Neural Comput & Applic (2018) 29:1025–1043
https://doi.org/10.1007/s00521-016-2624-x

ORIGINAL ARTICLE

An enhanced symbiosis organisms search algorithm: an empirical study

Salah Al-Sharhan[1] · **Mahamed G. H. Omran**[1]

Received: 3 January 2015 / Accepted: 12 October 2016 / Published online: 20 October 2016
© The Natural Computing Applications Forum 2016

Abstract Many nature-inspired optimization algorithms have recently been proposed to solve difficult optimization problems where the mathematical gradient-based approaches could not be used. However, those approaches were often not tested on a proper set of problems. Moreover, statistical tests are sometimes not used to validate the conclusions. Therefore, empirical analyses of such approaches are needed. In this paper, a very recent nature-inspired approach, symbiosis organisms search (SOS), is investigated. A set of unbiased and characteristically different problems are used to study the performance of SOS. In addition, a comparison with some recent optimization methods is conducted. Then, the effect of SOS only parameter, *eco_size*, is studied, and the use of different random distributions is also explored. Finally, three simple SOS variants are proposed and compared to the original SOS. Conclusions are validated using nonparametric statistical tests.

Keywords Symbiosis organisms search · Evolutionary algorithms · Nature-inspired optimization algorithms and metaheuristics

1 Introduction

Optimization is paramount in many applications (e.g., engineering, business and industry). The aim is to minimize (or maximize) an objective function. Mathematically speaking, the continuous nonlinear function optimization problem can be expressed as follows:

$$\mathrm{Min}\, f(x) : \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}$$

The vector $\boldsymbol{x} = (x_1, \ldots, x_D)$ is composed of D real-valued components that are called *design* or *decision variables*, and the vectors $\boldsymbol{l}$ and $\boldsymbol{u}$ are assumed finite and to satisfy $<\boldsymbol{u}$, the objective function $f(\boldsymbol{x})$ is typically multimodal, so that local optima do not in general correspond to global optima. The objective function may have a nonlinear, complex or non-differentiable form. Thus, classic gradient-based optimization algorithms cannot be used in this case and non-gradient algorithms are preferred.

Nature-inspired optimization algorithms, typically non-gradient approaches, have recently become very popular. Too many algorithms have been proposed in the past 20 years. Examples are differential evolution (DE) [15], particle swarm optimization (PSO) [8], artificial bee colony (ABC) [6], flower pollination algorithm (FPA) [18], backtracking search algorithm (BSA) [2] and symbiosis organism search (SOS) [1]. For more details, the reader is referred to Simon [12] and Yang [19].

Unfortunately, many new algorithms are developed without holistic testing on a proper set of benchmark functions and without using rigorous statistical analysis. This has resulted in a "metaheuristics bubble" with too many similar and inefficient algorithms [13, 14]. In this paper, the recently proposed SOS is empirically studied, and some of its strengths and weaknesses are identified.

The objectives of this paper are:

Salah Al-Sharhan and Mahamed G. H. Omran have contributed equally to this work.

✉ Salah Al-Sharhan
alsharhans@gust.edu.kw

Mahamed G. H. Omran
omran.m@gust.edu.kw

[1] Department of Computer Science, Gulf University for Science and Technology, P.O. Box 7207, 32093 Hawally, Kuwait

1. Evaluating the performance of SOS on a wide range of problems.
2. Applying statistical tests in order to validate conclusions.
3. Comparing SOS with other similar representative methods.
4. Studying the effect of its only parameter and providing guidance on how to set this parameter.
5. Investigating the use of different random distributions in order to generate new solutions.
6. Proposing simple variants of the original SOS algorithm.

The remainder of this paper is structured as follows: Sect. 2 provides an overview of the SOS algorithm. Section 3 discusses our experimental setup. A comparison with some *state-of-the-art* methods is presented in Sect. 4. The effect of the SOS parameter is studied in Sect. 5. Different random distributions are used in Sect. 6. Section 7 presents and compares three simple SOS variants. Section 8 concludes the paper.

2 The symbiosis organisms search (SOS) algorithm

SOS simulates the interactive behavior of organisms in nature. Organisms rely on other species for survival. This reliance-based relationship is called "symbiosis." SOS maintains a population of potential solutions. The initial population is called the ecosystem. Organisms in the ecosystem is randomly generated which each organism is representing a candidate solution to the given problem. A fitness function is assigned to each organism to reflect its degree of adaptation to the desired objective. SOS consists of three phases that resemble real-world biological interactions between two organisms:

1. Mutualism phase where an interaction benefits both organisms.
2. Commensalism phase where an interaction benefits one organism, while does not harm the other.
3. Parasitism phase where one organism is benefited, while the other is harmed.

Thus, the following algorithm outlines the SOS approach:

```
Initialization
Repeat
      Mutualism phase
      Commensalism phase
      Parasitism phase
Until a stopping criterion is met
```

A detailed description of the three phases is given next.

2.1 Mutualism phase

For each organism, $\boldsymbol{x}_i$, another organism, $\boldsymbol{x}_j$, is randomly chosen from the ecosystem to interact with $\boldsymbol{x}_i$ where $\boldsymbol{x}_i \neq \boldsymbol{x}_j$. The goal of this interaction is to increase the survival chances of both organisms. New candidate solutions are generated according to Eqs. 1, 2 and 3:

$$\boldsymbol{x}_{i\text{new}} = \boldsymbol{x}_i + \epsilon_1 \circ [\boldsymbol{x}_{\text{best}} - \theta \boldsymbol{x}_m] \tag{1}$$

$$\boldsymbol{x}_{j\text{new}} = \boldsymbol{x}_j + \epsilon_2 \circ [\boldsymbol{x}_{\text{best}} - \gamma \boldsymbol{x}_m] \tag{2}$$

$$\boldsymbol{x}_m = \frac{\boldsymbol{x}_i + \boldsymbol{x}_j}{2} \tag{3}$$

where $\boldsymbol{x}_{\text{best}}$ is the best organism in the ecosystem, $\boldsymbol{x}_m$ is a mutual vector which represents the relationship characteristic between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, ϵ_1 and ϵ_2 are two random vectors, and each entry takes the values between 0 and 1, $\circ$ is the Hadamard (i.e., elementwise) product, θ and γ are benefit factors that are determined randomly as either 1 or 2. These factors represent a level of benefit to each organism of the interaction.

The $\boldsymbol{x}_{\text{best}}$ organism is used in SOS to model the highest degree of adaptation, which is the objective for both organisms.

The modified organisms, $\boldsymbol{x}_{i\text{new}}$ and $\boldsymbol{x}_{j\text{new}}$, replace the original organisms, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, if they are fitter than them. Thus, a greedy selection is used in SOS.

