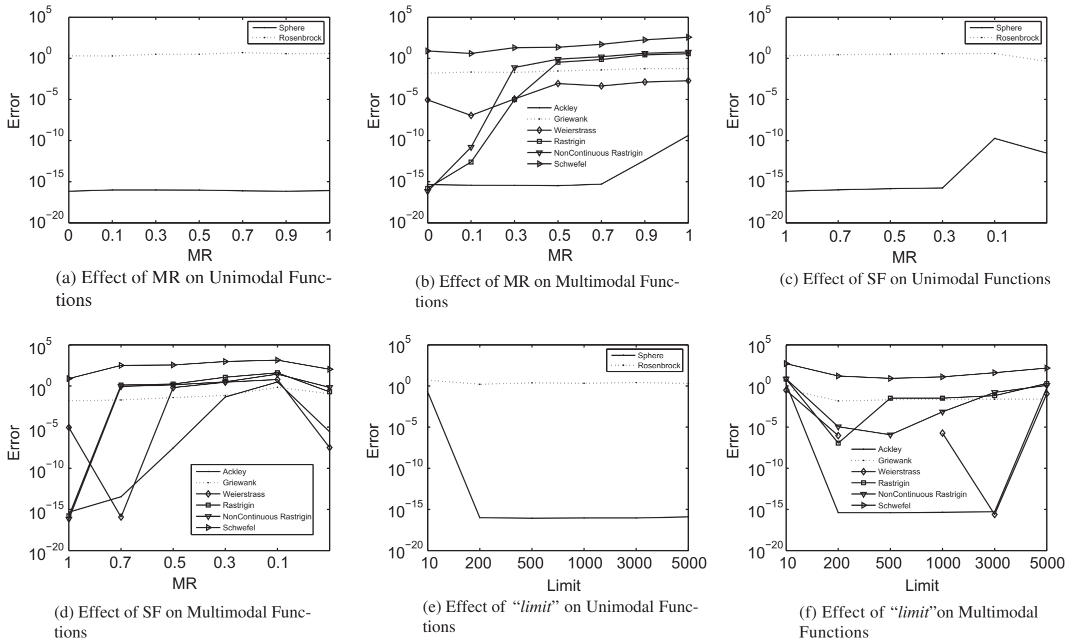
Table 2
Mean of best results obtained through 30 independent runs on basic functions, D:10, Max. Eval.: 30,000, MM:Multimodal,UM:Unimodal.

			UM 1 Sphere	UM Rosenbrock	MM Ackley	MM Griewank
PSO-w			7.96e−051 ± 3.56e−050	3.08e+000 ± 7.69e−001	1.58e−014 ± 1.60e−014	9.69e−002 ± 5.01e−002
PSO-cf			9.84e−105 ± 4.21e−104	6.98e−001 ± 1.46e+000	9.18e−001 ± 1.01e+000	1.19e−001 ± 7.11e−002
PSO-w-local			2.13e−035 ± 6.17e−035	3.92e+000 ± 1.19e+000	6.04e−015 ± 1.67e−015	7.80e−002 ± 3.79e−002
PSO-cf-local			1.37e−079 ± 5.60e−079	8.60e−001 ± 1.56e+000	5.78e−002 ± 2.58e−001	2.80e−002/6.34e−002
UPSO			9.84e−118 ± 3.56e−117	1.40e+000 ± 1.88e+000	1.33e+000 ± 1.48e+000	1.04e−001 ± 7.10e−002
FDR			2.21e−090 ± 9.88e−090	8.67e−001 ± 1.63e+000	3.18e−014 ± 6.40e−014	9.24e002 ± 5.61e−002
FIPS			3.15e−030 ± 4.56e−030	2.78e+000 ± 2.26e−001	3.75e−015 ± 2.13e−014	1.31e−001 ± 9.32e−002
CPSO-H			4.98e−045 ± 1.00e−044	1.53e+000 ± 1.70e+000	1.49e−014 ± 6.97e−015	4.07e−002 ± 2.80e−002
CLPSO			5.15e−029 ± 2.16e−028	2.46e+000 ± 1.70e+000	4.32e−10 ± 2.55e−014	**4.56e−003 ± 4.81e−003**
ABC	MR, SF:1, Limit = 200	0 (basic)	7.09E−017 ± 4.11E−017	2.08E+000 ± 2.44E+000	4.58E− 016 ± 1.76E−016	1.57E−002 ± 9.06E−003
		0.1	1.00E−016 ± 4.88E−017	1.96E+000 ± 2.22E+000	3.79E−016 ± 9.68E−017	2.17E−002 ± 1.78E−002
		0.3	1.01E−016 ± 5.29E−017	3.16E+000 ± 2.35E+000	3.65E−016 ± 1.84E−016	1.93E−002 ± 1.30E−002
		0.5	9.63E−017 ± 5.01E−017	3.20E+000 ± 1.81E+000	3.32E−016 ± 1.84E−016	2.94E−002 ± 2.47E−002
		0.7	7.92E−017 ± 4.87E−017	5.06E+000 ± 1.69E+000	5.13E−016 ± 6.56E−016	4.00E−002 ± 3.52E−002
		0.9	7.04E−017 ± 4.55E−017	3.66E+000 ± 1.97E+000	4.21E−013 ± 2.04E−012	5.65E−002 ± 3.05E−002
		1	8.28E−017 ± 4.95E−017	3.97E+000 ± 2.24E+000	4.29E−010 ± 2.31E−009	5.61E−002 ± 3.26E−002
	SF, MR:0, Limit = 200	0.7	1.05E−016 ± 5.26E−017	2.77E+000 ± 2.26E+000	3.41E−014 ± 1.12E−013	2.00E−002 ± 1.59E−002
		0.5	1.44E−016 ± 3.72E−017	3.22E+000 ± 2.05E+000	2.93E−008 ± 1.53E−007	3.87E−002 ± 2.64E−002
		0.3	1.77E−016 ± 5.48E−017	3.79E+000 ± 1.99E+000	4.59E−002 ± 2.10E−001	7.15E−002 ± 5.92E−002
		0.1	1.89E−010 ± 6.60E−010	3.89E+000 ± 1.49E+000	3.05E+000 ± 4.29E+000	6.81E−001 ± 8.39E−001
		ASF	3.00E−012 ± 5.48E−012	**4.42E−001 ± 8.67E−001**	2.70E−006 ± 1.46E−005	1.14E−001 ± 1.25E−001
	Limit, MR:, SF	10	2.00E−001 ± 3.24E−001	5.28E+000 ± 1.49E+000	1.82E+000 ± 5.62E−001	4.25E−001 ± 1.43E−001
		200	9.68E−017 ± 5.08E−017	1.61E+000 ± 1.93E+000	4.03E−016 ± 1.25E−016	1.52E−002 ± 1.28E−002
		500	8.33E−017 ± 4.99E−017	2.38E+000 ± 2.59E+000	3.92E−016 ± 1.07E−016	2.10E−002 ± 1.32E−002
		1000	9.04E−017 ± 4.81E−017	2.15E+000 ± 2.46E+000	4.39E−016 ± 2.22E−016	2.36E−002 ± 1.98E−002
		3000	9.08E−017 ± 4.99E−017	2.60E+000 ± 2.61E+000	5.11E−016 ± 1.78E−016	2.50E−002 ± 1.46E−002
		5000	1.21E−016 ± 6.66E−017	2.08E+000 ± 2.54E+000	5.87E−001 ± 1.36E+000	2.59E−002 ± 2.01E−002

Please cite this article in press as: B. Akay, D. Karaboga, A modified Artificial Bee Colony algorithm for real-parameter optimization, Inform. Sci. (2010), doi:10.1016/j.ins.2010.07.015

Table 3
Mean of best results obtained through 30 independent runs on basic functions, D: 10, Max. Eval.: 30,000, MM: Multimodal,UM: Unimodal.

			MM 5 Weierstrass	MM 6Rastrigin	MM 7 NCRastrigin	MM 8 schwefel
PSO-w			2.28e−003 ± 7.04e−003	5.82e+000 ± 2.96e+000	4.05e+000/ ± 2.58e+000	3.20e+002 ± 1.85e+002
PSO-cf			6.69e−001 ± 7.17e−001	1.25e+001 ± 5.17e+000	1.20e+001 ± 4.99e+000	9.87e+002 ± 2.76e+002
PSO-w-local			1.41e−006 ± 6.31e−006	3.88e+000 ± 2.30e+000	4.77e+000 ± 2.84e+000	3.26e+002 ± 1.32e+002
PSO-cf-local			7.85e−002 ± 5.16e−002	9.05e+000 ± 3.48e+000	5.95e+000 ± 2.60e+000	8.78e+002 ± 2.93e+002
UPSO			1.14e+000 ± 1.17e+000	1.17e+001 ± 6.11e+000	5.85e+000 ± 3.15e+000	1.08e+003 ± 2.68e+002
FDR			3.01e−003 ± 7.20e−003	7.51e+000 ± 3.05e+000	3.35e+000 ± 2.01e+000	8.51e+002 ± 2.76e+002
FIPS			2.02e−003 ± 6.40e−003	2.12e+000 ± 1.33e+000	4.35e+000 ± 2.80e+000	7.10e+001 ± 1.50e+002
CPSO-H			1.07e−015 ± 1.67e−015	0 ± 0	2.00e−001 ± 4.10e−001	2.13e+002 ± 1.41e+002
CLPSO			0 ± 0	0 ± 0	0 ± 0	**0 ± 0**
ABC	MR, SF: 1,Limit = 200	0 (basic)	9.01E−006 ± 4.61E−005	1.61E−016 ± 5.20E−016	6.64E−017 ± 3.96E−017	7.91E+000 ± 2.95E+001
		0.1	1.15E−007 ± 6.17E−007	2.54E−013 ± 1.37E−012	1.58E− 011 ± 7.62E−011	3.96E+000 ± 2.13E+001
		0.3	1.17E−005 ± 4.90E−005	9.61E−006 ± 5.17E−005	7.84E−002 ± 2.54E−001	1.97E+001 ± 4.41E+001
		0.5	8.80E−004 ± 2.94E−003	3.38E−001 ± 6.44E−001	8.00E−001 ± 7.02E−001	2.25E+001 ± 4.45E+001
		0.7	4.45E−004 ± 1.69E−003	7.31E−001 ± 7.23E−001	1.59E+000 ± 9.59E−001	5.13E+001 ± 8.31E+001
		0.9	1.34E−003 ± 5.62E−003	2.68E+000 ± 1.95E+000	4.21E+000 ± 1.37E+000	1.78E+002 ± 1.25E+002
		1	1.92E−003 ± 6.63E−003	3.71E+000 ± 1.58E+000	5.89E+000 ± 1.69E+000	3.69E+002 ± 1.52E+002
	SF, MR: 0, Limit = 200	0.7	1.18E−016 ± 6.38E−016	1.29E+000 ± 8.95E−001	9.00E−001 ± 7.00E−001	3.20E+002 ± 1.36E+002
		0.5	6.33E−001 ± 7.00E−001	1.73E+000 ± 1.18E+000	1.37E+000 ± 6.57E−001	3.59E+002 ± 1.16E+002
		0.3	2.89E+000 ± 1.26E+000	1.17E+001 ± 4.71E+000	3.17E+000 ± 1.42E+000	9.49E+002 ± 2.19E+002
		0.1	6.08E+000 ± 1.46E+000	3.94E+001 ± 1.41E+001	2.77E+001 ± 8.45E+000	1.40E+003 ± 2.61E+002
		ASF	3.24E−008 ± 3.06E−008	1.94E−001 ± 3.85E−001	6.80E−001 ± 7.80E−001	1.09E+002 ± 8.49E+001
		10	3.10E−001 ± 1.54E−001	6.97E+000 ± 1.69E+000	7.33E+000 ± 1.65E+000	5.16E+002 ± 8.75E+001
	Limit, MR: 1, SF: 1	200	9.41E−007 ± 5.07E−006	1.14E−007 ± 6.16E−007	1.12E−005 ± 6.04E−005	1.65E+001 ± 4.01E+001
		500	0.00E+000 ± 0.00E+000	3.33E−002 ± 1.79E−001	1.17E−006 ± 5.19E−006	8.39E+000 ± 2.95E+001
		1000	1.88E−006 ± 1.01E−005	3.32E−002 ± 1.79E−001	7.22E−004 ± 3.89E−003	1.30E+001 ± 3.55E+001
		3000	2.37E−016 ± 1.28E−015	6.63E−002 ± 2.48E−001	1.67E−001 ± 4.53E−001	4.08E+001 ± 6.42E+001
		5000	1.15E−001 ± 3.79E−001	2.02E+000 ± 3.35E+000	1.17E+000 ± 3.01E+000	1.53E+002 ± 1.20E+002

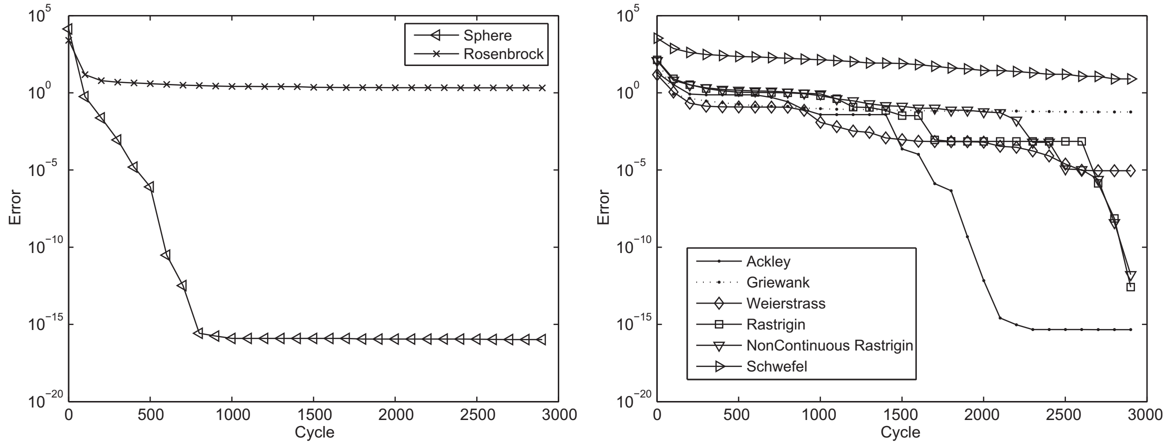


(a) Effect of MR on Unimodal Functions
(b) Effect of MR on Multimodal Functions
(c) Effect of SF on Unimodal Functions
(d) Effect of SF on Multimodal Functions
(e) Effect of "*limit*" on Unimodal Functions
(f) Effect of "*limit*"on Multimodal Functions

Fig. 2. Mean of best function values for different control parameter settings.

number, $(0 \leqslant R_{ij} \leqslant 1)$, is produced and if the random number is less than MR, then the parameter x_{ij} is modified as in the Eq. (5).

$$\upsilon_{ij} = \begin{cases} x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), & \text{if } R_{ij} < MR, \\ x_{ij}, & \text{otherwise}, \end{cases} \tag{5}$$



(a) Convergence graphs of Unimodal Functions

(b) Convergence graphs of Multimodal Functions

Fig. 3. Convergence graphs of the ABC algorithm on basic functions.