2.2 Commensalism phase

As in the mutualism phase, an organism, $\boldsymbol{x}_j$, is randomly selected to interact with $\boldsymbol{x}_i$. However, in this case, $\boldsymbol{x}_i$ tries to benefit of the interaction while $\boldsymbol{x}_j$ neither benefits nor suffers from the relationship. The $\boldsymbol{x}_i$ is updated using the following equation:

$$\boldsymbol{x}_{i\text{new}} = \boldsymbol{x}_i + \epsilon_3 \circ \left[\boldsymbol{x}_{\text{best}} - \boldsymbol{x}_j\right], \tag{4}$$

where ϵ_3 is a random vector and each entry takes the values between 0 and 1. The $\boldsymbol{x}_{i\text{new}}$ organism replaces $\boldsymbol{x}_i$ if it is fitter.

2.3 Parasitism phase

A parasite vector, $\boldsymbol{x}_p$, is created in the search space by duplicating organism $\boldsymbol{x}_i$ and modifying randomly chosen dimensions using a random number. Another organism, $\boldsymbol{x}_j$, is randomly selected from the ecosystem. The $\boldsymbol{x}_p$ replaces $\boldsymbol{x}_j$ if it is fitter.

From the above discussion, it is clear that each iteration of SOS requires four function evaluations. In addition, the only SOS parameter that needs to be tuned is the population size (*eco_size* in SOS terminology).

Table 1 Comparing SOS, JADE, SPSO2011 and BSA on the CEC05 functions ($D = 10$)

Functions (p value)	SOS	JADE	SPSO2011	BSA
f1 (0.0503)				
Median	7.78e−05	8.42e−05 (=)	8.32e−05 (=)	8.10e−05 (=)
Mean (SD)	7.44e−05 (1.96e−05)	8.33e−05 (1.29e−05)	8.04e−05 (1.73e−05)	7.58e−05 (1.75e−05)
FEs (SR %)	9.86e+03 (100)	1.32e+04 (100)	1.08e+04 (100)	1.57e+04 (100)
f2 (1.8012e−05)				
Median	7.91e−05	8.04e−05 (=)	9.21e−05 (+)	9.13e−05 (+)
Mean (SD)	7.66e−05 (1.16−05)	7.71−05 (1.85−05)	8.90−05 (1.08−05)	8.81−05 (1.08−05)
FEs (SR %)	1.73e+04 (100)	2.17e+04 (100)	3.11e+04 (100)	7.37e+04 (100)
f3 (1.9012−15)				
Median	7.93e+04	8.77−05 (−)	8.35e+04 (=)	1.21e+04 (−)
Mean (SD)	9.99e+04 (1.03e+05)	8.37−05 (1.35−05)	9.73e+04 (6.43e+04)	2.11e+04 (1.63e+04)
FEs (SR %)	*x* (0)	2.63e+04 (100)	*x* (0)	*x* (0)
f4 (4.3487−11)				
Median	8.92−05	8.24−05 (=)	9.36−05 (=)	5.44−04 (+)
Mean (SD)	8.28−05 (1.67−05)	7.97−05 (1.45−05)	9.14−05 (8.46−06)	1.97−03 (3.96−03)
FEs (SR %)	2.09e+04 (100)	2.32e+04 (100)	4.83e+04 (100)	9.30e+04 (23.3)
f5 (0.0176)				
Median	8.49−05	8.76−05 (=)	9.30−05 (=)	9.03−05 (=)
Mean (SD)	8.16−05 (1.64−05)	8.32−05 (1.29−05)	9.12−05 (9.65−06)	8.82−05 (8.68−06)
FEs (SR %)	1.73e+04 (100)	2.42e+04 (100)	1.95e+04 (100)	6.02e+04 (100)
f6 (2.8821−08)				
Median	1.52−01	9.40−05 (=)	6.68e+01 (+)	1.18−01 (=)
Mean (SD)	4.04e+00 (1.29e+01)	2.02e+00 (2.54e+00)	6.76e+02 (1.71e+03)	5.53−01 (9.51−01)
FEs (SR %)	*x* (0)	4.68e+04 (60)	*x* (0)	7.99e+04 (13.3)
f7 (2.9353−16)				
Median	1.27e+03	1.27e+03 (=)	1.27e+03 (+)	1.27e+03 (=)
Mean (SD)	1.27e+03 (2.60−13)	1.27e+03 (5.97−14)	1.27e+03 (2.28e+00)	1.27e+03 (8.97−02)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f8 (1.8329−07)				
Median	2.04e+01	2.04e+01 (=)	2.03e+01 (−)	2.03e+01 (=)
Mean (SD)	2.04e+01 (6.92−02)	2.04e+01 (5.85−02)	2.03e+01 (9.62−02)	2.03e+01 (6.39−02)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f9 (9.2775−17)				
Median	2.82e+00	8.65−05 (−)	7.19e+00 (=)	8.79−05 (−)
Mean (SD)	3.06e+00 (1.73e+00)	8.22−05 (1.68−05)	7.34e+00 (3.50e+00)	8.20−05 (1.52−05)
FEs (SR %)	*x* (0)	3.28e+04 (100)	*x* (0)	3.77e+04 (100)
f10 (5.6412−12)				
Median	1.52e+01	4.35e+00 (−)	7.23e+00 (−)	9.95e+00 (=)
Mean (SD)	1.55e+01 (6.61e+00)	4.22e+00 (1.20e+00)	7.53e+00 (2.32e+00)	1.04e+01 (4.66e+00)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f11 (7.6452−09)				
Median	2.49e+00	4.87e+00 (+)	6.02e+00 (+)	4.76e+00 (+)
Mean (SD)	2.62e+00 (1.40e+00)	4.84e+00 (6.20−01)	5.70e+00 (1.58e+00)	4.75e+00 (1.09e+00)
FEs (SR %)	9.40e+04 (3.33)	*x* (0)	*x* (0)	*x* (0)
f12 (1.4776−17)				
Median	3.22e+03	4.59e+02 (−)	1.07e+04 (+)	2.77e+02 (−)
Mean (SD)	3.08e+03 (1.15e+03)	4.62e+02 (2.20e+02)	1.13e+04 (5.28e+03)	2.87e+02 (1.37e+02)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)

Table 1 continued

Functions (*p* value)	SOS	JADE	SPSO2011	BSA
f13 (5.5157−15)				
Median	8.98−01	3.32−01 (−)	1.32e+00 (=)	3.18e−01 (−)
Mean (SD)	8.55−01 (3.10−01)	3.27−01 (4.76−02)	1.27e+00 (3.61e−01)	2.83e−01 (9.43e−02)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f14 (1.0727−06)				
Median	3.08e+00	2.83e+00 (=)	3.05e+00 (=)	3.27e+00 (+)
Mean (SD)	3.03e+00 (2.24e−01)	2.77e+00 (2.92−01)	3.01e+00 (3.54e−01)	3.23e+00 (2.35e−01)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
# of wins/draws/losses of SOS compared to a particular algorithm		1/8/5	5/7/2	4/6/4

+/=/− shows that SOS is significantly better/equal/significantly worse according to Friedman test at 5 % significance level. *x* (0) means that the algorithm failed to solve the problem within the admissible error

Table 2 Comparing SOS, JADE, SPSO2011 and BSA on the CEC08 functions ($D = 50$)

Functions (*p* value)	SOS	JADE	SPSO2011	BSA
F1 (0.0017)				
Median	9.63e−05	9.47e−05 (=)	9.79e−05 (=)	9.38e−05 (=)
Mean (SD)	9.14e−05 (1.38e−05)	9.33e−05 (6.13e−06)	9.70e−05 (3.31e−06)	9.17e−05 (7.04e−06)
FEs (SR %)	4.60e+04 (100)	2.73e+04 (100)	8.33e+04 (100)	6.86e+04 (100)
F2 (4.8222e−14)				
Median	2.59e+01	9.94e−05 (−)	1.64e+01 (=)	2.74e+01 (=)
Mean (SD)	2.60e+01 (3.08e+00)	9.92e−05 (7.77e−07)	1.82e+01 (7.74e+00)	2.75e+01 (4.90e+00)
FEs (SR %)	*x* (0)	1.47e+05 (100)	*x* (0)	*x* (0)
F3 (1.3133e−14)				
Median	3.85e+01	9.71e−05 (−)	1.92e+02 (+)	9.04e+01 (=)
Mean (SD)	6.54e+01 (5.72e+01)	5.32e−01 (1.38e+00)	4.92e+02 (8.46e+02)	8.43e+01 (5.80e+01)
FEs (SR %)	*x* (0)	2.16e+05 (86.7)	*x* (0)	*x* (0)
F4 (5.6622e−17)				
Median	1.04e+02	9.69e−05 (−)	2.15e+02 (=)	9.65e−05 (−)
Mean (SD)	1.06e+02 (2.95e+01)	9.56e−05 (4.84e−06)	2.14e+02 (6.36e+01)	4.31e−01 (5.65e−01)
FEs (SR %)	*x* (0)	1.70e+05 (100)	*x* (0)	2.08e+05 (60)
F5 (7.4973e−09)				
Median	3.19e−02	9.45e−05 (−)	9.96e−05 (=)	9.67e−05 (−)
Mean (SD)	4.72e−02 (6.68e−02)	1.15e−03 (4.05e−03)	4.17e−03 (6.20e−03)	3.37e−04 (1.33e−03)
FEs (SR %)	4.54e+04 (30)	2.80e+04 (93.3)	1.40e+05 (66.7)	7.08e+04 (96.7)
F6 (3.2567e−17)				
Median	2.79e+00	9.55e−05 (−)	1.76e+00 (−)	9.76e−05 (−)
Mean (SD)	2.91e+00 (4.44e−01)	9.48e−05 (4.32e−06)	1.77e+00 (5.10e−01)	9.64e−05 (3.43e−06)
FEs (SR %)	*x* (0)	3.55e+04 (100)	1.56e+05 (3.33)	8.79e+04 (100)
# of wins/draws/losses of SOS compared to a particular algorithm		0/1/5	1/4/1	0/3/3

+/=/− shows that SOS is significantly better/equal/significantly worse according to Friedman test at 5 % significance level. *x* (0) means that the algorithm failed to solve the problem within the admissible error

SOS was compared to several optimization approaches on 26 benchmark problems and 6 engineering design optimization problems from the structural engineering field [1]. However, the 26 problems are typically biased and relatively easy to solve. A biased function is a function whose minimum lies on a special line, like an axis of the system coordinates, or a symmetry axe of the search space, or a diagonal when the search space is a *D*-rectangle. Very common biased functions are those whose minimum is at the center of the search space, which is, at the same time, the origin of the system coordinates. Moreover, statistical tests were not used to validate the conclusions drawn from

Table 3 Comparing SOS, JADE, SPSO2011 and BSA on the engineering problems

Functions (*p* value)	SOS	JADE	SPSO2011	BSA
Compression Spring (0.2035)				
Median	5.56e−05	5.65e−05 (=)	5.60e−05 (=)	4.92e−05 (=)
Mean (SD)	5.50e−05 (3.02e−05)	5.32e−05 (2.97e−05)	1.53e−04 (5.15e−04)	4.77e−05 (2.79e−05)
FEs (SR %)	2.12e+04 (100)	1.27e+04 (100)	1.16e+04 (96.7)	3.58e+04 (100)
Gear Train (0.0042)				
Median	1.38e−06	1.45e−05 (+)	1.65e−05 (=)	1.36e−05 (+)
Mean (SD)	1.24e−05 (2.52e−05)	2.81e−05 (3.10e−05)	2.59e−05 (2.63e−05)	2.74e−05 (3.07e−05)
FEs (SR %)	2.10e+02 (100)	1.97e+02 (100)	8.67e+01 (100)	1.15e+02 (100)
Pressure Vessel (2.6729e−12)				
Median	5.74e−05	5.73e−05 (=)	1.27e−02 (+)	6.95e−05 (=)
Mean (SD)	5.31e−05 (3.28e−05)	4.78e−05 (3.06e−05)	2.83e+01 (8.72e+01)	6.19e−05 (2.76e−05)
FEs (SR %)	3.17e+04 (100)	1.87e+04 (100)	*x* (0)	3.92e+04 (100)
Frequency Modulation (4.0231e−15)				
Median	6.68e−05	3.25e−01 (+)	1.28e+01 (+)	2.23e−02 (+)
Mean (SD)	1.09e−02 (5.94e−02)	9.23e−01 (1.64e+00)	1.19e+01 (5.20e+00)	7.91e−01 (1.93e+00)
FEs (SR %)	3.84e+04 (96.7)	*x* (0)	*x* (0)	8.20e+04 (13.3)
Lennard-Jones (2.4090e−15)				
Median	8.95e−05	2.25e−02 (+)	4.75e−01 (+)	1.27e−03 (=)
Mean (SD)	1.45e−02 (7.53e−02)	2.49e−02 (1.16e−02)	4.55e−01 (3.40e−01)	3.08e−03 (5.13e−03)
FEs (SR %)	5.88e+04 (83.3)	*x* (0)	8.32e+04 (6.67)	8.60e+04 (3.33)
# of wins/draws/losses of SOS compared to a particular algorithm		3/2/0	3/2/0	2/3/0

+/=/− shows that SOS is significantly better/equal/significantly worse according to Friedman test at 5 % significance level. *x* (0) means that the algorithm failed to solve the problem within the admissible error

the experiments. Therefore, those conclusions are questionable.

In the next section, the experimental setup is described where a set of 25 problems are used to test the performance of SOS. This set consists of different types of problems with different characteristics that are typically not biased. In addition, the statistical test used to validate the results of the experiments is briefly stated.