Table 4
Mean and standard deviations of error values obtained from the ABC algorithm with different colony sizes (*CS*) for basic functions with different dimensions (*D*).

CS	Functions	D		
		10	50	100
10	Sphere	6.83E−017 ± 4.03E−017	1.40E−014 ± 4.13E−014	4.36E−005 ± 9.15E−005
	Rosenbrock	2.66E+000 ± 2.48E+000	6.22E+001 ± 2.65E+001	2.12E+002 ± 4.52E+001
	Ackley	4.56E−016 ± 1.69E−016	5.64E−001 ± 5.86E−001	1.13E+000 ± 7.01E−001
	Griewank	1.57E−002 ± 1.34E−002	1.49E−002 ± 2.91E−002	3.01E−001 ± 1.49E+000
	Weierstrass	0.00E+000 ± 0.00E+000	8.36E−001 ± 1.33E+000	1.12E+000 ± 1.32E+000
	Rastrigin	2.39E−011 ± 1.29E−010	6.30E+000 ± 3.92E+000	3.22E+001 ± 7.10E+000
	NCRastrigin	3.33E−002 ± 1.80E−001	9.11E+000 ± 4.81E+000	3.36E+001 ± 6.10E+000
	Schwefel	2.02E−003 ± 1.01E−002	1.13E+003 ± 3.34E+002	3.68E+003 ± 5.80E+002
20	Sphere	4.97E−017 ± 2.20E−017	7.60E−016 ± 1.30E−016	8.89E−007 ± 1.79E−006
	Rosenbrock	1.22E+000 ± 1.75E+000	4.68E+001 ± 2.31E+001	1.80E+002 ± 3.49E+001
	Ackley	2.69E−016 ± 6.59E−017	7.75E−011 ± 1.04E−010	2.22E−003 ± 2.25E−003
	Griewank	5.60E−003 ± 6.78E−003	4.84E−003 ± 7.14E−003	2.90E−003 ± 7.55E−003
	Weierstrass	0.00E+000 ± 0.00E+000	6.79E−010 ± 2.70E−009	1.53E−002 ± 3.42E−003
	Rastrigin	4.92E−017 ± 1.99E−017	2.19E−001 ± 4.03E−001	1.33E+001 ± 3.06E+000
	NCRastrigin	5.63E−017 ± 3.40E−017	7.37E−001 ± 6.79E−001	1.84E+001 ± 3.68E+000
	Schwefel	1.27E−004 ± 4.43E−008	3.61E+002 ± 1.60E+002	2.73E+003 ± 3.46E+002
50	Sphere	4.86E−017 ± 6.71E−018	6.43E−016 ± 7.59E−017	5.53E−008 ± 4.56E−008
	Rosenbrock	1.07E−001 ± 1.71E−001	3.08E+001 ± 1.15E+001	1.45E+002 ± 3.01E+001
	Ackley	2.30E−016 ± 4.96E−017	8.22E−012 ± 4.82E−012	3.08E−004 ± 1.98E−004
	Griewank	1.04E−003 ± 3.02E−003	4.44E−012 ± 2.22E−011	1.36E−004 ± 7.27E−004
	Weierstrass	0.00E+000 ± 0.00E+000	6.86E−013 ± 7.70E−013	8.39E−003 ± 1.71E−003
	Rastrigin	4.44E−017 ± 9.22E−018	3.80E−013 ± 1.19E−012	6.26E+000 ± 2.44E+000
	NCRastrigin	4.54E−017 ± 9.79E−018	1.42E−011 ± 5.06E−011	1.24E+001 ± 3.48E+000
	Schwefel	1.27E−004 ± 0.00E+000	4.02E+001 ± 7.17E+001	2.00E+003 ± 3.28E+002

where $k \in \{1, 2, \ldots SN\}$ is randomly chosen index that has to be different from i and *MR* is the modification rate which takes value between 0 and 1. A lower value of *MR* may cause solutions to improve slowly while a higher one may cause too much diversity in a solution and hence in the population.

4.2. Magnitude of the perturbation

Another modification is related to the ratio of the variance operator of the basic ABC algorithm. In basic ABC, a random perturbation which avoids getting stuck at local minima is added to the current solution in order to produce a new solution (2). This random perturbation is the difference of the solutions (x_i and x_k) weighted by a random real number ϕ_{ij}. The value of ϕ_{ij} varies within the range $[-1, 1]$ in the basic ABC while it varies within the range $[-SF, SF]$ in the modified ABC algorithm. Hence, magnitude of the perturbation is controlled by a control parameter called the scaling factor (*SF*). This value is set be-

Table 5
Time Complexity of the ABC algorithm on Rosenbrock function (D: Dimension).

	T_0	T_1	$\widehat{T}_2$	Complexity $((\widehat{T}_2 - T_1)/T_0)$
D = 10	0.411260649048971	0.233486099490298	0.704668386624557	1.1457
D = 50	0.411260649048971	0.474651666622434	1.33588620900142	2.0941
D = 100	0.411260649048971	0.732597858107665	1.98444847040615	3.0439

Table 6
Values for the control parameters of the ABC algorithm used for hybrid functions, D: Dimension of the problem.

Colony size	D
Max. Cycle number	10,000
"*limit*"	200
MR	0.4
SF	1

Table 7
Time Complexity of the ABC Algorithm on Composite Function 3 (D: Dimension).

	T_0	T_1	$\widehat{T}_2$	Complexity $((\widehat{T}_2 - T_1)/T_0)$
D = 10	0.37499985191971	0.84299985319376	7.62500013224781	18.0853412190309
D = 30		0.890999869443476	22.5620004348457	57.7893576609787
D = 50		0.938000343739986	45.5779997399077	117.08004553051

Table 8
D: 10, Colony Size: 10, "*limit*": 200, Cycle: 10,000 Max.FES: 100,000, MR = 0.4, MTE: Mean of Term. Err. of 25 runs, STE: Std of Term. Err. of 25 runs.

FES	Prob	1	2	3	4	5	6	7	8
1.00E+03	Best	1.76E+00	3.92E+01	2.98E+04	2.12E+02	3.64E+02	5.93E+03	5. 22E+00	2.04E+01
	7th	1.54E+01	1.47E+02	5.66E+04	5.55E+02	1.55E+03	4.37E+04	1.02E+01	2.07E+01
	Median	4.44E+01	2.08E+02	8.52E+04	8.49E+02	2.34E+03	1.04E+05	1.82E+01	2.08E+01
	19th	7.29E+01	3.11E+02	1.03E+05	1.37E+03	2.86E+03	1.88E+05	2.89E+01	2.08E+01
	Worst	1.03E+02	6.94E+02	1.71E+05	3.01E+03	6.79E+03	8.13E+05	6.54E+01	2.09E+01
	Mean	4.43E+01	2.41E+02	8.51E+04	1.07E+03	2.66E+03	1.73E+05	2.39E+01	2.07E+01
	Std	2.91E+01	1.54E+02	3.74E+04	7.28E+02	1.66E+03	1.97E+05	1.70E+01	1.30E−01
1.00E+04	Best	0.00E+00	0.00E+00	6.33E+03	0.00E+00	4.82E−02	5.67E+00	4.15E−01	2.03E+01
	7th	0.00E+00	0.00E+00	1.51E+04	0.00E+00	3.36E−01	1.24E+01	5.58E−01	2.04E+01
	Median	0.00E+00	0.00E+00	2.02E+04	0.00E+00	1.22E+00	1.62E+01	6.72E−01	2.05E+01
	19th	0.00E+00	0.00E+00	2.24E+04	0.00E+00	9.11E+00	2.93E+01	8.56E−01	2.06E+01
	Worst	0.00E+00	0.00E+00	3.72E+04	0.00E+00	6.73E+01	1.23E+02	1.16E+00	2.07E+01
	Mean	0.00E+00	0.00E+00	1.96E+04	0.00E+00	9.13E+00	2.68E+01	7.17E−01	2.05E+01
	Std	0.00E+00	0.00E+00	6.69E+03	0.00E+00	1.57E+01	2.77E+01	2.02E−01	1.09E−01
1.00E+05	Best	0.00E+00	0.00E+00	7.83E+02	0.00E+00	0.00E+00	1.07E+00	2.38E−01	2.02E+01
	7th	0.00E+00	0.00E+00	3.82E+03	0.00E+00	0.00E+00	3.23E+00	3.05E−01	2.03E+01
	Median	0.00E+00	0.00E+00	6.33E+03	0.00E+00	0.00E+00	4.44E+00	3.37E−01	2.04E+01
	19th	0.00E+00	0.00E+00	8.05E+03	0.00E+00	0.00E+00	6.50E+00	3.90E−01	2.04E+01
	Worst	0.00E+00	0.00E+00	1.42E+04	0.00E+00	0.00E+00	8.84E+00	4.95E−01	2.05E+01
	Mean	0.00E+00	0.00E+00	6.27E+03	0.00E+00	0.00E+00	4.69E+00	3.46E−01	2.04E+01
	Std	0.00E+00	0.00E+00	2.83E+03	0.00E+00	0.00E+00	2.24E+00	6.43E−02	6.52E−02
	MTE	8.63E−09	7.65E−09	6.27E+03	1.74E−09	1.15E−03	4.69E+00	3.46E−01	2.04E+01
	STE	1.31E−09	1.40E−09	2.83E+03	7.95E−09	2.33E−03	2.24E+00	6.43E−02	6.52E−02

fore running the algorithm. A lower value of *SF* allows the search to fine tuning the process in small steps while causing slow convergence. A larger value of *SF* speeds up the search, but it reduces the exploitation capability of the perturbation process. For some classes of problems, lower values of *SF* are appropriate while for some, higher ones are convenient. For this reason, the modified algorithm may change *SF* automatically during the search, called adaptive *SF* (*ASF*). Automatic tuning of *SF* is conducted by using Rechenberg's 1/5 mutation rule which states that the ratio of successful mutations to all mutations should be 1/5 [2]. Changing step size according to 1/5 rule in every m number of cycles is performed as in the Eq. (6):

Table 9
D: 10, Colony Size: 10, "*limit*": 200, Cycle: 10,000 Max.FES: 100,000, MR = 0.4, MTE: Mean of Term. Err. of 25 runs, STE: Std of Term. Err. of 25 runs.

FES	Prob	9	10	11	12	13	14	15	16	17
1.00E+03	Best	1.95E+01	4.52E+01	7.70E+00	2.81E+03	2.04E+00	3.58E+00	2. 57E+02	2.69E+02	2.68E+02
	7th	2.72E+01	6.47E+01	9.81E+00	6.98E+03	3.17E+00	4.12E+00	3.95E+02	3.18E+02	3.18E+02
	Median	3.04E+01	6.99E+01	1.04E+01	9.63E+03	3.64E+00	4.24E+00	4.56E+02	3.40E+02	3.35E+02
	19th	3.71E+01	7.73E+01	1.07E+01	1.42E+04	4.47E+00	4.32E+00	5.14E+02	3.63E+02	3.74E+02
	Worst	4.74E+01	9.35E+01	1.14E+01	2.45E+04	7.20E+00	4.51E+00	5.88E+02	3.88E+02	5.27E+02
	Mean	3.19E+01	7.02E+01	1.02E+01	1.11E+04	3.92E+00	4.22E+00	4.49E+02	3.37E+02	3.44E+02
	Std	6.72E+00	1.12E+01	8.16E−01	5.53E+03	1.20E+00	1.80E−01	7.53E+01	3.44E+01	5.06E+01
1.00E+04	Best	1.28E−02	1.91E+01	6.62E+00	6.01E+01	8.00E−01	3.40E+00	1.31E+02	1.77E+02	2.02E+02
	7th	1.05E+00	3.10E+01	7.50E+00	6.76E+02	1.44E+00	3.66E+00	1.99E+02	2.14E+02	2.48E+02
	Median	2.35E+00	3.58E+01	7.86E+00	1.64E+03	1.58E+00	3.81E+00	2.32E+02	2.32E+02	2.63E+02
	19th	4.91E+00	4.10E+01	8.55E+00	2.25E+03	1.84E+00	3.97E+00	2.66E+02	2.71E+02	2.78E+02
	Worst	7.28E+00	4.90E+01	8.83E+00	3.07E+03	2.18E+00	4.10E+00	3.49E+02	3.05E+02	3.64E+02
	Mean	2.91E+00	3.52E+01	7.90E+00	1.42E+03	1.60E+00	3.81E+00	2.30E+02	2.42E+02	2.66E+02
	Std	2.35E+00	7.45E+00	6.38E−01	8.30E+02	3.59E−01	1.86E−01	5.45E+01	3.32E+01	2.96E+01
1.00E+05	Best	2.02E+01	1.47E+01	4.37E+00	6.01E+01	1.19E−01	3.07E+00	1.03E+02	1.75E+02	1.84E+02
	7th	2.03E+01	1.91E+01	5.69E+00	2.97E+02	3.79E−01	3.43E+00	1.31E+02	1.85E+02	2.04E+02
	Median	2.04E+01	2.27E+01	6.12E+00	3.43E+02	5.03E−01	3.51E+00	1.49E+02	1.94E+02	2.15E+02
	19th	2.04E+01	2.70E+01	6.66E+00	4.99E+02	1.92E−01	3.65E+00	1.68E+02	2.11E+02	2.29E+02
	Worst	2.05E+01	2.96E+01	7.20E+00	1.02E+03	8.43E−01	3.78E+00	1.97E+02	2.22E+02	2.52E+02
	Mean	2.04E+01	2.27E+01	6.13E+00	3.99E+02	4.90E−01	3.51E+00	1.48E+02	1.98E+02	2.16E+02
	Std	6.52E−02	4.24E+00	6.65E−01	2.11E+02	1.92E−01	1.55E−01	2.40E+01	1.46E+01	1.61E+01
	MTE	7.89E−09	2.27E+01	6.13E+00	3.99E+02	4.90E−01	3.51E+00	1.48E+02	1.98E+02	2.16E+02
	STE	1.82E−09	4.24E+00	6.65E−01	2.11E+02	1.92E−01	1.55E−01	2.40E+01	1.46E+01	1.61E+01

Table 10
D: 10, Colony Size: 10, "*limit*": 200, Cycle: 10,000, Max.FES: 100,000, MR = 0.4, MTE: Mean of Term. Err. of 25 runs, STE: Std of Term. Err. of 25 runs.

FES	Prob	18	19	20	21	22	23	24	25
1.00E+03	Best	5.87E+02	5.49E+02	5.22E+02	7.02E+02	8.79E+02	7.49E+02	2.01E+02	2.00E+02
	7th	6.57E+02	6.29E+02	6.54E+02	9.55E+02	9.23E+02	9.66E+02	2.01E+02	2.00E+02
	Median	7.00E+02	6.71E+02	7.05E+02	9.96E+02	9.53E+02	1.02E+03	2.02E+02	2.00E+02
	19th	9.44E+02	7.19E+02	8.14E+02	1.03E+03	9.70E+02	1.04E+03	2.02E+02	2.00E+02
	Worst	1.02E+03	1.07E+03	9.96E+02	1.13E+03	1.09E+03	1.15E+03	5.56E+02	2.01E+02
	Mean	7.78E+02	7.04E+02	7.47E+02	9.91E+02	9.54E+02	1.00E+03	2.16E+02	2.00E+02
	Std	1.47E+02	1.29E+02	1.31E+02	8.80E+01	4.88E+01	8.19E+01	6.94E+01	2.77E−01
1.00E+04	Best	4.14E+02	3.95E+02	4.38E+02	5.20E+02	8.49E+02	5.49E+02	2.01E+02	2.00E+02
	7th	5.38E+02	5.15E+02	5.09E+02	5.82E+02	8.58E+02	5.64E+02	2.01E+02	2.00E+02
	Median	5.67E+02	5.71E+02	5.46E+02	6.19E+02	8.64E+02	6.38E+02	2.01E+02	2.00E+02
	19th	5.87E+02	6.18E+02	5.77E+02	8.00E+02	8.70E+02	8.73E+02	2.01E+02	2.00E+02
	Worst	8.15E+02	6.83E+02	6.49E+02	8.03E+02	8.91E+02	8.84E+02	2.02E+02	2.00E+02
	Mean	5.66E+02	5.62E+02	5.49E+02	6.67E+02	8.66E+02	6.82E+02	2.01E+02	2.00E+02
	Std	7.06E+01	7.06E+01	4.80E+01	1.13E+02	9.70E+00	1.32E+02	3.88E−01	6.65E−03
1.00E+05	Best	3.65E+02	3.58E+02	3.88E+02	3.97E+02	8.79E+02	7.49E+02	2.00E+02	2.00E+02
	7th	4.66E+02	4.51E+02	4.40E+02	5.29E+02	9.23E+02	9.66E+02	2.00E+02	2.00E+02
	Median	4.87E+02	4.78E+02	5.00E+02	5.82E+02	9.53E+02	1.02E+03	2.01E+02	2.00E+02
	19th	5.04E+02	4.94E+02	5.05E+02	6.29E+02	9.70E+02	1.04E+03	2.01E+02	2.00E+02
	Worst	5.42E+02	5.24E+02	5.42E+02	8.00E+02	1.09E+03	1.15E+03	2.01E+02	2.00E+02
	Mean	4.77E+02	4.68E+02	4.77E+02	6.00E+02	9.54E+02	1.00E+03	2.01E+02	2.00E+02
	Std	4.44E+01	3.81E+01	4.60E+01	9.76E+01	4.88E+01	8.19E+01	9.01E−02	1.95E−03
	MTE	4.77E+02	4.68E+02	4.77E+02	6.00E+02	9.54E+02	1.00E+03	2.01E+02	2.00E+02
	STE	4.44E+01	3.81E+01	4.60E+01	9.76E+01	4.88E+01	8.19E+01	9.01E−02	1.95E−03

$$SF(t+1) = \begin{cases} SF(t) * 0.85 & \text{if } \varphi(m) < 1/5, \\ SF(t)/0.85 & \text{if } \varphi(m) > 1/5, \\ SF(t) & \text{if } \varphi(m) = 1/5. \end{cases} \tag{6}$$

If the algorithm cannot improve the solution with respect to Rechenberg's 1/5 rule, that is the ratio of successful mutations to all mutations ($\varphi(m)$) is less than 1/5, *SF* is decreased. If $\varphi(m)$ is greater than 1/5 then *SF* is increased in order to speed up the search.

Table 11
D: 30, Colony Size: 30, "*limit*": 200, Cycle: 10,000, Max.FES: 300,000, MR = 0.4, MTE: Mean of Term. Err. of 25 runs, STE: Std of Term. Err. of 25 runs.

FES	Prob	1	2	3	4	5	6	7	8
1.00E+03	Best	6.59E+03	1.18E+05	1.02E+06	1.34E+05	1.86E+04	6.87E+08	1.31E+03	2.11E+01
	7th	1.51E+04	1.83E+05	1.41E+06	3.97E+05	2.36E+04	1.47E+09	1.76E+03	2.12E+01
	Median	1.73E+04	2.11E+05	1.55E+06	4.73E+05	2.51E+04	2.22E+09	1.94E+03	2.12E+01
	19th	1.82E+04	2.42E+05	1.83E+06	5.86E+05	2.79E+04	4.61E+09	2.27E+03	2.13E+01
	Worst	2.04E+04	2.88E+05	2.18E+06	8.62E+05	3.03E+04	7.29E+09	2.54E+03	2.13E+01
	Mean	1.59E+04	2.11E+05	1.60E+06	4.83E+05	2.55E+04	3.04E+09	2.00E+03	2.12E+01
	Std	3.51E+03	4.38E+04	2.89E+05	1.52E+05	2.73E+03	1.91E+09	3.63E+02	5.03E−02
1.00E+04	Best	2.34E−01	1.59E+00	3.35E+05	5.70E+02	8.40E+03	1.35E+04	5.82E+00	2.10E+01
	7th	3.33E−01	4.15E+00	4.59E+05	7.30E+02	1.14E+04	2.03E+04	1.35E+01	2.11E+01
	Median	4.33E−01	5.77E+00	4.97E+05	9.02E+02	1.19E+04	3.09E+04	1.65E+01	2.11E+01
	19th	6.82E−01	8.42E+00	5.48E+05	1.74E+03	1.32E+04	4.36E+04	2.00E+01	2.11E+01
	Worst	9.78E−01	1.56E+01	7.29E+05	2.37E+03	1.55E+04	1.63E+05	2.38E+01	2.12E+01
	Mean	5.18E−01	6.61E+00	5.00E+05	1.19E+03	1.20E+04	3.64E+04	1.60E+01	2.11E+01
	Std	2.24E−01	3.40E+00	8.34E+04	5.83E+02	1.64E+03	2.84E+04	4.40E+00	5.44E−02
1.00E+05	Best	0.00E+00	0.00E+00	1.86E+05	1.86E+05	5.20E+03	5.93E+01	1.09E−03	2.08E+01
	7th	0.00E+00	0.00E+00	2.41E+05	2.41E+05	6.32E+03	1.44E+02	4.11E−03	2.10E+01
	Median	0.00E+00	0.00E+00	2.67E+05	2.67E+05	7.06E+03	1.82E+02	5.03E−02	2.10E+01
	19th	0.00E+00	0.00E+00	2.98E+05	2.98E+05	7.61E+03	2.93E+02	2.41E−01	2.10E+01
	Worst	0.00E+00	0.00E+00	3.39E+05	3.39E+05	8.74E+03	3.56E+02	7.56E−01	2.11E+01
	Mean	0.00E+00	0.00E+00	2.65E+05	2.65E+05	6.91E+03	2.09E+02	1.92E−01	2.10E+01
	Std	0.00E+00	0.00E+00	3.95E+04	3.95E+04	9.13E+02	8.59E+01	2.62E−01	6.15E−02
3.00E+05	Best	0.00E+00	0.00E+00	1.76E+05	1.76E+05	4.53E+03	3.35E+01	2.68E−08	2.08E+01
	7th	0.00E+00	0.00E+00	1.99E+05	1.99E+05	5.54E+03	9.37E+01	3.69E−01	2.09E+01
	Median	0.00E+00	0.00E+00	2.16E+05	2.16E+05	6.13E+03	1.43E+02	1.17E+02	2.10E+01
	19th	0.00E+00	0.00E+00	2.39E+05	2.39E+05	6.43E+03	1.82E+02	1.72E+02	2.10E+01
	Worst	0.00E+00	0.00E+00	2.72E+05	2.72E+05	7.68E+03	2.60E+02	2.60E+02	2.10E+01
	Mean	0.00E+00	0.00E+00	2.20E+05	2.20E+05	6.02E+03	1.38E+02	1.05E+02	2.09E+01
	Std	0.00E+00	0.00E+00	2.53E+04	2.53E+04	7.16E+02	5.81E+01	8.20E+01	5.63E−02
	MTE	9.35E−09	9.06E−09	2.20E+05	9.01E−09	6.02E+03	1.38E+02	1.49E−02	2.09E+01
	STE	6.21E−10	5.80E−10	2.53E+04	8.89E−10	7.16E+02	5.81E+01	7.23E−02	5.63E−02

Table 12
D: 30, Colony Size: 30, "*limit*": 200, Cycle: 10,000, Max.FES: 300,000, MR = 0.4, MTE: Mean of Term. Err. of 25 runs, STE: Std of Term. Err. of 25 runs.

FES	Prob	9	10	11	12	13	14	15	16	17
1.00E+03	Best	2.13E+02	3.44E+02	4.06E+01	5.63E+05	3.79E+01	1.39E+01	6.21E+02	4.39E+02	4.47E+02
	7th	2.74E+02	4.13E+02	4.22E+01	7.42E+05	5.10E+01	1.40E+01	7.39E+02	4.96E+02	5.37E+02
	Median	2.95E+02	4.54E+02	4.33E+01	7.85E+05	7.35E+01		8.31E+02	5.29E+02	5.84E+02
	19th	3.03E+02	4.79E+02	4.46E+01	8.77E+05	1.06E+02	1.41E+01	8.63E+02	5.58E+02	6.48E+02
	Worst	3.28E+02	5.19E+02	4.66E+01	9.66E+05	1.31E+02	1.42E+01	9.03E+02	6.55E+02	8.43E+02
	Mean	2.88E+02	4.44E+02	4.34E+01	8.04E+05	7.82E+01	1.44E+01	8.05E+02	5.25E+02	6.08E+02
	Std	2.39E+01	4.56E+01	1.80E+00	9.42E+04	2.87E+01	1.41E+01	7.60E+01	4.77E+01	9.69E+01
1.00E+04	Best	1.01E+02	2.27E+02	3.65E+01	1.10E+05	1.45E+01	1.42E−01	4.74E+02	2.86E+02	3.00E+02
	7th	1.18E+02	2.52E+02	3.83E+01	2.17E+05	1.69E+01	1.35E+01	5.26E+02	3.33E+02	3.43E+02
	Median	1.23E+02	2.67E+02	4.00E+01	2.26E+05	1.75E+01	1.37E+01	5.36E+02	3.52E+02	3.69E+02
	19th	1.32E+02	2.73E+02	4.06E+01	2.49E+05	1.85E+01	1.38E+01	5.41E+02	3.70E+02	3.88E+02
	Worst	1.50E+02	2.85E+02	4.19E+01	3.36E+05	2.02E+01	1.39E+01	6.22E+02	4.21E+02	4.87E+02
	Mean	1.24E+02	2.62E+02	3.97E+01	2.35E+05	1.75E+01	1.40E+01	5.36E+02	3.54E+02	3.73E+02
	Std	1.22E+01	1.61E+01	1.49E+00	4.81E+04	1.54E+00	1.38E+01	2.98E+01	3.68E+01	4.58E+01
1.00E+05	Best	6.07E+01	1.62E+02	3.43E+01	8.86E+04	9.13E+00	1.21E−01	2.00E+02	2.77E+02	2.67E+02
	7th	7.12E+01	2.06E+02	3.59E+01	1.03E+05	1.15E+01	1.31E+01	3.00E+02	2.99E+02	3.10E+02
	Median	7.72E+01	2.16E+02	3.68E+01	1.23E+05	1.20E+01	1.35E+01	3.00E+02	3.16E+02	3.27E+02
	19th	8.14E+01	2.25E+02	3.72E+01	1.41E+05	1.28E+01	1.35E+01	3.03E+02	3.33E+02	3.38E+02
	Worst	9.04E+01	2.42E+02	3.84E+01	1.74E+05	1.35E+01	1.36E+01	4.84E+02	4.00E+02	3.84E+02
	Mean	7.67E+01	2.14E+02	3.65E+01	1.22E+05	1.19E+01	1.37E+01	3.15E+02	3.21E+02	3.27E+02
	Std	7.60E+00	1.58E+01	1.11E+00	2.27E+04	1.09E+00	1.35E+01	6.80E+01	3.09E+01	2.89E+01
3.00E+05	Best	5.49E+01	1.60E+02	3.36E+01	5.09E+04	8.21E+00	1.41E−01	2.00E+02	2.69E+02	2.46E+02
	7th	6.15E+01	1.99E+02	3.48E+01	8.65E+04	1.02E+01	1.28E+01	3.00E+02	2.90E+02	2.88E+02
	Median	6.53E+01	2.02E+02	3.59E+01	9.52E+04	1.08E+01	1.32E+01	3.00E+02	3.02E+02	3.04E+02
	19th	7.27E+01	2.10E+02	3.64E+01	1.05E+05	1.14E+01	1.34E+01	3.00E+02	3.18E+02	3.17E+02
	Worst	7.95E+01	2.24E+02	3.68E+01	1.24E+05	1.21E+01	1.34E+01	3.00E+02	3.52E+02	3.29E+02
	Mean	6.60E+01	2.01E+02	3.56E+01	9.55E+04	1.07E+01	1.36E+01	2.88E+02	3.06E+02	3.01E+02
	Std	6.74E+00	1.44E+01	8.84E−01	1.75E+04	9.32E−01	1.33E+01	3.25E+01	2.18E+01	2.00E+01
	MTE	6.60E+01	2.01E+02	3.56E+01	9.55E+04	1.07E+01	1.88E−01	2.88E+02	3. 06E+02	3.01E+02
	STE	6.74E+00	1.44E+01	8.84E−01	1.75E+04	9.32E−01	1.33E+01	3.25E+01	2. 18E+01	2.00E+01

Please cite this article in press as: B. Akay, D. Karaboga, A modified Artificial Bee Colony algorithm for real-parameter optimization, Inform. Sci. (2010), doi:10.1016/j.ins.2010.07.015

Table 13
D: 30, Colony Size: 30, "*limit*": 200, Cycle: 10,000, Max.FES: 300,000, MR = 0.4, MTE: Mean of Term. Err. of 25 runs, STE: Std of Term. Err. of 25 runs.