3 Experimental setup

To test the performance of the competing methods, we have chosen 25 functions: 14 CEC05 functions (namely *f1*–*f14*), five real-world engineering problems (namely Gear Train, Compression Spring, Pressure Vessel, Lennard-Jones and Frequency Modulation Sound Parameter Identification) and the 6 CEC08 large-scale global optimization problems (i.e., *F1*–*F6*). The 14 CEC05 functions have different characteristics (e.g., rotated, non-separable, shifted, unimodal, multimodal, composite). For more details about the CEC05 functions, the reader is referred to Suganthan et al. [16]. More information on the first three real-world problems can be found in Sandgren [11]. For the Lennard-Jones and Frequency Identification problems, see Das and Suganthan [3]. For more details about the CEC08, refer to Tang et al. [17].

For all functions (except the CEC08 functions), we have used 100,000 function evaluations (FEs). For the CEC08 functions, 5000D FEs were used as suggested by Tang et al. [17]. The results of all methods have been obtained using 30 independent runs of the algorithms against each problem.

Given the best-of-run error, which is defined as the absolute difference between the best-of-the-run $f(\boldsymbol{x}^*)$ value and the actual optimum $f(\boldsymbol{x}')$ of a given function, i.e.,

$$\text{err.} = |f(\boldsymbol{x}^*) - f(\boldsymbol{x}')|$$

The *effectiveness* of a method is measured using three metrics:

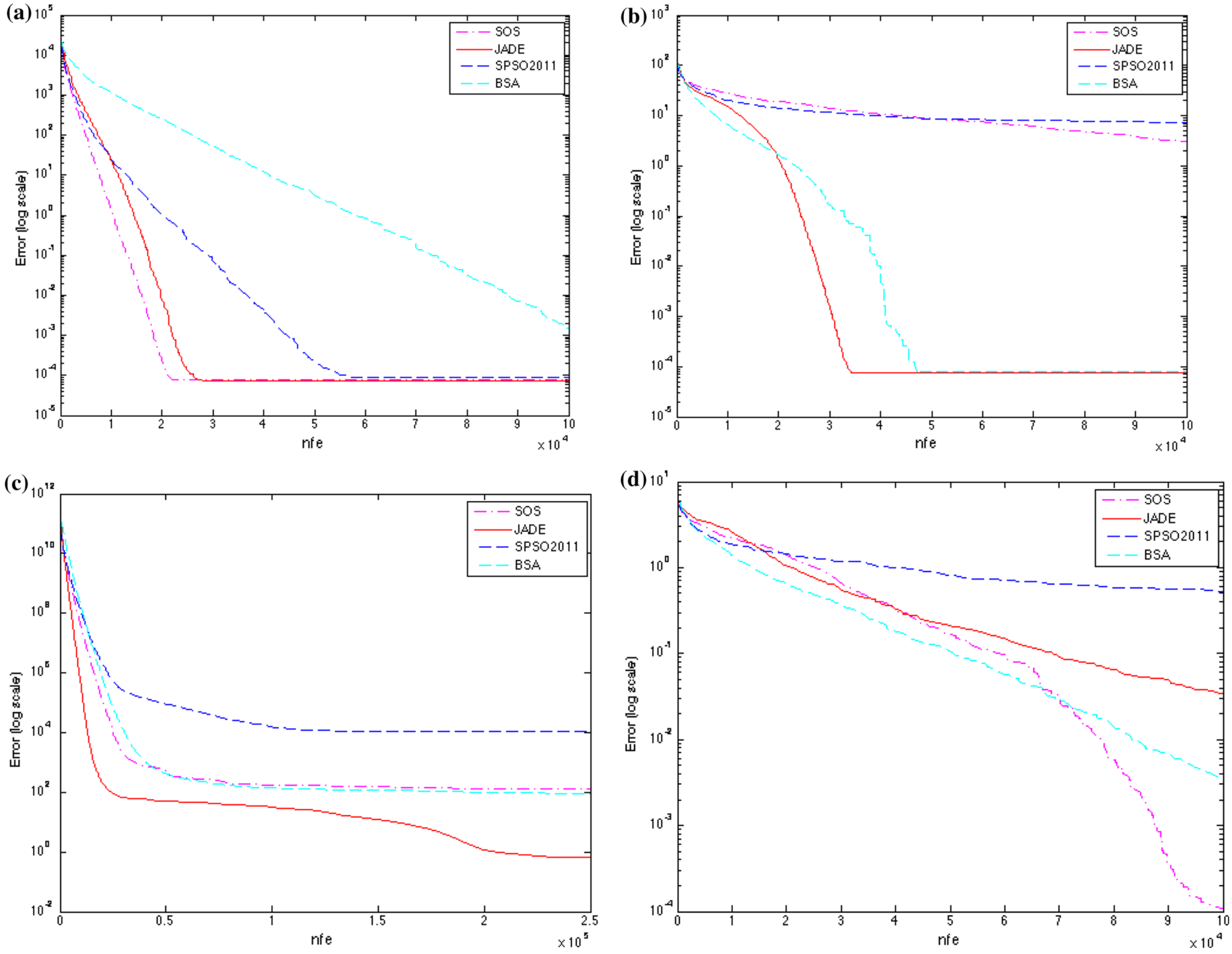

Fig. 1 Mean best error curves of *SOS, JADE, SPSO2011* and *BSA* for selected problems. **a** CEC05 f4, **b** CEC05 f9, **c** CEC08 F3, **d** Lennard-Jones

1. The median of the best-of-run error.
2. The mean and the standard deviation of the best-of-run error.
3. Success rate (SR): The number of successful runs, where a run is successful if err. $\leq$ admissible error. In this paper, the admissible error is set to 1e−4.

In solving real-world problems, the function evaluation time overwhelms the algorithm overhead. Hence, the mean number of FEs needed to reach acceptable accuracy will be much more interesting than the CPU time [21]. Hence, the *efficiency* of a method is measured in terms of the number of FEs.

All programs are implemented using MATLAB version 8.1.0.604 (R2013a), and machine epsilon is 2.2204e−16. For the pseudo-random number generator (RNG), we have used the `rand` built-in function provided by MATLAB. This function implements the Mersenne Twister RNG [9]. We warmed the RNG by calling it 10,000 at the start of the program as suggested by Jones [5]. The nonparametric Friedman's test with $\alpha = 0.05$ and the Dunn–Sidak correction as a post hoc test have been used to compare the difference in performance of the different algorithms. The post hoc statistical analysis could help up to detect concrete differences between the competing algorithms. In this study, the null hypothesis, H_0, states that there is no difference between the *medians* of errors of the different algorithms. Nonparametric statistics do not need prior assumptions related to the sample of data being analyzed [4].