FES	Prob	18	19	20	21	22	23	24	25
1.00E+03	Best	1.21E+03	1.17E+03	1.21E+03	1.20E+03	1.13E+03	1.22E+03	2.03E+02	2.01E+02
	7th	1.28E+03	1.26E+03	1.29E+03	1.23E+03	1.24E+03	1.26E+03	2.04E+02	2.05E+02
	Median	1.32E+03	1.31E+03	1.31E+03	1.25E+03	1.36E+03	1.27E+03	2.05E+02	2.14E+02
	19th	1.36E+03	1.34E+03	1.35E+03	1.27E+03	1.40E+03	1.29E+03	2.08E+02	2.19E+02
	Worst	1.41E+03	1.41E+03	1.41E+03	1.29E+03	1.46E+03	1.32E+03	2.20E+02	7.05E+02
	Mean	1.31E+03	1.30E+03	1.32E+03	1.25E+03	1.33E+03	1.28E+03	2.07E+02	2.36E+02
	Std	5.11E+01	6.07E+01	4.58E+01	2.34E+01	9.25E+01	2.57E+01	4.42E+00	9.71E+01
1.00E+04	Best	1.01E+03	9.95E+02	1.00E+03	5.00E+02	9.15E+02	7.38E+02	2.01E+02	2.00E+02
	7th	1.03E+03	1.03E+03	1.04E+03	5.63E+02	9.40E+02	8.79E+02	2.01E+02	2.00E+02
	Median	1.04E+03	1.05E+03	1.06E+03	7.69E+02	9.67E+02	9.15E+02	2.01E+02	2.00E+02
	19th	1.06E+03	1.07E+03	1.07E+03	8.28E+02	9.81E+02	9.70E+02	2.01E+02	2.00E+02
	Worst	1.08E+03	1.13E+03	1.11E+03	9.52E+02	1.01E+03	1.03E+03	2.01E+02	2.00E+02
	Mean	1.04E+03	1.05E+03	1.06E+03	7.24E+02	9.62E+02	9.18E+02	2.01E+02	2.00E+02
	Std	2.13E+01	3.48E+01	2.66E+01	1.46E+02	2.56E+01	6.50E+01	6.52E−02	4.92E−02
1.00E+05	Best	8.00E+02	8.00E+02	8.00E+02	5.00E+02	8.91E+02	5.45E+02	2.01E+02	2.00E+02
	7th	8.43E+02	8.03E+02	8.59E+02	5.00E+02	9.06E+02	8.44E+02	2.01E+02	2.00E+02
	Median	9.70E+02	9.58E+02	9.57E+02	6.45E+02	9.12E+02	8.46E+02	2.01E+02	2.00E+02
	19th	9.77E+02	9.69E+02	9.74E+02	8.00E+02	9.19E+02	8.47E+02	2.01E+02	2.00E+02
	Worst	9.93E+02	9.90E+02	1.01E+03	8.00E+02	9.28E+02	8.49E+02	2.01E+02	2.00E+02
	Mean	9.26E+02	9.13E+02	9.22E+02	6.42E+02	9.12E+02	8.22E+02	2.01E+02	2.00E+02
	Std	7.31E+01	7.68E+01	6.80E+01	1.41E+02	9.31E+00	7.86E+01	5.05E−02	4.62E−03
3.00E+05	Best	8.00E+02	8.00E+02	8.00E+02	5.00E+02	8.91E+02	5.42E+02	2.01E+02	2.00E+02
	7th	8.00E+02	8.00E+02	8.00E+02	5.00E+02	8.99E+02	8.44E+02	2.01E+02	2.00E+02
	Median	8.00E+02	8.00E+02	8.00E+02	6.45E+02	9.04E+02	8.44E+02	2.01E+02	2.00E+02
	19th	8.00E+02	8.00E+02	8.00E+02	8.00E+02	9.07E+02	8.44E+02	2.01E+02	2.00E+02
	Worst	9.71E+02	9.45E+02	9.80E+02	8.00E+02	9.17E+02	8.46E+02	2.01E+02	2.00E+02
	Mean	8.12E+02	8.17E+02	8.23E+02	6.42E+02	9.04E+02	8.20E+02	2.01E+02	2.00E+02
	Std	4.07E+01	4.50E+01	5.31E+01	1.41E+02	6.01E+00	8.13E+01	5.45E−02	2.23E−03
	MTE	8.12E+02	8.17E+02	8.23E+02	6.42E+02	9.04E+02	8.20E+02	2.01E+02	2. 00E+02
	STE	4.07E+01	4.50E+01	5.31E+01	1.41E+02	6.01E+00	8.13E+01	5.45E−02	2. 23E−03

Table 14
D: 50, Colony Size: 50, "*limit*": 200, Cycle: 10,000, Max.FES: 500,000, MR = 0.4, MTE: Mean of Term. Err. of 25 runs, STE: Std of Term. Err. of 25 runs.

FES	Prob	1	2	3	4	5	6	7	8
1.00E+03	Best	5.97E+04	1.15E+06	4.73E+06	2.51E+06	5.12E+04	3.16E+10	7.03E+03	2.12E+01
	7th	7.98E+04	1.70E+06	5.81E+06	3.02E+06	5.92E+04	4.21E+10	8.11E+03	2.13E+01
	Median	8.91E+04	1.85E+06	6.71E+06	3.44E+06	6.22E+04	5.49E+10	9.00E+03	2.13E+01
	19th	1.00E+05	2.09E+06	7.39E+06	3.90E+06	6.52E+04	5.85E+10	9.77E+03	2.14E+01
	Worst	1.18E+05	2.37E+06	8.65E+06	4.57E+06	7.18E+04	7.53E+10	1.06E+04	2.14E+01
	Mean	8.90E+04	1.85E+06	6.62E+06	3.43E+06	6.23E+04	5.20E+10	8.94E+03	2.13E+01
	Std	1.43E+04	3.01E+05	1.09E+06	5.49E+05	5.08E+03	1.10E+10	9.65E+02	4.73E−02
1.00E+04	Best	1.01E+03	1.85E+04	1.79E+06	2.41E+05	2.19E+04	1.16E+08	2.14E+02	2.12E+01
	7th	1.46E+03	2.87E+04	2.17E+06	2.93E+05	2.74E+04	1.95E+08	2.79E+02	2.12E+01
	Median	1.57E+03	3.15E+04	2.34E+06	3.25E+05	2.94E+04	2.58E+08	3.16E+02	2.12E+01
	19th	1.78E+03	3.57E+04	2.49E+06	3.37E+05	3.14E+04	3.26E+08	3.50E+02	2.13E+01
	Worst	2.58E+03	4.37E+04	2.95E+06	4.33E+05	3.36E+04	4.87E+08	4.36E+02	2.13E+01
	Mean	1.65E+03	3.19E+04	2.35E+06	3.23E+05	2.90E+04	2.66E+08	3.20E+02	2.12E+01
	Std	3.48E+02	5.46E+03	2.45E+05	5.11E+04	3.19E+03	9.09E+07	4.84E+01	4.29E−02
1.00E+05	Best	0.00E+00	0.00E+00	1.01E+06	1.31E+02	1.05E+04	7.40E+02	7.95E−01	2.11E+01
	7th	0.00E+00	0.00E+00	1.19E+06	7.13E+02	1.22E+04	4.55E+03	8.76E−01	2.12E+01
	Median	0.00E+00	0.00E+00	1.38E+06	1.44E+03	1.31E+04	6.45E+03	9.19E−01	2.12E+01
	19th	0.00E+00	0.00E+00	1.41E+06	2.63E+03	1.42E+04	1.28E+04	9.36E−01	2.12E+01
	Worst	0.00E+00	0.00E+00	1.54E+06	9.84E+03	1.55E+04	2.38E+04	9.73E−01	2.12E+01
	Mean	0.00E+00	0.00E+00	1.32E+06	2.13E+03	1.32E+04	9.37E+03	9.05E−01	2.12E+01
	Std	0.00E+00	0.00E+00	1.40E+05	2.23E+03	1.39E+03	6.53E+03	4.46E−02	3.37E−02
5.00E+05	Best	0.00E+00	0.00E+00	9.76E+05	4.49E+00	7.93E+03	7.40E+02	7.18E−01	2.11E+01
	7th	0.00E+00	0.00E+00	1.07E+06	1.57E+02	9.67E+03	1.69E+03	7.81E−01	2.11E+01
	Median	0.00E+00	0.00E+00	1.10E+06	2.27E+02	1.03E+04	2.11E+03	8.03E−01	2.11E+01
	19th	0.00E+00	0.00E+00	1.22E+06	4.59E+02	1.09E+04	3.40E+03	8.54E−01	2.12E+01
	Worst	0.00E+00	0.00E+00	1.31E+06	1.81E+03	1.20E+04	4.55E+03	8.79E−01	2.12E+01
	Mean	0.00E+00	0.00E+00	1.13E+06	3.82E+02	1.03E+04	2.47E+03	8.10E−01	2.11E+01
	Std	0.00E+00	0.00E+00	9.38E+04	3.98E+02	9.13E+02	1.10E+03	4.70E−02	2.65E−02
	MTE	9.34E−09	9.28E−09	1.13E+06	3.82E+02	1.03E+04	2.47E+03	8.10E−01	2. 11E+01
	STE	6.05E−10	6.19E−10	9.38E+04	3.98E+02	9.13E+02	1.10E+03	4.70E−02	2.65E−02

Please cite this article in press as: B. Akay, D. Karaboga, A modified Artificial Bee Colony algorithm for real-parameter optimization, Inform. Sci. (2010), doi:10.1016/j.ins.2010.07.015

Table 15
D: 50, Colony Size: 50, "*limit*": 200, Cycle: 10,000, Max.FES: 500,000, MR = 0.4, MTE: Mean of Term. Err. of 25 runs, STE: Std of Term. Err. of 25 runs.

FES	Prob	9	10	11	12	13	14	15	16	17
1.00E+03	Best	6.10E+02	9.08E+02	7.71E+01	4.09E+06	5.91E+02	2.35E+01	8.11E+02	5.30E+02	6.17E+02
	7th	6.77E+02	1.03E+03	7.89E+01	4.44E+06	9.04E+02	2.40E+01	9.78E+02	6.27E+02	7.30E+02
	Median	7.26E+02	1.11E+03	8.08E+01	4.77E+06	1.07E+03	2.40E+01	1.03E+03	6.70E+02	7.68E+02
	19th	7.58E+02	1.18E+03	8.15E+01	5.24E+06	1.36E+03	2.41E+01	1.05E+03	7.56E+02	8.27E+02
	Worst	8.02E+02	1.30E+03	8.24E+01	5.78E+06	2.19E+03	2.43E+01	1.08E+03	8.20E+02	9.87E+02
	Mean	7.17E+02	1.11E+03	8.02E+01	4.86E+06	1.15E+03	2.40E+01	1.01E+03	6.82E+02	7.91E+02
	Std	4.83E+01	1.04E+02	1.58E+00	4.79E+05	4.50E+02	1.84E−01	7.04E+01	8.14E+01	9.44E+01
1.00E+04	Best	3.41E+02	4.65E+02	7.26E+01	1.18E+06	4.73E+01	2.33E+01	5.68E+02	3.35E+02	3.53E+02
	7th	3.50E+02	5.61E+02	7.51E+01	1.50E+06	6.00E+01	2.36E+01	5.80E+02	3.55E+02	3.99E+02
	Median	3.61E+02	5.82E+02	7.58E+01	1.68E+06	6.32E+01	2.37E+01	5.92E+02	3.83E+02	4.16E+02
	19th	3.68E+02	5.97E+02	7.67E+01	1.80E+06	6.86E+01	2.38E+01	6.40E+02	3.97E+02	4.29E+02
	Worst	3.98E+02	6.13E+02	7.86E+01	1.94E+06	7.61E+01	2.39E+01	7.19E+02	4.40E+02	4.75E+02
	Mean	3.62E+02	5.74E+02	7.58E+01	1.65E+06	6.35E+01	2.37E+01	6.15E+02	3.80E+02	4.17E+02
	Std	1.60E+01	3.14E+01	1.43E+00	1.89E+05	7.23E+00	1.54E−01	4.49E+01	2.88E+01	2.80E+01
1.00E+05	Best	2.36E+02	4.49E+02	6.95E+01	8.33E+05	3.21E+01	2.30E+01	2.01E+02	3.00E+02	2.98E+02
	7th	2.74E+02	4.81E+02	7.11E+01	9.70E+05	3.66E+01	2.33E+01	2.18E+02	3.24E+02	3.26E+02
	Median	2.82E+02	4.93E+02	7.23E+01	1.07E+06	3.81E+01	2.34E+01	3.00E+02	3.46E+02	3.39E+02
	19th	2.86E+02	5.02E+02	7.31E+01	1.14E+06	3.91E+01	2.34E+01	3.05E+02	3.58E+02	3.54E+02
	Worst	2.98E+02	5.06E+02	7.43E+01	1.21E+06	4.28E+01	2.36E+01	5.12E+02	3.86E+02	4.10E+02
	Mean	2.77E+02	4.88E+02	7.21E+01	1.04E+06	3.81E+01	2.34E+01	2.97E+02	3.44E+02	3.39E+02
	Std	1.39E+01	1.60E+01	1.30E+00	1.10E+05	2.41E+00	1.41E−01	8.91E+01	2.39E+01	2.43E+01
5.00E+05	Best	2.36E+02	4.31E+02	6.62E+01	7.31E+05	2.91E+01	2.29E+01	2.00E+02	2.68E+02	2.81E+02
	7th	2.51E+02	4.49E+02	6.99E+01	8.33E+05	3.42E+01	2.31E+01	2.00E+02	3.24E+02	3.01E+02
	Median	2.61E+02	4.57E+02	7.03E+01	8.91E+05	3.54E+01	2.32E+01	2.00E+02	3.44E+02	3.10E+02
	19th	2.64E+02	4.68E+02	7.09E+01	9.44E+05	3.66E+01	2.33E+01	3.00E+02	3.57E+02	3.17E+02
	Worst	2.76E+02	4.88E+02	7.25E+01	1.05E+06	3.75E+01	2.33E+01	4.00E+02	3.86E+02	3.30E+02
	Mean	2.59E+02	4.58E+02	7.03E+01	8.88E+05	3.50E+01	2.32E+01	2.52E+02	3.39E+02	3.08E+02
	Std	9.91E+00	1.41E+01	1.44E+00	8.77E+04	1.98E+00	1.13E−01	5.74E+01	3.16E+01	1.24E+01
	MTE	2.59E+02	4.58E+02	7.03E+01	8.88E+05	3.50E+01	2.32E+01	2.52E+02	3.39E+02	3.08E+02
	STE	9.91E+00	1.41E+01	1.44E+00	8.77E+04	1.98E+00	1.13E−01	5.74E+01	3.16E+01	1.24E+01

Table 16
D: 50, Colony Size: 50, "*limit*": 200, Cycle: 10,000, Max.FES: 500,000, MR = 0.4, MTE: Mean of Term. Err. of 25 runs, STE: Std of Term. Err. of 25 runs.

FES	Prob	18	19	20	21	22	23	24	25
1.00E+03	Best	1.24E+03	1.30E+03	1.29E+03	1.33E+03	1.19E+03	1.36E+03	2.10E+02	4.55E+02
	7th	1.29E+03	1.31E+03	1.33E+03	1.37E+03	1.23E+03	1.39E+03	2.39E+02	8.13E+02
	Median	1.33E+03	1.36E+03	1.36E+03	1.40E+03	1.26E+03	1.41E+03	2.75E+02	1.24E+03
	19th	1.34E+03	1.37E+03	1.38E+03	1.42E+03	1.32E+03	1.43E+03	3.44E+02	1.44E+03
	Worst	1.38E+03	1.42E+03	1.40E+03	1.46E+03	1.35E+03	1.48E+03	4.21E+02	1.58E+03
	Mean	1.32E+03	1.36E+03	1.36E+03	1.40E+03	1.27E+03	1.41E+03	2.90E+02	1.16E+03
	Std	3.75E+01	3.66E+01	2.89E+01	3.39E+01	5.05E+01	2.78E+01	6.09E+01	3.56E+02
1.00E+04	Best	1.04E+03	1.04E+03	1.03E+03	9.47E+02	9.08E+02	1.00E+03	2.01E+02	2.01E+02
	7th	1.07E+03	1.05E+03	1.06E+03	1.03E+03	9.41E+02	1.06E+03	2.01E+02	2.02E+02
	Median	1.09E+03	1.05E+03	1.07E+03	1.04E+03	9.54E+02	1.09E+03	2.01E+02	2.02E+02
	19th	1.09E+03	1.07E+03	1.09E+03	1.05E+03	9.65E+02	1.11E+03	2.01E+02	2.03E+02
	Worst	1.12E+03	1.10E+03	1.10E+03	1.06E+03	9.84E+02	1.16E+03	2.02E+02	2.05E+02
	Mean	1.08E+03	1.06E+03	1.07E+03	1.04E+03	9.52E+02	1.09E+03	2.01E+02	2.02E+02
	Std	1.71E+01	1.54E+01	1.89E+01	2.47E+01	1.69E+01	3.72E+01	2.92E−01	8.16E−01
1.00E+05	Best	9.88E+02	9.77E+02	9.76E+02	5.00E+02	8.77E+02	5.82E+02	2.01E+02	2.00E+02
	7th	9.90E+02	9.80E+02	9.80E+02	5.00E+02	8.86E+02	5.89E+02	2.01E+02	2.01E+02
	Median	9.91E+02	9.82E+02	9.81E+02	9.92E+02	8.89E+02	5.98E+02	2.01E+02	2.01E+02
	19th	9.93E+02	9.84E+02	9.85E+02	1.01E+03	8.92E+02	6.07E+02	2.01E+02	2.01E+02
	Worst	1.00E+03	9.88E+02	9.89E+02	1.01E+03	9.06E+02	6.12E+02	2.01E+02	2.02E+02
	Mean	9.92E+02	9.82E+02	9.82E+02	8.35E+02	8.90E+02	5.97E+02	2.01E+02	2.01E+02
	Std	3.00E+00	3.07E+00	3.70E+00	2.32E+02	5.95E+00	1.01E+01	2.19E−02	4.54E−01
5.00E+05	Best	9.80E+02	9.66E+02	9.67E+02	5.00E+02	8.59E+02	5.58E+02	2.00E+02	2.00E+02
	7th	9.84E+02	9.69E+02	9.69E+02	5.00E+02	8.74E+02	5.66E+02	2.01E+02	2.01E+02
	Median	9.86E+02	9.70E+02	9.69E+02	9.92E+02	8.76E+02	5.70E+02	2.01E+02	2.01E+02
	19th	9.87E+02	9.71E+02	9.70E+02	1.01E+03	8.78E+02	5.74E+02	2.01E+02	2.01E+02
	Worst	9.90E+02	9.73E+02	9.72E+02	1.01E+03	8.84E+02	5.86E+02	2.01E+02	2.02E+02
	Mean	9.85E+02	9.70E+02	9.70E+02	8.34E+02	8.75E+02	5.71E+02	2.01E+02	2.01E+02
	Std	2.12E+00	1.38E+00	1.22E+00	2.31E+02	4.83E+00	7.12E+00	3.27E−02	4.54E−01
	MTE	9.85E+02	9.70E+02	9.70E+02	8.34E+02	8.75E+02	5.71E+02	2.01E+02	2.01E+02
	STE	2.12E+00	1.38E+00	1.22E+00	2.31E+02	4.83E+00	7.12E+00	3.27E−02	4.54E−01

Please cite this article in press as: B. Akay, D. Karaboga, A modified Artificial Bee Colony algorithm for real-parameter optimization, Inform. Sci. (2010), doi:10.1016/j.ins.2010.07.015

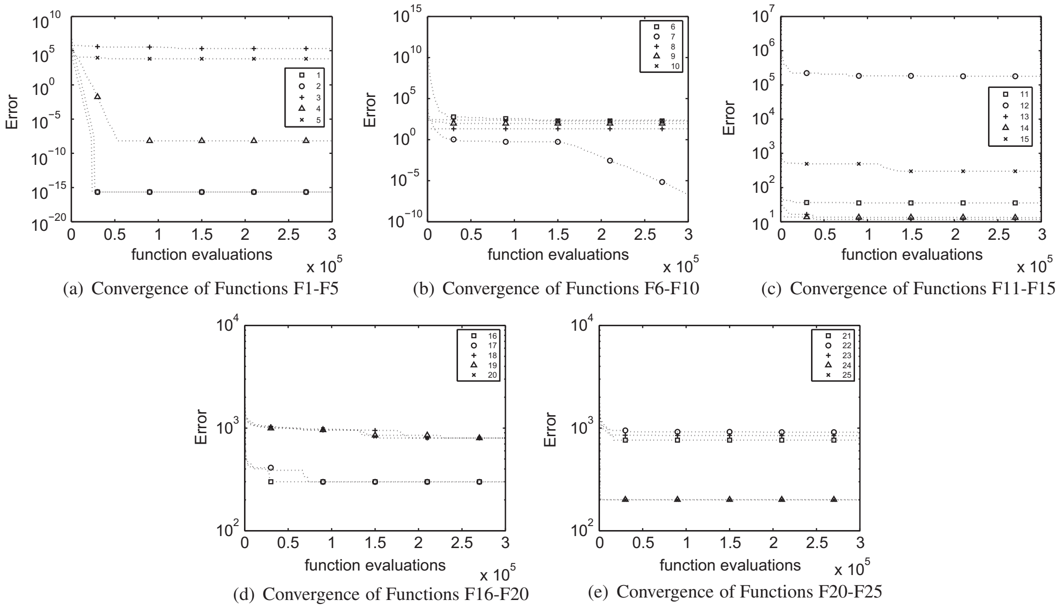


Fig. 4. Logarithmic scaled best function error of median runs of 25 runs for 30-dimension of 25 functions in the set. Numbers in the legend correspond to the function number.

Table 17
Results of state-of-art algorithms for hybrid functions, Dimension = 10, Colony Size = 20, Cycle = 5000 for ABC, PSO-RDL: Recombination with Dynamic Linkage Discovery in PSO [17], DMS-PSO: Dynamic multi-swarm particle swarm optimizer with local search [33], SPC-PNX [4], DE: Differential Evolution [50], SADE: Self-adaptive Differential Evolution [43], restart CMA-ES: Restart Covariance Matrix Adaptation Evolution Strategy with Increasing Population Size [1].

		F1	F2	F3	F4	F5
PSO-RDL	Mean	2.50E−14	1.77E−13	9.68E−02	2.47E−07	2.09E−07
	Std.	2.88E−14	2.03E−13	3.34E−01	7.07E−07	7.22E−07
DMS-PSO	Mean	0.00E+00	1.30E−13	7.01E−09	1.89E−03	1.14E−06
	Std.	0.00E+00	1.56E−13	2.66E−09	1.89E−03	2.18E−06
SPC-PNX	Mean	8.90E−09	9.63E−09	1.08E+05	9.38E−09	9.15E−09
	Std.	9.39E−10	3.30E−10	8.72E+04	6.33E−10	6.32E−10
DE	Mean	0.00E+00	0.00E+00	1.94E−06	9.09E−15	0.00E+00
	Std.	0.00E+00	0.00E+00	4.63E−06	3.15E−14	0.00E+00
SaDE	Mean	0.00E+00	1.05E−13	1.67E−05	1.42E−05	1.23E−02
	Std.	0.00E+00	5.11E−13	3.12E−05	7.09E−05	1.46E−02
Restart CMA-ES	Mean	5.20E−09	4.70E−09	5.60E−09	5.02E−09	6.58E−09
	Std.	1.94E−09	1.56E−09	1.93E−09	1.71E−09	2.17E−09
ABC	Mean	4.89E−17	4.81E−17	2.50E+03	1.50E−16	5.82E+01
	Std.	7.23E−18	5.89E−18	8.68E+02	5.47E−17	4.11E+01
ABC (SF = 0.7)	Mean	8.85E−17	1.04E−16	1.37E+03	1.29E−04	1.32E+02
	Std.	5.36E−17	4.90E−17	5.54E+02	3.05E−04	1.09E+02
ABC (SF = 0.5)	Mean	1.16E−16	1.14E−16	9.90E+02	1.25E−02	3.26E+02
	Std.	4.91E−17	4.60E−17	4.62E+02	1.25E−02	3.30E+02
ABC (SF = 0.3)	Mean	1.52E−16	1.83E−16	5.88E+02	1.29E−01	8.07E+02
	Std.	3.43E−17	8.43E−17	2.79E+02	9.46E−02	8.12E+02
ABC (MR = 1)	Mean	4.64E−17	4.62E−17	1.52E+03	4.61E−17	7.28E−13
	Std.	6.91E−18	9.16E−18	7.67E+02	6.95E−18	1.15E−12
ABC (MR = 0.8)	Mean	8.80E−17	1.10E−16	3.57E+03	8.24E−17	1.82E−12
	Std.	4.84E−17	4.66E−17	1.11E+03	4.73E−17	1.46E−12
ABC (MR = 0.6)	Mean	4.75E−17	4.58E−17	3.72E+03	4.82E−17	1.89E−12
	Std.	6.79E−18	7.75E−18	1.29E+03	6.45E−18	1.74E−12
ABC (MR = 0.4)	Mean	4.48E−17	4.30E−17	3.58E+03	4.48E−17	6.18E−12
	Std.	8.91E−18	9.77E−18	1.60E+03	7.31E−18	5.52E−12
ABC (MR = 0.2)	Mean	4.80E−17	4.89E−17	3.22E+03	6.98E−17	1.66E+00
	Std.	5.46E−18	4.62E−18	2.14E+03	4.05E−17	1.61E+00
ABC (ASF-MR: 0.9)	Mean	4.94E−17	4.76E−17	5.23E−04	4.41E−12	4.12E−04
	Std.	4.29E−18	8.58E−18	5.23E−04	1.32E−11	6.31E−04

The pseudo-code of the ABC algorithm is given below:

```
Initialize the population of solutions x_{i,j}, i = 1...SN, j = 1...D, trial_i = 0 trial_i is the non-improvement number of the
  solution x_i, used for abandonment
Evaluate the population
cycle = 1
repeat
   {——Produce a new food source population for Employed bee——}
  for i = 1 to SN do
     Produce a new food source υ_i for the employed bee of the food source x_i by using (2) (in the case of modified ABC
     algorithm by using (5)) and evaluate its quality
     Apply a greedy selection process between υ_i and x_i and select the better one
     If solution x_i does not improve trial_i = trial_i + 1, otherwise trial_i = 0
  end for
  Calculate the probability values p_i by (4) for the solutions using fitness values
   {——Produce a new food source population for onlookers——-}
  t = 0,i = 1
  repeat
     if random < p_i then
       Produce a new υ_ij food source by (2) (in the case of the modified ABC algorithm by using (5)) for the onlooker
  bee
       Apply a greedy selection process between υ_i and x_i and select the better one
       If solution x_i does not improve trial_i = trial_i + 1, otherwise trial_i = 0
       t = t + 1
     end if
  until (t = SN)
   {——Determine Scout——}
  if max(trial_i) > limit then
     Replace x_i with a new randomly produced solution by (1)
  end if
     Memorize the best solution achieved so far
     cycle = cycle+1
until (cycle = Maximum Cycle Number)
```

5. Experiments and discussion

We have tested the ABC algorithm and its variants in two groups of functions. The first group consists of basic functions, and the second one has one set of composite functions.

5.1. Experiments on basic functions

In the first part of the experiments, in order to assess the performance of the ABC algorithm, we considered basic functions used in [32] and given in Table 1. There are two groups of functions in the table. The first group consists of unimodal functions: Sphere and Rosenbrock. Sphere function is a continuous, convex and unimodal function. Since the global optimum of the Rosenbrock function is inside a long, narrow, parabolic-shaped flat valley, and the variables are dependent, the gradients generally do not point towards the optimum, and it is difficult to converge the global optimum. The second group consists of multimodal functions: Ackley, Griewank, Weierstrass, Rastrigin, Noncontinuous Rastrigin and Schwefel. The Ackley function has a surface with many local optima due to its exponential term. The variables of Griewank function have interdependence since the function has a product term. The multimodality is removed by the increment in dimensionality ($n > 30$) and the problem seems unimodal. The Weierstrass function is continuous everywhere but differentiable nowhere. Non-Continuous and Continuous Rastrigin functions are based on the Sphere function with the addition of cosine modulation to produce many local minima. The surface of Schwefel function is composed of numerous peaks and valleys. The second best minimum of the function is far from the global minimum, and the global minimum is near the boundaries of the search domain [10,41].

In the experiments, the population size was 10, and the maximum number of function evaluations was 30,000 for 10-dimensional problems. All experiments were conducted 30 times, independently for each function. While making this comprehensive study, we examined the ABC algorithm under different control parameter settings (number of parameters changed in each cycle, step size in the production of neighboring solutions and "*limit*"). The results obtained by ABC algorithm with different control parameters are compared against the results of PSO variants presented in [32]. Settings of the PSO variants can be found in [32]. PSO-w (PSO with inertia factor), PSO-cf (PSO with

Table 18
Results of state-of-art algorithms for hybrid functions, Dimension = 10, Colony Size = 20, Cycle = 5000 for ABC, PSO-RDL: Recombination with Dynamic Linkage Discovery in PSO [17], DMS-PSO: Dynamic multi-swarm particle swarm optimizer with local search [33], SPC-PNX [4], DE: Differential Evolution [50], SADE: Self-adaptive Differential Evolution [43], Restart CMA-ES: Restart Covariance Matrix Adaptation Evolution Strategy with Increasing Population Size [1].

		F6	F7	F8	F9	F10
PSO-RDL	Mean	9.57E−01	5.73E−02	2.00E+01	1.25E+01	3.86E+01
	Std.	1.74E+00	4.66E−02	2.17E−04	8.17E+00	1.80E+01
DMS-PSO	Mean	6.89E−08	4.52E−02	2.00E+01	0.00E+00	3.62E+00
	Std.	3.19E−07	3.26E−02	5.54E−09	0.00E+00	8.55E−01
SPC-PNX	Mean	1.89E+01	8.26E−02	2.10E+01	4.02E+00	7.30E+00
	Std.	4.00E+01	6.24E−02	5.79E−02	2.27E+00	5.21E+00
DE	Mean	1.59E−01	1.46E−01	2.04E+01	9.55E−01	1.25E+01
	Std.	7.97E−01	1.38E−01	7.58E−02	9.73E−01	7.96E+00
SaDE	Mean	1.20E−08	1.99E−02	2.00E+01	0.00E+00	4.97E+00
	Std.	1.93E−08	1.07E−02	5.39E−08	0.00E+00	1.69E+00
Restart CMA-ES	Mean	4.87E−09	3.31E−09	2.00E+01	2.39E−01	7.96E−02
	Std.	1.66E−09	2.02E−09	3.89E−03	4.34E−01	2.75E−01
ABC (SF = 1)	Mean	3.31E+00	2.52E−01	2.03E+01	4.87E−17	2.22E+01
	Std.	5.18E+00	9.29E−02	6.07E−02	1.79E−17	7.32E+00
ABC (SF = 0.7)	Mean	8.89E+00	2.28E−01	2.03E+01	4.00E−01	3.77E+01
	Std.	1.40E+01	7.54E−02	5.67E−02	4.86E−01	8.98E+00
ABC (SF = 0.5)	Mean	9.48E+00	2.27E−02	2.03E+01	3.21E−01	5.13E+01
	Std.	9.25E+00	8.10E−02	7.24E−02	4.62E−01	8.56E+00
ABC (SF = 0.3)	Mean	1.37E+01	2.49E−01	2.03E+01	5.28E+00	7.50E+01
	Std.	1.36E+01	1.08E−01	5.88E−01	2.32E+00	1.65E+01
ABC (MR = 1)	Mean	1.27E+00	4.84E−01	2.04E+01	1.20E+01	2.92E+01
	Std.	1.38E+00	9.07E−02	7.50E−02	3.56E+00	4.33E+00
ABC (MR = 0.8)	Mean	6.02E+00	5.20E−01	2.03E+01	2.35E+01	3.85E+01
	Std.	1.62E+01	8.40E−02	6.33E−02	3.59E+00	4.09E+00
ABC (MR = 0.6)	Mean	4.95E+00	3.01E−01	2.04E+01	1.83E−01	2.41E+01
	Std.	4.00E+00	7.15E−02	7.40E−02	4.31E−01	4.37E+00
ABC (MR = 0.4)	Mean	2.76E+00	1.27E−02	2.04E+01	5.29E−17	1.43E+01
	Std.	3.62E+00	2.91E−01	7.67E−02	2.33E−17	2.61E+00
ABC (MR = 0.2)	Mean	2.51E+00	9.48E−02	2.04E+01	5.00E−17	1.69E+00
	Std.	5.61E+00	3.73E−02	6.79E−02	1.93E−17	4.62E+00
ABC (ASF-MR: 0.9)	Mean	3.36E+00	6.52E−02	2.00E+01	7.65E+00	1.25E+01
	Std.	2.49E+00	1.44E−02	1.28E−02	2.15E+00	2.93E+00

constriction factor), PSO-w-local (PSO with inertia factor using a local neighborhood), PSO-cf-local (PSO with constriction factor using local neighborhood), UPSO (Unified PSO combining local and global neighborhood topologies), FDR-PSO (Fitness Distance Ratio based PSO uses the neighbor with higher fitness), FIPS (Fully Informed Particle Swarm), CPSO-H (Cooperative PSO), CLPSO (Comprehensive Learning PSO) are the versions of PSO which are investigated in [32].