Table 4 Effect of *eco_size* on the performance of SOS on the CEC05 functions ($D = 10$)

Functions (*p* value)	25	50	75	100
f1 (0.4170)				
Median	8.01e−05	8.13e−05	7.07e−05	7.93e−05
Mean (SD)	7.55e−05 (1.77e−05)	7.88e−05 (1.66e−05)	7.05e−05 (1.54e−05)	7.51e−05 (1.77e−05)
FEs (SR %)	5.64e+03 (100)	9.80e+03 (100)	1.41e+04 (100)	1.85e+04 (100)
f2 (1.2268e−04)				
Median	*9.16e−05*	*8.85e−05*	**7.30e−05**	**7.22e−05**
Mean (SD)	8.74e−05 (1.20e−05)	8.24e−05 (1.61e−05)	7.16e−05 (1.57e−05)	7.35e−05 (1.67e−05)
FEs (SR %)	1.18e+04 (100)	1.72e+04 (100)	2.47e+04 (100)	3.18e+04 (100)
f3 (0.0131)				
Median	4.93e+04	5.66e+04	**5.27e+04**	*7.72e+04*
Mean (SD)	6.38e+04 (4.75e+04)	7.51e+04 (6.85e+04)	6.13e+04 (3.75e+04)	9.29e+04 (6.32e+04)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f4 (0.3561)				
Median	8.72e−05	8.20e−05	8.80e−05	8.15e−05
Mean (SD)	8.46e−05 (1.33e−05)	8.05e−05 (1.17e−05)	8.38e−05 (1.37e−05)	7.58e−05 (1.89e−05)
FEs (SR %)	2.01e+04 (100)	2.07e+04 (100)	2.82e+04 (100)	3.67e+04 (100)
f5 (0.7819)				
Median	7.90e−05	8.01e−05	8.59e−05	8.20e−05
Mean (SD)	7.68e−05 (1.70e−05)	7.49e−05 (1.92e−05)	8.00e−05 (1.95e−05)	8.07e−05 (1.39e−05)
FEs (SR %)	1.10e+04 (100)	1.76e+04 (100)	2.47e+04 (100)	3.18e+04 (100)
f6 (0.8590)				
Median	3.86e−01	3.68e−01	3.04e−01	5.24e−01
Mean (SD)	9.43e−01 (1.46e+00)	5.25e+00 (1.73e+01)	6.42e+00 (1.60e+01)	5.87e+00 (2.01e+01)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f7 (0.2615)				
Median	1.27e+03	1.27e+03	1.27e+03	1.27e+03
Mean (SD)	1.27e+03 (8.44e−14)	1.27e+03 (5.97e−14)	1.27e+03 (5.97e−14)	1.27e+03 (0.00e+00)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f8 (0.3504)				
Median	2.04e+01	2.03e+01	2.03e+01	2.04e+01
Mean (SD)	2.04e+01 (6.16e−02)	2.03e+01 (8.48e−02)	2.03e+01 (6.74e−02)	2.04e+01 (7.21e−02)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f9 (1.1440e−14)				
Median	**8.77e−05**	*3.34e+00*	*5.40e+00*	*9.60e+00*
Mean (SD)	5.48e−01 (1.11e+00)	3.79e+00 (2.75e+00)	5.97e+00 (2.79e+00)	9.37e+00 (2.62e+00)
FEs (SR %)	7.94e+04 (70)	*x* (0)	*x* (0)	*x* (0)
f10 (6.4789e−07)				
Median	**1.04e+01**	*1.70e+01*	*1.99e+01*	*2.35e+01*
Mean (SD)	1.17e+01 (6.21e+00)	1.65e+01 (5.88e+00)	1.94e+01 (5.25e+00)	2.23e+01 (5.22e+00)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f11 (0.0272)				
Median	*3.32e+00*	**2.20e+00**	2.41e+00	3.51e+00
Mean (SD)	3.44e+00 (1.25e+00)	2.14e+00 (1.47e+00)	2.92e+00 (1.93e+00)	3.48e+00 (2.30e+00)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f12 (7.4220e−06)				
Median	**1.74e+03**	2.92e+03	*3.88e+03*	*4.11e+03*
Mean (SD)	2.27e+03 (1.55e+03)	3.10e+03 (1.34e+03)	4.03e+03 (1.10e+03)	4.09e+03 (1.56e+03)

Table 4 continued

Functions (*p* value)	25	50	75	100
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f13 (2.1544e−06)				
Median	**6.06e−01**	9.03e−01	*1.07e+00*	*1.25e+00*
Mean (SD)	6.05e−01 (3.04e−01)	8.63e−01 (3.85e−01)	1.02e+00 (3.71e−01)	1.25e+00 (3.39e−01)
FEs (SR %)	*x* (0)	5.30e+04 (3.33)	8.95e+04 (3.33)	*x* (0)
f14 (0.0011)				
Median	**3.01e+00**	3.08e+00	*3.21e+00*	*3.30e+00*
Mean (SD)	2.95e+00 (2.80e−01)	3.05e+00 (2.90e−01)	3.12e+00 (3.16e−01)	3.26e+00 (2.25e−01)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
# of times the algorithm is best	5	1	2	1

Bold is statistically better than italic at 5 % significance level

Table 5 Effect of *eco_size* on the performance of SOS on the CEC08 functions ($D = 50$)

Functions (*p* value)	25	50	75	100
F1 (0.1351)				
Median	7.94e−05	9.23e−05	9.34e−05	9.24e−05
Mean (SD)	6.61e−05 (3.35e−05)	7.99e−05 (2.79e−05)	9.05e−05 (1.24e−05)	9.07e−05 (8.34e−06)
FEs (SR %)	5.49e+04 (100)	4.80e+04 (100)	5.15e+04 (100)	5.73e+04 (100)
F2 (3.3030e−15)				
Median	*3.19e+01*	*2.58e+01*	**2.23e+01**	**1.83e+01**
Mean (SD)	3.21e+01 (2.22e+00)	2.60e+01 (2.74e+00)	2.24e+01 (3.31e+00)	1.81e+01 (4.20e+00)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
F3 (0.7055)				
Median	9.33e+01	8.36e+01	8.65e+01	9.21e+01
Mean (SD)	1.14e+02 (7.41e+01)	9.68e+01 (6.73e+01)	1.28e+02 (1.88e+02)	9.80e+01 (7.77e+01)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
F4 (1.1475e−09)				
Median	**7.98e+01**	**1.07e+02**	*1.40e+02*	*1.62e+02*
Mean (SD)	8.07e+01 (1.91e+01)	1.01e+02 (2.72e+01)	1.41e+02 (3.71e+01)	1.59e+02 (3.91e+01)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
F5 (1.7635e−04)				
Median	*1.05e−01*	4.18e−02	1.23e−02	**3.75e−03**
Mean (SD)	1.57e−01 (1.67e−01)	5.85e−02 (7.35e−02)	6.20e−02 (1.01e−01)	2.46e−02 (5.61e−02)
FEs (SR %)	*x* (0)	4.69e+04 (16.7)	5.17e+04 (30)	6.74e+04 (50)
F6 (1.0727e−06)				
Median	*3.17e+00*	2.70e+00	2.53e+00	**2.37e+00**
Mean (SD)	3.24e+00 (6.15e−01)	2.75e+00 (4.35e−01)	2.58e+00 (3.21e−01)	2.45e+00 (4.32e−01)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
# of times the algorithm is best	1	1	1	3