While making an experimental comparison, if the difference of error rates is less than 10^{-7}, this difference is considered as insignificant in a practical sense [14]. The winner algorithm for each problem was given in boldface in the tables if it was significantly different. Comparison results obtained by PSO variants and the ABC algorithm with different control parameters set for Sphere, Rosenbrock, Ackley and Griewank functions are given in Table 2 and for Weierstrass, Rastrigin, Non-continuous Rastrigin and Schwefel functions are given in Table 3. From the results, for the Rosenbrock function, the ABC algorithm that adjusts step size automatically produces the best result. For the Griewank and Schwefel functions, CLPSO performs the best. Of the other functions (Sphere, Ackley, Rastrigin, NCRastrigin), CLPSO and basic ABC algorithms exhibit similar performance. For all functions, except the Rosenbrock function, initialization ranges for the algorithms were different from the search ranges. Initial ranges and search ranges for functions can be found in Table 1. The ABC algorithm is robust against initialization conditions since the scout unit helps the search space to be explored efficiently.

The modification rate (MR), scaling factor (SF) and "*limit*" are control parameters of the ABC algorithm which needed to be tuned for better performance. We investigated the effect of the control parameters on the performance of the ABC algorithm by manually trying some different values before the run. The results presented in Tables 2,3 were demonstrated in the Fig. 2(a)–(f). From these figures, generally, the basic version of the ABC algorithm in which MR = 0 (just one parameter is changed), and SF is 1 (ϕ is in the range $[-1,1]$) is a better choice than other structures tried out for both unimodal and multimodal basic functions. MR parameter is more important for the ABC algorithm on hybrid functions which are rotated and shifted versions of function combinations. For "*limit*" control parameter, 200 is more appropriate than other values for both unimodal and multimodal functions. This value can change depending on the dimension of the problem. Convergence rates of the basic ABC algorithm are shown in Fig. 3(a) for unimodal basic functions and in Fig. 3(b) for multimodal basic functions.

Table 19
Results of state-of-art algorithms for hybrid functions, Dimension = 10, Colony Size = 20, Cycle = 5000 for ABC, PSO-RDL: Recombination with Dynamic Linkage Discovery in PSO [17], DMS-PSO: Dynamic multi-swarm particle swarm optimizer with local search [33], SPC-PNX [4], DE: Differential Evolution [50], SADE: Self-adaptive Differential Evolution [43], restart CMA-ES: Restart Covariance Matrix Adaptation Evolution Strategy with Increasing Population Size [1].

		F11	F12	F13	F14	F15
PSO-RDL	Mean	5.58E+00	1.31E+02	8.87E−01	3.78E+00	2.71E+02
	Std.	1.42E+00	4.50E+02	4.06E−01	3.44E−01	1.59E+02
DMS-PSO	Mean	4.62E+00	2.40E+00	3.69E−01	2.36E+00	4.85E+00
	Std.	5.84E−01	4.36E+00	5.64E−02	3.38E−01	1.34E+01
SPC-PNX	Mean	1.91E+00	2.60E+02	8.38E−01	3.05E+00	2.54E+02
	Std.	1.16E+00	4.89E+02	2.69E−01	4.37E−01	1.51E+02
DE	Mean	8.47E−01	3.17E+01	9.77E−01	3.45E+00	2.59E+02
	Std.	1.40E+00	4.20E+01	4.67E−01	4.40E−01	1.83E+02
SaDE	Mean	4.89E+00	4.50E−07	2.20E−01	2.92E+00	3.20E+01
	Std.	6.62E−01	8.51E−07	4.11E−02	2.06E−01	1.11E+02
Restart CMA-ES	Mean	9.34E−01	2.93E+01	6.96E−01	3.01E+00	2.28E+02
	Std.	9.00E−01	1.42E+02	1.50E−01	3.49E−01	6.80E+01
ABC (SF = 1)	Mean	5.46E+00	9.85E+01	2.96E−02	3.41E+00	1.53E−01
	Std.	5.84E−01	6.16E+01	2.12E−02	1.53E−01	3.34E−01
ABC (SF = 0.7)	Mean	5.67E+00	1.16E+02	5.87E−02	3.44E+00	2.25E+00
	Std.	7.24E−01	1.02E+02	2.80E−02	1.73E−01	9.89E+00
ABC (SF = 0.5)	Mean	5.75E+00	1.10E+02	1.03E−01	3.36E+00	5.29E−01
	Std.	7.21E−01	8.68E+01	5.83E−02	1.45E−01	2.50E+00
ABC (SF = 0.3)	Mean	6.09E+00	1.52E+02	1.64E−01	3.53E+00	1.17E+01
	Std.	9.60E−01	3.41E+02	8.43E−02	1.59E−01	2.08E+01
ABC (MR = 1)	Mean	8.38E+00	1.30E+02	1.77E+00	3.66E+00	3.01E+02
	Std.	6.64E−01	1.76E+02	3.52E−01	1.45E−01	7.66E+01
ABC (MR = 0.8)	Mean	6.88E+00	1.72E+03	1.48E+00	3.59E+00	2.87E+02
	Std.	5.20E−01	5.35E+02	3.36E−01	1.12E−01	3.49E+01
ABC (MR = 0.6)	Mean	6.79E+00	4.22E+02	7.52E−01	3.56E+00	1.99E+02
	Std.	5.19E−01	2.10E+01	3.17E−01	1.44E−01	4.73E+01
ABC (MR = 0.4)	Mean	5.58E+00	2.22E+02	2.05E−01	3.49E+00	1.08E+02
	Std.	6.38E−01	1.80E+02	9.97E−02	1.27E−01	4.34E+01
ABC (MR = 0.2)	Mean	5.26E+00	1.55E+02	6.91E−02	3.35E+00	1.71E+01
	Std.	5.84E−01	6.58E+01	5.28E−02	2.12E−01	3.15E+01
ABC (ASF-MR: 0.9)	Mean	2.39E+00	1.71E+01	6.32E−01	3.24E+00	2.41E+02
	Std.	1.04E+00	1.46E+01	1.74E−01	2.95E−01	8.64E+01

5.2. On the scalability and time complexity of ABC

It is generally difficult for optimization algorithms to solve high dimensional problems. Performance of an algorithm deteriorates as the problem dimension increases. In order to cope with this problem, the algorithm needs more information about the search space to direct the solutions to the optima. Increasing the population size or the number of evaluations exponentially might improve the performance, but the performance in case of a high dimension problem is also related to the landscape of the problem. Moreover, it is more difficult for the algorithm to solve a high dimensional problem when there is an epistatic interaction between the parameters, many local optima, misleadingness and hard structural properties of the search space [28]. This is defined as scalability problem.

In order to analyze the scalability of ABC, we investigated the performance of ABC with respect to growing dimensions. For this experiment, functions given in Table 1 with different dimensions including 10, 50 and 100 were used. The scalability test was repeated for the colony sizes of 10, 20 and 50. Mean and standard deviation of errors for the functions for each case were reported in Table 4. When the problem dimension was increased from 10 to 50 and then to 100, the performance of the ABC algorithm was influenced from this change as expected. However, as seen from the table, an increase in problem dimension did not require exponential increment in population size or evaluation number. Therefore, it can be stated that the ABC algorithm is not very sensitive to increments in problem dimensions and has a good scalability.

The effect of scalability on the computational complexity of the ABC algorithm was also analyzed. For this purpose, time complexity of the ABC algorithm for Rosenbrock function with different dimensions was calculated as described in [54]. Rosenbrock function was chosen since it has interaction between its parameters. In order to determine the time complexity, after code execution time (T_0) and execution time of Rosenbrock function for 200,000 evaluations (T_1) were calculated, mean of five executions' time of ABC on Rosenbrock function through 200,000 evaluations ($\widehat{T}_2$) was computed. Then, the complexity of the algorithm was determined by $((\widehat{T}_2 - T_1)/T_0)$ and given in Table 5. Table 5 shows that $\widehat{T}_2$ increases by less than a factor of dimension increment. Consequently, it can be stated that the time complexity of the ABC algorithm does not depend on the problem dimension excessively and it scales with $O(n)$.

By onlookers, a new population is formed by searching the neighborhoods of the solutions chosen depending on their quality. Since the number of onlookers is equal to SN, in each cycle $2xSN$ searches are conducted by employed bees and onlooker bees. Hence, when the maximum cycle number (MCN) is reached, totally, $2xSNxMCN$ searches are carried out. So, the search complexity of ABC is proportional to $2xSNxMCN$.

Table 20
Results of state-of-art algorithms for hybrid functions, Dimension = 10, Colony Size = 20, Cycle = 5000 for ABC, PSO-RDL: Recombination with Dynamic Linkage Discovery in PSO [17], DMS-PSO: Dynamic multi-swarm particle swarm optimizer with local search [33], SPC-PNX [4], DE: Differential Evolution [50], SADE: Self-adaptive Differential Evolution [43], restart CMA-ES: Restart Covariance Matrix Adaptation Evolution Strategy with Increasing Population Size [1].

		F16	F17	F18	F19	F20
PSO-RDL	Mean	2.20E+02	2.22E+02	1.02E+03	9.85E+02	9.59E+02
	Std.	1.74E+02	1.00E+02	1.19E+02	1.02E+02	1.06E+02
DMS-PSO	Mean	9.48E+01	1.10E+02	7.61E+02	7.14E+02	8.22E+02
	Std.	1.01E+01	4.35E+00	1.85E+02	2.01E+02	4.59E+01
SPC-PNX	Mean	1.10E+02	1.19E+02	4.40E+02	3.80E+02	4.40E+02
	Std.	9.87E+00	1.07E+01	2.25E+02	1.87E+02	2.29E+02
DE	Mean	1.13E+02	1.15E+02	4.00E+02	4.20E+02	4.60E+02
	Std.	1.80E+01	2.01E+01	2.04E+02	2.18E+02	2.38E+02
SaDE	Mean	1.01E+02	1.14E+02	7.19E+02	7.05E+02	7.13E+02
	Std.	6.17E+00	9.97E+00	2.09E+02	1.90E+02	2.01E+02
Restart CMA-ES	Mean	9.13E+01	1.23E+02	3.32E+02	2.26E+02	3.00E+02
	Std.	3.49E+00	2.09E+01	1.12E+02	9.93E+01	0.00E+00
ABC (SF = 1)	Mean	1.75E+02	1.96E+02	4.46E+02	4.51E+02	4.38E+02
	Std.	2.11E+01	2.25E+01	4.83E+01	4.09E+01	3.30E+01
ABC (SF = 0.7)	Mean	1.77E+02	2.04E+02	3.86E+02	4.31E+02	4.09E+02
	Std.	1.64E+01	2.47E+01	9.60E+01	3.16E+01	2.66E+01
ABC (SF = 0.5)	Mean	2.12E+02	2.33E+02	4.16E+02	4.31E+02	4.39E+02
	Std.	2.49E+01	2.87E+01	4.40E+01	4.66E+01	2.97E+01
ABC (SF = 0.3)	Mean	2.93E+02	3.06E+02	4.75E+02	4.78E+02	4.61E+02
	Std.	4.40E+01	5.80E+01	1.32E+02	4.73E+01	4.80E+01
ABC (MR = 1)	Mean	1.98E+02	2.02E+02	5.50E+02	5.16E+02	5.04E+02
	Std.	1.20E+01	1.75E+01	1.13E+02	1.19E+02	1.07E+02
ABC (MR = 0.8)	Mean	2.25E+02	2.41E+02	4.62E+02	4.83E+02	4.62E+02
	Std.	1.17E+01	2.31E+01	2.15E+01	2.74E+01	3.12E+01
ABC (MR = 0.6)	Mean	2.02E+02	2.18E+02	5.03E+02	5.03E+02	5.19E+02
	Std.	1.79E+01	1.55E+01	4.60E+01	6.14E+01	5.19E+01
ABC (MR = 0.4)	Mean	1.75E+02	1.90E+02	4.61E+02	5.20E+02	5.18E+02
	Std.	1.80E+01	1.44E+01	4.14E+01	2.76E+01	3.23E+01
ABC (MR = 0.2)	Mean	1.59E+02	1.86E+02	4.33E+02	4.34E+02	4.24E+02
	Std.	1.71E+01	1.66E+01	4.21E+01	5.99E+01	3.22E+01
ABC (ASF-MR: 0.9)	Mean	1.85E+02	1.75E+02	3.71E+02	3.92E+02	4.30E+02
	Std.	3.62E+01	3.27E+01	5.78E+01	6.43E+01	7.00E+01

5.3. Experiments on composite functions

In the second part of the experiments, we tested the ABC algorithm and its modified versions on the real-parameter optimization problems defined in [54]. Some algorithms produce good results on some functions while they cannot achieve desired performance on some others. If an algorithm has an operator for producing neighboring solutions by copying one parameter to another, it can converge to the global optima quickly when the global optimum lies on symmetric dimensions. A similar situation is when the global optima are at the origin. When an algorithm has local search capability, finding the global optimum is simpler [34]. For this reason, in [54], composite problems are constructed by combining simple functions via the Gaussian function in order to obtain more challenging problems. Composition functions are randomly located, asymmetrical and multimodal problems. Functions in the set have different characteristics, and they are categorized in a systematic manner that will determine how the algorithms behave under the common evaluation criteria specified for CEC2005 [54]. This set contains total 25 functions comprising unimodal, multimodal, shifted, rotated and hybrid composition functions. Detailed information about these functions is available in [54].

The ABC algorithm was initialized uniformly within the search space except for 7th and 25th problems. Initial ranges for these two problems are specified in the report [54]. Other problems except 7 and 25 have the global optimum within bounds. For dimension D, values of 10, 30 and 50 were employed. The ABC algorithm was terminated when the number of function evaluations reached the MaxFES, or the error of function value was equal to 10^{-8} or less. In [54], maximum function evaluation sizes were 100,000, 300,000 and 500,000 for problem dimensions 10, 30 and 50, respectively. For fair comparison, the same evaluation numbers as in [54] were employed. The ABC algorithm was run through 10,000 cycles for all dimensions. Therefore, the colony sizes were 10, 30 and 50 for the dimensions 10, 30 and 50, respectively. For each function, the algorithm was run 25 times. Error values were sorted from the best to worst, and beside the best (1st) and the worst (25th), 7th, median (13th) and 19th function values are reported in Tables 8–10 for D = 10, Tables 11–13 for D = 30, Tables 14–16 for D = 50. In Fig. 4(a)–(e) logarithmic scaled convergence graphs of problems for D = 30 of median run (13th) are presented.

In the experiments, we tried different modification rates. For some functions (functions 7, 9, 12 and 13), better results were obtained with lower values of MR while for some functions (functions 4 and 16) using higher *MR* values produced better results. For this reason we have chosen an average value for *MR* and set it to 0.4. "*Limit*" was set to 200 for all functions

Table 21
Results of state-of-art algorithms for hybrid functions, Dimension = 10, Colony Size = 20, Cycle = 5000 for ABC, PSO-RDL: Recombination with Dynamic Linkage Discovery in PSO [17], DMS-PSO: Dynamic multi-swarm particle swarm optimizer with local search [33], SPC-PNX [4], DE: Differential Evolution [50], SADE: Self-adaptive Differential Evolution [43], restart CMA-ES: Restart Covariance Matrix Adaptation Evolution Strategy with Increasing Population Size [1].

		F21	F22	F23	F24	F25
PSO-RDL	Mean	9.94E+02	8.87E+02	1.08E+03	7.20E+02	1.76E+03
	Std.	3.27E+02	7.12E+01	2.87E+02	3.96E+02	1.54E+01
DMS-PSO	Mean	5.36E+02	6.92E+02	7.30E+02	2.24E+02	3.66E+02
	Std.	2.18E+02	1.56E+02	1.66E+02	8.31E+01	1.51E+02
SPC-PNX	Mean	6.80E+02	7.49E+02	5.76E+02	2.00E+02	4.06E+02
	Std.	2.69E+02	9.37E+01	8.22E+01	0.00E+00	2.38E−01
DE	Mean	4.92E+02	7.18E+02	5.72E+02	2.00E+02	9.23E+02
	Std.	4.00E+01	1.58E+02	4.48E+01	0.00E+00	3.40E−01
SaDE	Mean	4.64E+02	7.32E+02	6.64E+02	2.00E+02	3.76E+02
	Std.	1.58E+02	9.15E+01	1.53E+02	0.00E+00	3.15E+00
Restart CMA-ES	Mean	5.00E+02	7.29E+02	5.59E+02	2.00E+02	3.74E+02
	Std.	0.00E+00	3.18E−13	6.86E+00	3.24E−11	3.22E+00
ABC (SF = 1)	Mean	4.07E+02	8.59E+02	4.98E+02	2.02E+02	2.00E+02
	Std.	5.89E+01	7.29E+01	4.44E+01	5.76E−03	4.20E−03
ABC (SF = 0.7)	Mean	3.82E+02	7.79E+02	5.05E+02	2.02E+02	2.00E+02
	Std.	6.89E+01	1.73E+02	3.78E+01	8.26E−03	2.52E−03
ABC (SF = 0.5)	Mean	3.80E+02	8.26E+02	5.08E+02	2.02E+02	2.00E+02
	Std.	6.82E+01	1.64E+02	3.49E+01	7.62E−03	3.20E−03
ABC (SF = 0.3)	Mean	3.84E+02	9.20E+02	5.01E+02	2.02E+02	2.00E+02
	Std.	6.43E+01	1.31E+02	3.87E+01	8.32E−03	2.51E−03
ABC (MR = 1)	Mean	7.76E+02	8.38E+02	8.00E+02	2.02E+02	2.00E+02
	Std.	8.00E+01	3.40E+00	1.39E+02	4.26E−03	4.86E−04
ABC (MR = 0.8)	Mean	7.58E+02	8.40E+02	8.20E+02	2.02E+02	2.00E+02
	Std.	7.62E+01	4.41E+00	9.93E+01	5.38E−03	7.01E−04
ABC (MR = 0.6)	Mean	7.08E+02	8.41E+02	7.49E+02	2.02E+02	2.00E+02
	Std.	1.14E+02	3.85E+00	1.31E+02	5.42E−03	9.16E−04
ABC (MR = 0.4)	Mean	5.56E+02	8.43E+02	5.77E+02	2.02E+02	2.00E+02
	Std.	9.68E+01	5.97E+00	6.66E+01	6.28E−03	1.94E−03
ABC (MR = 0.2)	Mean	4.39E+02	7.89E+02	5.09E+02	2.02E+02	2.00E+02
	Std.	1.13E+02	1.32E+02	3.86E+01	5.79E−03	2.81E−03
ABC (ASF-MR: 0.9)	Mean	5.81E+02	8.20E+02	5.66E+02	2.02E+02	2.00E+02
	Std.	1.46E+02	6.11E+00	4.07E+01	8.45E−03	4.13E−03

and dimensions. Number of solutions in the population was 10, 30 and 50 for dimensions 10, 30 and 50, respectively. Values of control parameters are listed on Table 6.

In order to point out the relation between dimension and complexity for different dimensions, algorithm complexity was calculated as in basic functions. As for basic functions, code execution time (T_0), execution time of function 3 for 200,000 evaluations (T_1) and for five runs, mean of the algorithm execution times on function 3 for 200,000 evaluations ($\widehat{T}_2$ were calculated. The complexity of the algorithm was then determined by $((\widehat{T}_2 - T_1)/T_0)$ and given in Table 7. Our system was Windows XP (SP3) on Pentium (R) M 1.60 GHz processor with 1 GB of RAM and the programming language used was Delphi 7.

From the results in Tables 8–10 for D = 10, the ABC algorithm reached the given accuracy on functions 1, 2, 4, and 9. In case where dimension was 30, the ABC algorithm reached the given accuracy on functions 1, 2 and 4, while when D = 50, it reached that just on functions 1 and 2.

The ABC algorithm was not executed using an optimal control parameter set while producing the results given in Tables 8–16. We tried different parameter values for *MR*, *SF* and *limit*, and compared the performance of ABC against other algorithms that are included in the special session of CEC05 on real-parameter optimization: Recombination with Dynamic Linkage Discovery in Particle Swarm Optimization (PSO-RDL) [17], Dynamic Multi-swarm Particle Swarm Optimizer with local search (DMS-PSO) [33], SPC-PNX [4], Differential Evolution [50], Self-Adaptive Differential Evolution (SADE) [43], Restart Covariance Matrix Adaptation Evolution Strategy (Restart CMA-ES)[1]. The dimension was 10, colony size was 20 and maximum evaluation number was 100,000. For control parameter *MR*, values 1, 0.8, 0.6, 0.4, 0.2; for *SF*, values 1, 0.7, 0.5, 0.3 and adaptive scaling were employed.

Comparison results are given in the Tables 17–21. In order to demonstrate the results better, values in the tables are presented in Fig. 5(a)-Fig. 6(l). From the results in the tables and figures, all algorithms show similar performances on functions 1 and 2. On function 3, the ABC algorithm shows better performance by the increment in MR parameter and adaptive scaling. This means that the ABC algorithm needs more parameters to be mutated in the neighborhood of the current solution for function 3 because the function is non-separable. For function 4, reducing scaling factor affects the performance of the ABC algorithm negatively. Other algorithms show similar performance for function 4. For function 5, the ABC algorithm produces the best results when MR is incremented. In addition, decreasing the step size worsens the performance of the ABC algorithm. For function 6, other algorithms outperform ABC in all cases. For function 8, DMS-PSO, PSO-RDL, Restart CMA-ES and ABC algorithm with ASF demonstrate equal performances. Performance of the ABC algorithm on function 9 is affected

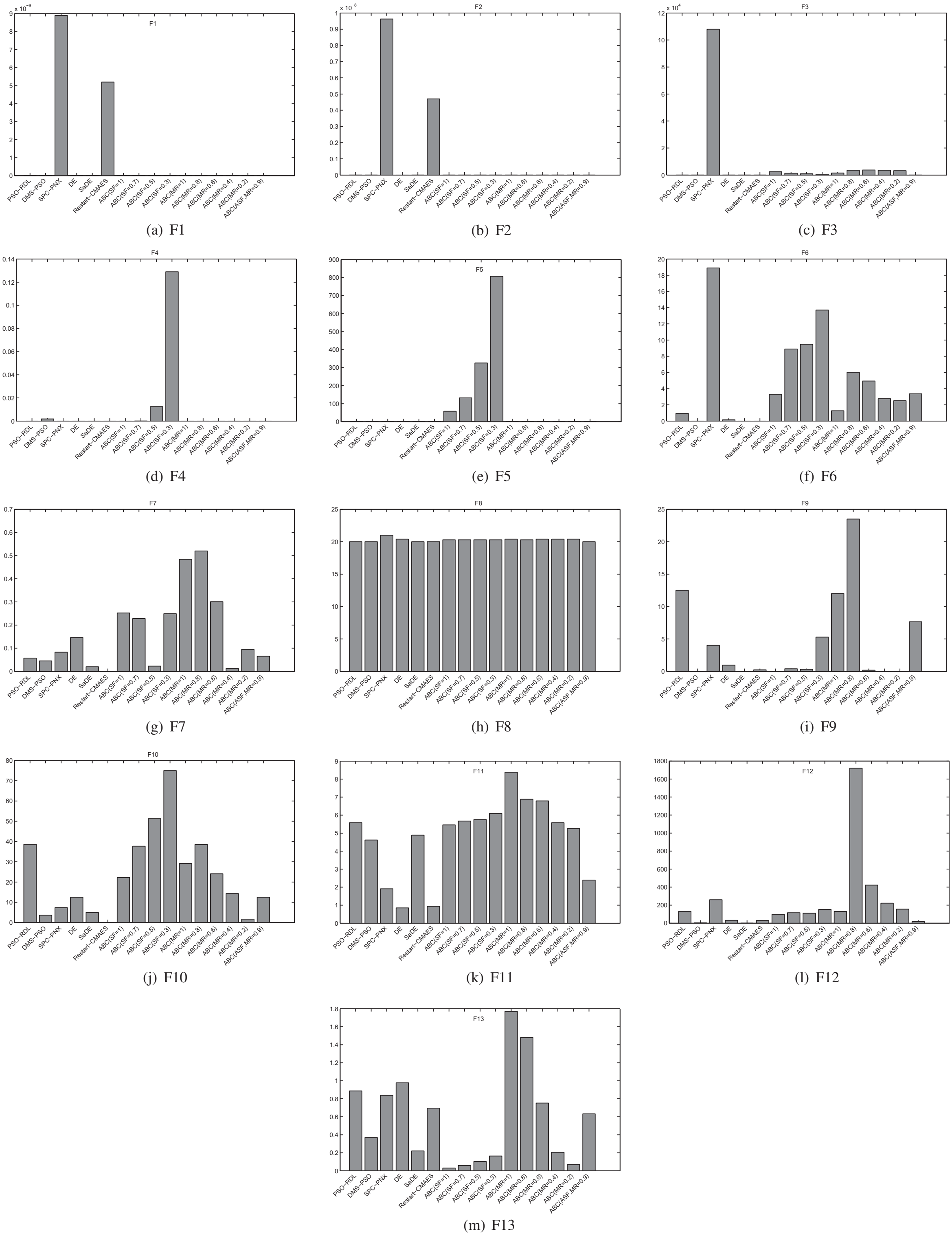


Fig. 5. Comparison of the state-of art algorithms and the variants of the ABC algorithm for composite functions F1–F13.

by the decrement on MR positively. SADE, DMS-PSO produce similar or better performances on this function compared to ABC. On function 11, restart CMAES and DE algorithms are better than others while on functions 12 and 14, SADE is better than others. On functions 13, 15, 23 and 25, better results are achieved by the basic ABC algorithm (SF = 1). On functions 7,

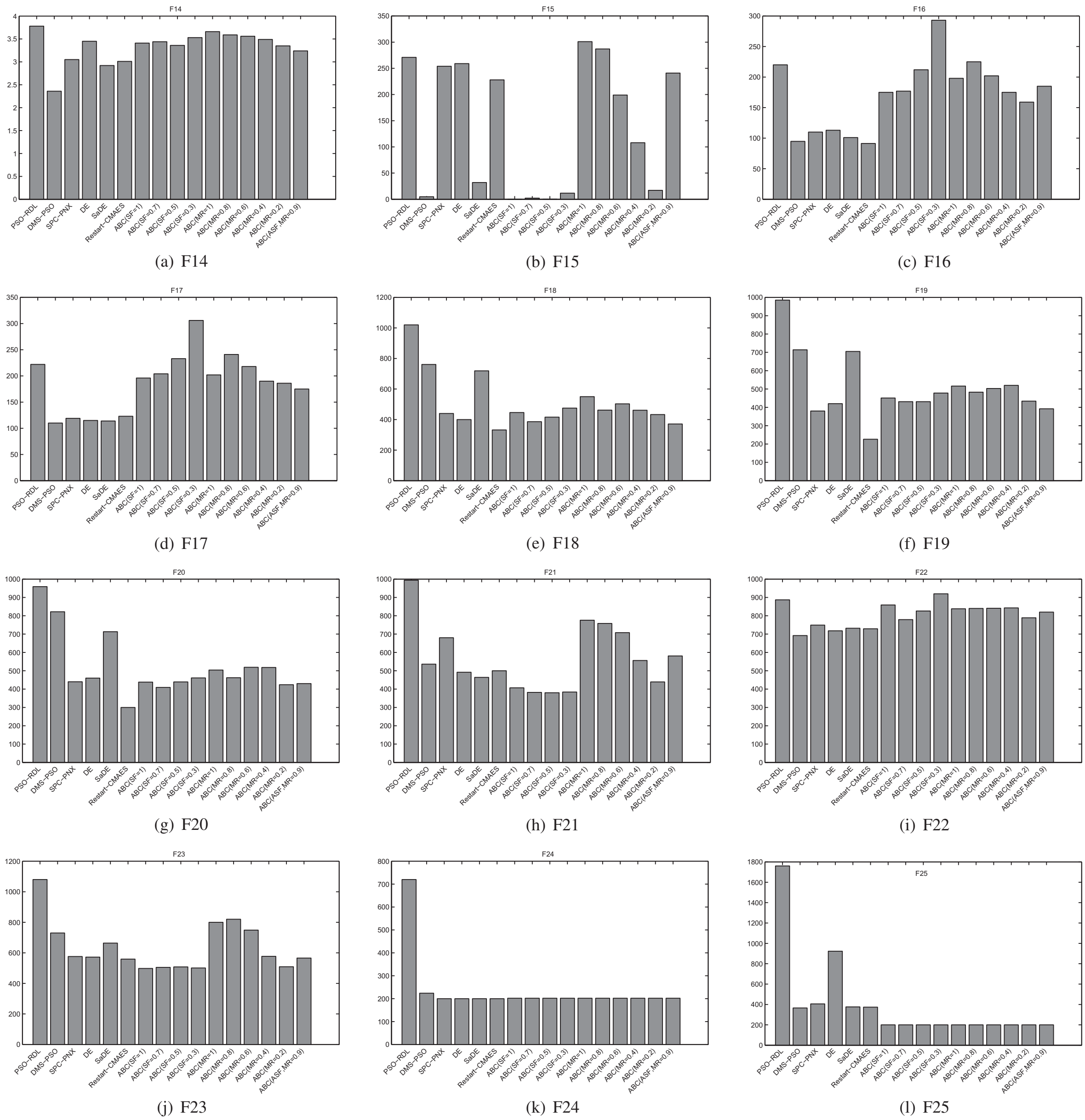


Fig. 6. Comparison of the state-of art algorithms and the variants of the ABC algorithm for composite functions F14–F25.