Bold is statistically better than italic at 5 % significance level

4 SOS versus state-of-the-art methods

In this section, we have compared SOS to three representative population-based optimization methods, namely DE, PSO and BSA. For DE, a self-adaptive DE, called JADE, proposed by Zhang and Sanderson [20] is used. JADE can be considered as a good performing DE variant and it is one of the best evolutionary algorithms for global optimization [22] and this is why we have chosen it. For PSO, the most recent standard SPSO2011 is used. A MATLAB implementation of SPSO2011 can be downloaded from http://www.particleswarm.info. BSA, on the other hand, is

Table 6 Effect of *eco_size* on the performance of SOS on the engineering problems

Functions (*p* value)	25	50	75	100
Compression Spring (0.6504)				
Median	4.68e−05	4.39e−05	5.13e−05	5.45e−05
Mean (SD)	3.35e−04 (1.57e−03)	4.66e−05 (2.70e−05)	5.35e−05 (3.04e−05)	5.19e−05 (2.98e−05)
FEs (SR %)	1.65e+04 (96.7)	1.98e+04 (100)	2.16e+04 (100)	2.48e+04 (100)
Gear Train (0.7494)				
Median	1.70e−06	2.22e−06	5.47e−06	3.82e−06
Mean (SD)	1.29e−05 (1.93e−05)	1.02e−05 (1.63e−05)	1.66e−05 (2.12e−05)	1.39e−05 (1.97e−05)
FEs (SR %)	1.22e+02 (100)	1.90e+02 (100)	2.35e+02 (100)	2.87e+02 (100)
Pressure Vessel (0.9893)				
Median	6.21e−05	7.15e−05	5.50e−05	6.61e−05
Mean (SD)	5.50e−05 (3.11e−05)	6.10e−05 (3.04e−05)	5.59e−05 (3.27e−05)	5.84e−05 (3.06e−05)
FEs (SR %)	1.63e+04 (100)	3.03e+04 (100)	4.41e+04 (100)	5.93e+04 (100)
Frequency Modulation (0.1005)				
Median	8.93e−05	8.07e−05	8.13e−05	9.18e−05
Mean (SD)	2.26e+00 (4.30e+00)	1.50e−01 (8.20e−01)	9.12e−01 (2.37e+00)	5.21e−01 (1.36e+00)
FEs (SR %)	3.24e+04 (73.3)	4.20e+04 (93.3)	6.55e+04 (70)	7.08e+04 (63.3)
Lennard-Jones (9.5493e−07)				
Median	**8.41e−05**	**8.16e−05**	1.88e−04	*4.10e−02*
Mean (SD)	8.26e−05 (1.31e−05)	9.44e−05 (8.38e−05)	9.36e−02 (3.25e−01)	2.41e−01 (3.58e−01)
FEs (SR %)	2.64e+04 (100)	5.66e+04 (96.7)	8.36e+04 (46.7)	8.58e+04 (20)
# of times the algorithm is best	1	1	0	0

Bold is statistically better than italic at 5 % significance level

a very recent and simple metaheuristic that has shown to perform well on a wide range of problems [2]. The three optimization methods have been used to check if SOS can compete with some similar and recent approaches.

Table 1 shows that JADE clearly outperforms SOS on the 14 CEC05 benchmark functions. However, compared to SPSO2011, SOS manages to perform better on 5 functions while performing worse on two. SOS and BSA perform comparably on this set of benchmark problems. Comparing the efficiency of the four approaches, SOS seems to require less number of function evaluations to reach an “optimal” solution.

On high-dimensional problems, Table 2 shows that JADE outperforms SOS on almost all functions. SOS performs comparably to SPSO2011, while BSA generally performs better than SOS especially on multimodal functions. In addition, Table 2 shows that JADE is the most efficient approach when applied to the CEC08 problems.

Table 3 compares the performance of the four methods on real-world engineering problems. Based on the reported results, SOS is the best approach for these problems. The other three methods could not beat SOS on any single problem.

To summarize the results of Tables 1, 2 and 3, we can say that JADE is the best approach on the benchmark functions especially on high-dimensional problems. SOS, on the other hand, works particularly well on real-world engineering problems. In addition, SOS generally performs better than SPSO2011 on the standard benchmark functions. BSA performs better than SOS on high-dimensional problems.

The results reported in this section suggest that SOS needs further improvement in order to compete well with other state-of-the-art approaches.

Figure 1 shows the progress of the mean best errors found by the four approaches over 30 runs for selected problems.

5 Effect of *eco_size*

In this section, the effect of the population size, *eco_size*, on the performance of SOS is investigated.