10, 16, 18, 19 and 20 restart CMAES is better, while on functions 17 and 22, the DMS-PSO algorithm is better. On function 24, SPC-PNX, DE, SADE and restart CMAES algorithms perform equally. On function 21, ABC produces the best performance; however, it should be noted that lower values of SF produce better results.

6. A comparative discussion on evolutionary computing paradigms vs. ABC

In the previous sections, comparative results of PSO, DE, ES and ABC variants were presented. In this section, we offer a thorough comparative analysis by considering standard versions of these algorithms.

Exploration, which is the ability to search the solution space to find promising new solutions, and exploitation, which is the ability to find the optimum solution in the neighborhood of a good solution, are two important aspects in evolutionary computing paradigms. However, different algorithms in evolutionary computing employ different operators for exploration and exploitation [48].

In ES, a mutant vector is created by adding a normally distributed random step size to each vector component [49,51]. In basic ABC, a step size which is a randomly weighted difference of the current solution and a solution randomly selected is applied to only one component of the current solution to produce a neighboring solution. Recent versions of ES such as CMA-ES use self-adaptive mechanisms for step size. Earlier versions of ES and ABC do not have a recombination operator. A fitness-based probabilistic selection scheme used in the ABC algorithm does not exist in ES. One advantage of ABC over ES is the diversification controlled by the random selection process in the scout bees phase which makes ABC escape from local minima.

In PSO, a new position vector is calculated using the particle's current and best solution and the swarm's best solution while in ABC, a new solution vector is calculated using the employed bee's current solution and a randomly chosen solution. In PSO, the new solution is replaced with the old one without considering which one is better. However, in ABC, a greedy selection scheme is applied between the new solution and the old one, and the better one is preferred for inclusion in the population. In this way, the information of a good member of the population is distributed among the other members due to the greedy selection mechanism employed. ABC also uses a probabilistic selection scheme in the onlooker bees phase in addition to this greedy selection scheme. The ABC algorithm also has a scout phase which provides diversity in the population by allowing new random solutions to be inserted into the population instead of the solutions which do not provide improvements while the PSO algorithm does not have such a process. Moreover, PSO has more control parameters than ABC.

The neighboring solution production mechanism used in ABC is similar to the self-adapting mutation process of DE. From this point of view, in DE and ABC algorithms, the solutions in the population directly affect the mutation operation since the operation is based on the difference between them. However, in DE, the difference is weighted by a constant scaling factor while in ABC, it is weighted by a random step size. Unlike DE, in ABC, there is no explicit crossover. Although both algorithms employ greedy selection between the current solution and a new solution, in DE, there is no operation as in the scout bees phase of ABC to insert a random solution into the population during a search. Therefore, although the local convergence speed of a standard DE is quite good, it might result in the premature convergence in optimizing multimodal problems if a sufficient diversity is not provided within the initial population [25].

The performance of ABC is very good in terms of the local and the global optimization due to the selection schemes employed and the neighboring production mechanism used. ABC balances exploration and exploitation efficiently.

7. Conclusion

In this work, we investigated the performance of standard and modified versions of the Artificial Bee Colony algorithm and compared their performances against state-of-the-art algorithms presented in the literature. Besides comparing the Artificial Bee Colony algorithm against some other algorithms, comparisons between its own versions were also conducted. Although the standard ABC algorithm modifies only one parameter while producing a new neighboring solution, the modified ABC algorithm employs a control parameter that determines how many parameters to be modified for the production of a neighboring solution. A scaling factor that tunes the step size adaptively was introduced. From the results, it can be concluded that the standard ABC algorithm can efficiently solve basic functions while the modified ABC algorithm produces promising results on hybrid functions compared to state-of-the-art algorithms.

References

[1] A. Auger, N. Hansen, A restart CMA evolution strategy with increasing population size, in: The 2005 IEEE Congress on Evolutionary Computation, vol. 2, September 2005, pp. 1769–1776.
[2] T. Bäck, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford University Press, Oxford, UK, 1996.
[3] A.R. Baig, M. Rashid, Foraging for fitness: A Honey Bee Behaviour Based Algorithm for Function Optimization, Technical Report, NUCES, Pakistan, November 2006.
[4] P.J. Ballester, J. Stephenson, J.N. Carter, K. Gallagher, Real-parameter optimization performance study on the cec-2005 benchmark with spc-pnx, in: The 2005 IEEE Congress on Evolutionary Computation, vol. 1, September 2005, pp. 498–505.
[5] B. Basturk, D. Karaboga, An artificial bee colony (abc) algorithm for numeric function optimization, IEEE Swarm Intelligence Symposium 2006 (Indianapolis, Indiana, USA), May 2006.
[6] A. Baykasoglu, L. Ozbakir, P. Tapkan, Swarm intelligence focus on ant and particle swarm optimization, Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem, I-Tech Education and Publishing, Vienna, Austria, 2007, pp. 113–144.
[7] E. Bendes, C. Ozkan, Direk lineer trasformasyon ynteminde yapay zeka tekniklerinin uygulanmas, in: UZALCBS08 (Kayseri, Turkiye), September 13–15, 2008.
[8] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, Inc., New York, NY, USA, 1999.
[9] C.S. Chong, A.I Sivakumar, Y.H. Malcolm Low, K.L. Gay, A bee colony optimization algorithm to job shop scheduling, in: WSC '06: Proceedings of the 38th conference on Winter simulation, Winter Simulation Conference, 2006, pp. 1954–1961.
[10] O.B. Domingo, C. Hervás-Martínez, N. García-Pedrajas, Improving crossover operator for real-coded genetic algorithms using virtual parents, Journal of Heuristics 13 (3) (2007) 265–314.
[11] M. Dorigo, V. Maniezzo, A. Colorni, Positive feedback as a search strategy, Technical Report 91-016, Politecnico di Milano, Italy, 1991.
[12] H. Drias, S. Souhila, S. Yahi, Cooperative bees swarm for solving the maximum weighted satisfiability problem, Computational Intelligence and Bioinspired Systems, LNCS, vol. 3512/2005, June 2005, pp. 318–325.
[13] L. Fenglei, D. Haijun, F. Xing, The parameter improvement of bee colony algorithm in TSP problem, Science Paper Online, November 2007.
[14] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the CEC2005 special session on real parameter optimization, Journal of Heuristics (2008).
[15] S. Ghosh, I. Marshall, Simple model of collective decision making during nectar source selection by honey bees, in: CD Rom of Workshop on Memory and Learning Mechanisms in Autonomous Robots (ECAL 2005), September 2005, p. 10.
[16] S. Hemamalini, S.P. Simon, Economic load dispatch with valve-point effect using artificial bee colony algorithm, in: XXXII National Systems Conference (India), December 17–19, 2008.

[17] M-C. Jian, Introducing Recombination with Dynamic Linkage Discovery to Particle Swarm Optimization, Technical Report NCL-TR-2006006, Natural Computing Laboratory (NCLab), Department of Computer Science, National Chiao Tung University, 2006.
[18] D. Karaboga, An Idea Based On Honey Bee Swarm for Numerical Optimization, Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
[19] D. Karaboga, B. Akay, An artificial bee colony (abc) algorithm on training artificial neural networks, in: 15th IEEE Signal Processing and Communications Applications, SIU 2007 (Eskisehir, Turkiye), June 2007, pp. 1–4.
[20] D. Karaboga, B. Akay, Effect of region scaling on the initialization of particle swarm optimization differential evolution and artificial bee colony algorithms on multimodal high dimensional problems, in: International Conference on Multivariate Statistical Modelling and High Dimensional Data Mining (Kayseri, TURKEY), June 19–23, 2008.
[21] D. Karaboga, B. Akay,, Solving large scale numerical problems using artificial bee colony algorithm, in: Sixth International Symposium on Intelligent and Manufacturing Systems Features, Strategies and Innovation (Sakarya, Turkiye), October 14–17, 2008.
[22] D. Karaboga, B. Akay, C. Ozturk, Modeling decisions for artificial intelligence, Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks, LNCS 4617/2007, Springer-Verlag, 2007, pp. 318–329.
[23] D. Karaboga, B. Basturk, Advances in soft computing: foundations of fuzzy logic and soft computing, in: Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, LNCS, 4529/2007, Springer-Verlag, 2007, pp. 789–798.
[24] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Journal of Global Optimization 39 (3) (2007) 459–471.
[25] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Applied Soft Computing 8 (1) (2008) 687–697.
[26] D. Karaboga, C. Ozturk, B. Akay, Training neural networks with ABC optimization algorithm on medical pattern classification, in: International Conference on Multivariate Statistical Modelling and High Dimensional Data Mining (Kayseri, TURKEY), June 19–23, 2008.
[27] N. Karaboga, A new design method based on artificial bee colony algorithm for digital iir filters, Journal of The Franklin Institute 346 (4) (2009) 328–348.
[28] H. Kargupta, Scalable evolutionary computation, Evolutionary Computation 7 (1) (1999) 45–68.
[29] I. Kassabalidis, M.A. El-Sharkawi, R.J. II Marks, P. Arabshahi, A.A. Gray, Swarm intelligence for routing in communication networks, Global Telecommunications Conference, GLOBECOM '01, 6, IEEE, 2001, pp. 3613–3617.
[30] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks (Piscataway, NJ), 1995, pp. 1942–1948.
[31] S.Y. Ko, I. Gupta, Y. Jo, A new class of nature-inspired algorithms for self-adaptive peer-to-peer computing, ACM Transaction on Autonomous Adaptive System 3 (3) (2008) 1–34.
[32] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Transactions on Evolutionary Computation 10 (3) (2006) 281–295.
[33] J.J. Liang and P.N. Suganthan, Dynamic multi-swarm particle swarm optimizer with local search, in: The 2005 IEEE Congress on Evolutionary Computation, vol. 1, 2005, pp. 522–528.
[34] J.J. Liang, P.N. Suganthan, K. Deb, Novel composition test functions for numerical global optimization, in: Swarm Intelligence Symposium, 2005, SIS 2005, IEEE Proceedings 2005, June 2005, pp. 68–75.
[35] X. Lu, Y. Zhou, A novel global convergence algorithm: bee collecting pollen algorithm, in: ICIC '08: Proceedings of the Fourth International Conference on Intelligent Computing, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 518–525.
[36] P. Lucic, D. Teodorovic, Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence, in: Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis (Sao Miguel, Azores Islands, Portugal), June 2001, pp. 441–445.
[37] C. Ozturk, D. Karaboga, Classification by neural networks and clustering with artificial bee colony (ABC) algorithm, in: Sixth International Symposium on Intelligent and Manufacturing Systems Features, Strategies and Innovation (Sakarya, Turkiye), October 14–17, 2008.
[38] P.J. Pawar, R.V. Rao, J.P. Davim, Optimization of process parameters of abrasive flow machining process using artificial bee colony algorithm, in: Advances in Mechanical Engineering (AME-2008) (Surat, India), December 15–17, 2008.
[39] P.J. Pawar, R.V. Rao, J.P. Davim, Optimization of process parameters of milling process using particle swarm optimization and artificial bee colony algorithm, in: Advances in Mechanical Engineering (AME-2008) (Surat, India), December 15–17, 2008.
[40] P.J. Pawar, R.V. Rao, R. Shankar, Multi-objective optimization of electro–chemical machining process parameters using artificial bee colony (ABC) algorithm, in: Advances in Mechanical Engineering (AME-2008) (Surat, India), December 15–17, 2008.
[41] K. Penev, G. Littlefair, Free search – a comparative analysis, Information Sciences 172 (2005) 173–193.
[42] D.T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi, The Bees Algorithm, Technical Report, Manufacturing Engineering Centre, Cardiff University, UK, 2005.
[43] A.K. Qin, P.N. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in: The 2005 IEEE Congress on Evolutionary Computation, vol. 2, 2005, pp. 1785–1791.
[44] F. Qingxian, D. Haijun, Bee colony algorithm for the function optimization, Science Paper Online, August 2008.
[45] H. Quan, X. Shi, On the analysis of performance of the improved artificial-bee-colony algorithm, in: Fourth IEEE International Conference on Natural Computation, ICNC (Jinan, China), August 25–27, 2008.
[46] N. Quijano, K.M. Passino, Honey bee social foraging algorithms for resource allocation, part I, in: Algorithm and theory, American Control Conference, ACC '07, 2007, pp. 3383–3388.
[47] R.S. Rao, S.V.L. Narasimham, M. Ramalingaraju, Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm, International Journal of Electrical Power and Energy Systems Engineering 1 (2) (2008) 116–122.
[48] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, Gsa: a gravitational search algorithm, Information Sciences 179 (2009) 2232–2248.
[49] I. Rechenberg, Cybernetic solution path of an experimental problem, in: Library translation 1122, Royal Aircraft Establishment, Farnborough, Hants, UK, 1965.
[50] J. Ronkkonen, S. Kukkonen, K.V. Price, Real-parameter optimization with differential evolution, in: The 2005 IEEE Congress on Evolutionary Computation, vol. 1, 2005, pp. 506–513.
[51] H.P. Schwefel, Kybernetische evolution als strategie der experimentellen forschung in der stromungstechnik, Master's thesis, Technical University of Berlin, Germany, 1965.
[52] T.D. Seeley, The Wisdom of the Hive: The Social Physiology of Honey Bee Colonies, Harvard University Press, 1995.
[53] A. Singh, An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem, Applied Soft Computing 9 (2) (2008) 625–631.
[54] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, A. ChenY.P., Auger, S. Tiwari, Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization, Technical Report, Nanyang Technological University, Singapore, 2005, <http://www.ntu.edu.sg/home/EPNSugan>.
[55] V. Tereshko, Reaction–diffusion model of a honeybee colony's foraging behaviour, in: PPSN VI: Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature, Springer-Verlag, London, UK, 2000, pp. 807–816.
[56] P-W. Tsai, J-S. Pan, B-Y. Liao, S-C. Chu, Interactive artificial bee colony (IABC) optimization, ISI2008 (Taiwan), December 12, 2008.
[57] H.R. Wedde, M. Farooq, The wisdom of the hive applied to mobile ad-hoc networks, in: Swarm Intelligence Symposium, SIS 2005, IEEE Proceedings 2005, 2005, pp. 341–348.
[58] X.S. Yang, Engineering optimizations via nature-inspired virtual bee algorithms, in: Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, LNCS, vol. 3562/2005, June 2005, pp. 317– 323.

Please cite this article in press as: B. Akay, D. Karaboga, A modified Artificial Bee Colony algorithm for real-parameter optimization, Inform. Sci. (2010), doi:10.1016/j.ins.2010.07.015