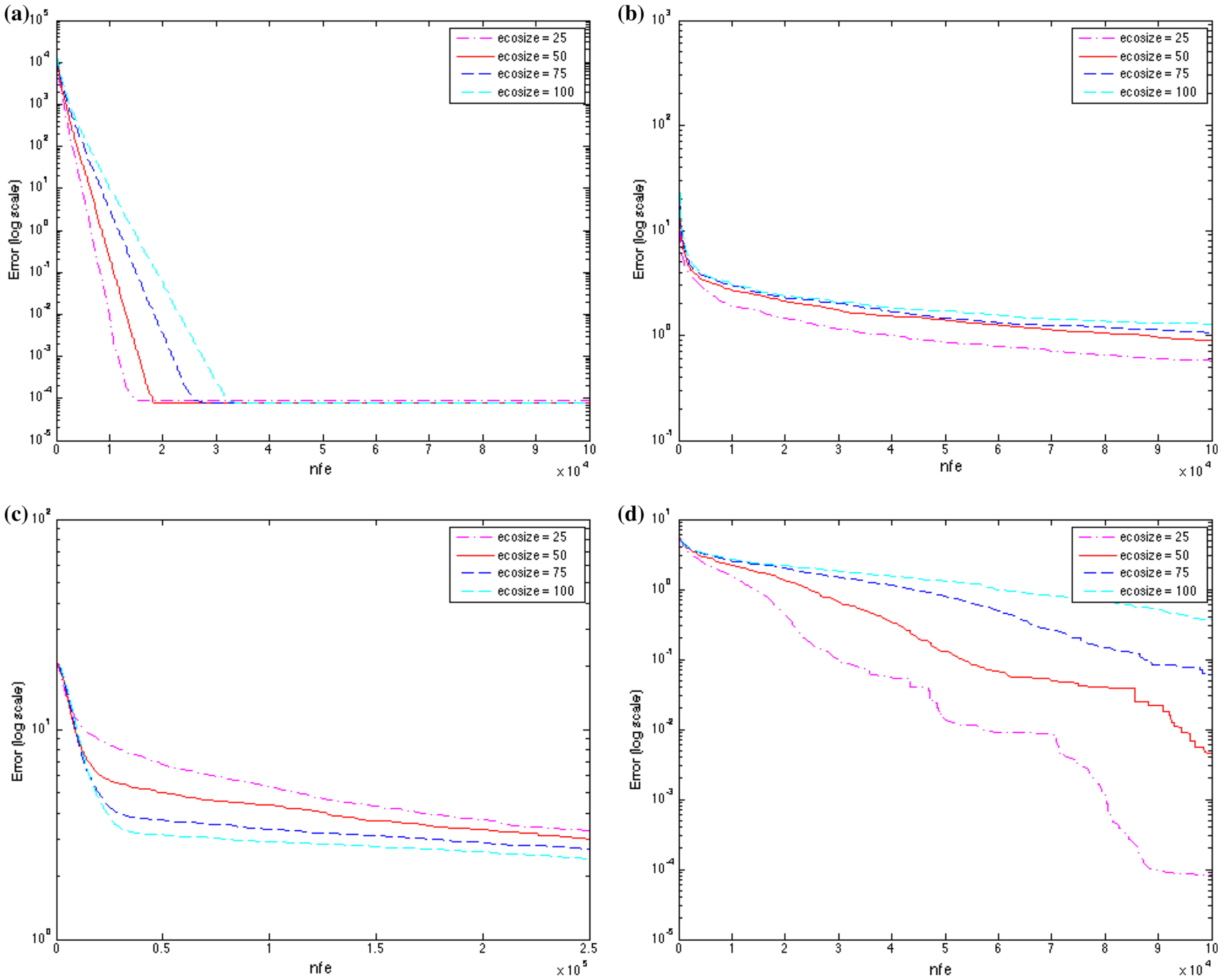

Fig. 2 Mean best error curves of different *eco_size* values for selected problems. **a** CEC05 f2, **b** CEC05 f13, **c** CEC08 F6, **d** Lennard-Jones

Tables 4, 5 and 6 present the effect of *eco_size* on the performance of SOS when applied to CEC05 functions, CEC08 functions and the engineering problems, respectively. Table 4 shows that using a relatively small *eco_size* (i.e., 25) generally improves the performance of SOS on multimodal functions.

On the other hand, on high-dimensional CEC08 problems, Table 5 suggests that using a relatively big value of *eco_size* (i.e., 100) yields better solutions.

For the engineering problems, results reported in Table 6 show that generally no significant difference is noted between the different values of *eco_size* except for the Lennard-Jones problem, where a relatively small value of *eco_size* ($\leq$50) significantly generates better solutions.

Therefore, based on the results reported in this section, a relatively small value of *eco_size* should be used for low-dimensional problems, while a relatively big value should be used for high-dimensional problems (i.e., *eco_size* is proportionally to *D*).

Figure 2 shows the convergence curves of SOS using different values of *eco_size* over 30 runs for selected problems. The figure shows that using 25 organisms enables SOS to converge to a better solution faster than bigger values of *eco_size* on low-dimensional problems (Fig. 2a, b, d). However, Fig. 2c shows that for a high-dimensional function, the performance improves as *eco_size* increases.

6 Commensalism phase: effect of using different distributions

Randomization plays an important role in both exploration and exploitation of a metaheuristic algorithm. Simple random numbers are often drawn from either a uniform

Table 7 Effect of using different distributions in the commensalism phase of SOS when applied to the CEC05 functions ($D = 10$)

Functions (*p* value)	Uniform	Normal	Lévy
f1 (0.9048)			
Median	7.80e−05	7.52e−05	7.35e−05
Mean (SD)	7.36e−05 (1.71e−05)	7.32e−05 (1.65e−05)	7.45e−05 (1.49e−05
FEs (SR %)	9.87e+03 (100)	1.03e+04 (100)	1.08e+04 (100)
f2 (0.7408)			
Median	8.32e−05	8.71e−05	8.57e−05
Mean (SD)	8.11e−05 (1.33e−05)	8.38e−05 (1.38e−05)	8.27e−05 (1.33e−05)
FEs (SR %)	1.76e+04 (100)	1.81e+04 (100)	1.88e+04 (100)
f3 (0.4346)			
Median	5.66e+04	5.70e+04	7.63e+04
Mean (SD)	8.59e+04 (8.37e+04)	6.32e+04 (6.24e+04)	9.29e+04 (6.18e+04)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)
f4 (0.9048)			
Median	7.99e−05	8.49e−05	8.55e−05
Mean (SD)	8.00e−05 (1.58e−05)	8.14e−05 (1.47e−05)	7.94e−05 (1.58e−05)
FEs (SR %)	2.03e+04 (100)	2.13e+04 (100)	2.23e+04 (100)
f5 (7.2211e−04)			
Median	**8.04e−05**	*8.70e−05*	*9.49e−05*
Mean (SD)	7.70e−05 (1.70e−05)	8.50e−05 (1.02e−05)	2.13e−03 (9.58e−03)
FEs (SR %)	1.77e+04 (100)	1.78e+04 (100)	3.30e+04 (86.7)
f6 (0.3558)			
Median	1.55e−01	1.23e−01	2.72e+00
Mean (SD)	4.62e+00 (1.37e+01)	6.60e+00 (2.17e+01)	2.78e+00 (2.57e+00)
FEs (SR %)	6.56e+04 (3.33)	*x* (0)	3.78e+04 (3.33)
f7 (3.8430e−05)			
Median	**1.27e+03**	**1.27e+03**	*1.27e+03*
Mean (SD)	1.27e+03 (4.22e−14)	1.27e+03 (5.97e−14)	1.27e+03 (6.22e−02)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)
f8 (0.7919)			
Median	2.04e+01	2.04e+01	2.04e+01
Mean (SD)	2.04e+01 (6.54e−02)	2.04e+01 (5.48e−02)	2.04e+01 (7.32e−02)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)
f9 (0.0145)			
Median	*2.91e+00*	*3.23e+00*	**2.00e+00**
Mean (SD)	3.28e+00 (2.43e+00)	3.35e+00 (2.58e+00)	1.77e+00 (1.09e+00)
FEs (SR %)	*x* (0)	9.40e+04 (3.33)	*x* (0)
f10 (0.0017)			
Median	1.24e+01	1.43e+01	**8.08e+00**
Mean (SD)	1.28e+01 (5.30e+00)	1.61e+01 (9.19e+00)	8.57e+00 (3.20e+00)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)
f11 (0.7919)			
Median	2.37e+00	2.53e+00	2.54e+00
Mean (SD)	2.57e+00 (1.61e+00)	2.54e+00 (1.27e+00)	2.85e+00 (1.45e+00)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)
f12 (6.1283e−05)			
Median	*2.82e+03*	*3.19e+03*	**1.57e+03**
Mean (SD)	2.85e+03 (1.38e+03)	3.45e+03 (1.49e+03)	1.56e+03 (6.58e+02)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)

Table 7 continued

Functions (*p* value)	Uniform	Normal	Lévy
f13 (1.9884e−08)			
Median	*1.04e+00*	*8.68e−01*	**4.48e−01**
Mean (SD)	9.55e−01 (3.12e−01)	8.86e−01 (3.46e−01)	4.07e−01 (1.80e−01)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)
f14 (0.0029)			
Median	*3.03e+00*	*3.14e+00*	**2.91e+00**
Mean (SD)	3.04e+00 (2.46e−01)	3.16e+00 (3.00e−01)	2.87e+00 (2.69e−01)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)
# of times the algorithm is best	2	1	5

Bold is statistically better than italic at 5 % significance level

Table 8 Effect of using different distributions in the commensalism phase of SOS when applied to the CEC08 functions ($D = 50$)

Functions (*p* value)	Uniform	Normal	Lévy
F1 (1.6364e−10)			
Median	**9.54e−05**	**9.42e−05**	*8.46e+00*
Mean (SD)	9.28e−05 (9.42e−06)	9.05e−05 (1.12e−05)	1.65e+01 (2.20e+01)
FEs (SR %)	4.61e+04 (100)	4.85e+04 (100)	*x* (0)
F2 (1.3363e−09			
Median	**2.61e+01**	**2.45e+01**	*3.50e+01*
Mean (SD)	2.63e+01 (3.33e+00)	2.43e+01 (2.88e+00)	3.50e+01 (3.33e+00)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)
F3 (9.9255e−11)			
Median	**8.19e+01**	**8.31e+01**	*4.70e+04*
Mean (SD)	9.97e+01 (6.35e+01)	9.50e+01 (6.10e+01)	1.17e+05 (1.73e+05)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)
F4 (0.7408)			
Median	1.05e+02	1.01e+02	1.04e+02
Mean (SD)	1.03e+02 (2.36e+01)	9.84e+01 (2.71e+01)	1.01e+02 (1.77e+01)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)
F5 (1.6919e−10)			
Median	**2.95e−02**	**7.61e−02**	*1.42e+00*
Mean (SD)	5.21e−02 (6.55e−02)	7.76e−02 (8.18e−02)	1.53e+00 (4.83e−01)
FEs (SR %)	4.78e+04 (20)	4.74e+04 (26.7)	*x* (0)
F6 (9.8743e−09)			
Median	**3.00e+00**	**2.81e+00**	*4.23e+00*
Mean (SD)	3.04e+00 (5.50e−01)	2.84e+00 (4.63e−01)	4.20e+00 (5.65e−01)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)
# of times the algorithm is best	5	5	0

Bold is statistically better than italic at 5 % significance level

distribution or a Gaussian normal distribution. Equation (4) in the commensalism phase uses a uniform distribution to generate the vector, ϵ_3, of random numbers between 0 and 1. In this section, we investigate the use of distributions with "infinite" support by the search space like normal and Lévy distributions to generate ϵ_3. If ϵ_3 is drawn from a

Table 9 Effect of using different distributions in the commensalism phase of SOS when applied to the engineering problems

Functions (*p* value)	Uniform	Normal	Lévy
Compression Spring (0.1072)			
Median	6.13e−05	5.56e−05	4.93e−05
Mean (SD)	6.16e−05 (2.64e−05)	5.16e−05 (2.95e−05)	1.66e−04 (6.64e−04)
FEs (SR %)	2.00e+04 (100)	2.06e+04 (100)	1.33e+04 (96.7)
Gear Train (0.3471)			
Median	5.80e−06	2.40e−06	3.66e−06
Mean (SD)	1.94e−05 (2.69e−05)	1.67e−05 (2.72e−05)	1.79e−05 (2.68e−05)
FEs (SR %)	1.97e+02 (100)	1.97e+02 (100)	1.97e+02 (100)
Pressure Vessel (0.6483)			
Median	4.99e−05	6.43e−05	5.80e−05
Mean (SD)	5.24e−05 (2.82e−05)	6.19e−05 (2.58e−05)	6.16e−05 (2.31e−05)
FEs (SR %)	3.02e+04 (100)	3.09e+04 (100)	3.35e+04 (100)
Frequency Modulation (0.0036)			
Median	**8.53e−05**	**8.40e−05**	*9.78e−05*
Mean (SD)	4.77e−01 (1.29e+00)	1.02e+00 (2.46e+00)	2.49e+00 (3.76e+00)
FEs (SR %)	4.50e+04 (80)	4.74e+04 (80)	5.50e+04 (60)
Lennard-Jones (0.2725)			
Median	8.88e−05	8.68e−05	9.24e−05
Mean (SD)	3.54e−03 (1.89e−02)	5.31e−04 (2.45e−03)	2.26e−04 (3.92e−04)
FEs (SR %)	5.69e+04 (93.3)	5.29e+04 (93.3)	7.16e+04 (83.3)
# of times the algorithm is best	1	1	0

Bold is statistically better than italic at 5 % significance level

normal distribution (i.e., $r_{3,d} \sim N(0,1)$ where $\epsilon_{3,d}$ is the dth component of ϵ_3), the random walk (which is a random process consists of taking a series of consecutive random steps) is called a Brownian random walk. However, if Lévy distribution is used, the random walk is called a Lévy flight. Lévy flights are more efficient than Brownian random walks in exploring unknown, large-scale search spaces [19].

Table 7 compares the three distributions on the CEC05 functions. The results show that on unimodal functions, there is generally no significant difference between the three distributions. On multimodal functions, however, Lévy distribution outperforms uniform and normal distributions given that Lévy flights are efficient in exploring search spaces and thus more suitable for multimodal functions.

However, Table 8 shows that on high-dimensional problems, uniform and normal distributions outperform Lévy distribution. This suggests that Lévy distributions require more time (i.e., more function evaluations) to reach good solutions when applied to high-dimensional problems.

On engineering problems, Table 9 generally shows that no significant difference is observed between the three distributions.

From Tables 7, 8 and 9, we can conclude that using a distribution with infinite support does not mean that an optimization algorithm become better. It is true on some problems (like CEC05 multimodal functions when $D = 10$), but it is false on other problems (e.g., CEC08 using $D = 50$) because such a distribution may need far more iterations to find an “optimal” solution.

Similar analysis could be carried in the mutualism phase where ϵ_1 and ϵ_2 in Eqs. (1) and (2) can be drawn from normal and Lévy distributions.

Figure 3 shows the convergence curves of SOS using the three random distributions for some representative problems.

7 Enhanced SOS algorithm

In this section, three new variants are proposed and compared with the original SOS. The three SOS variants are explained next.

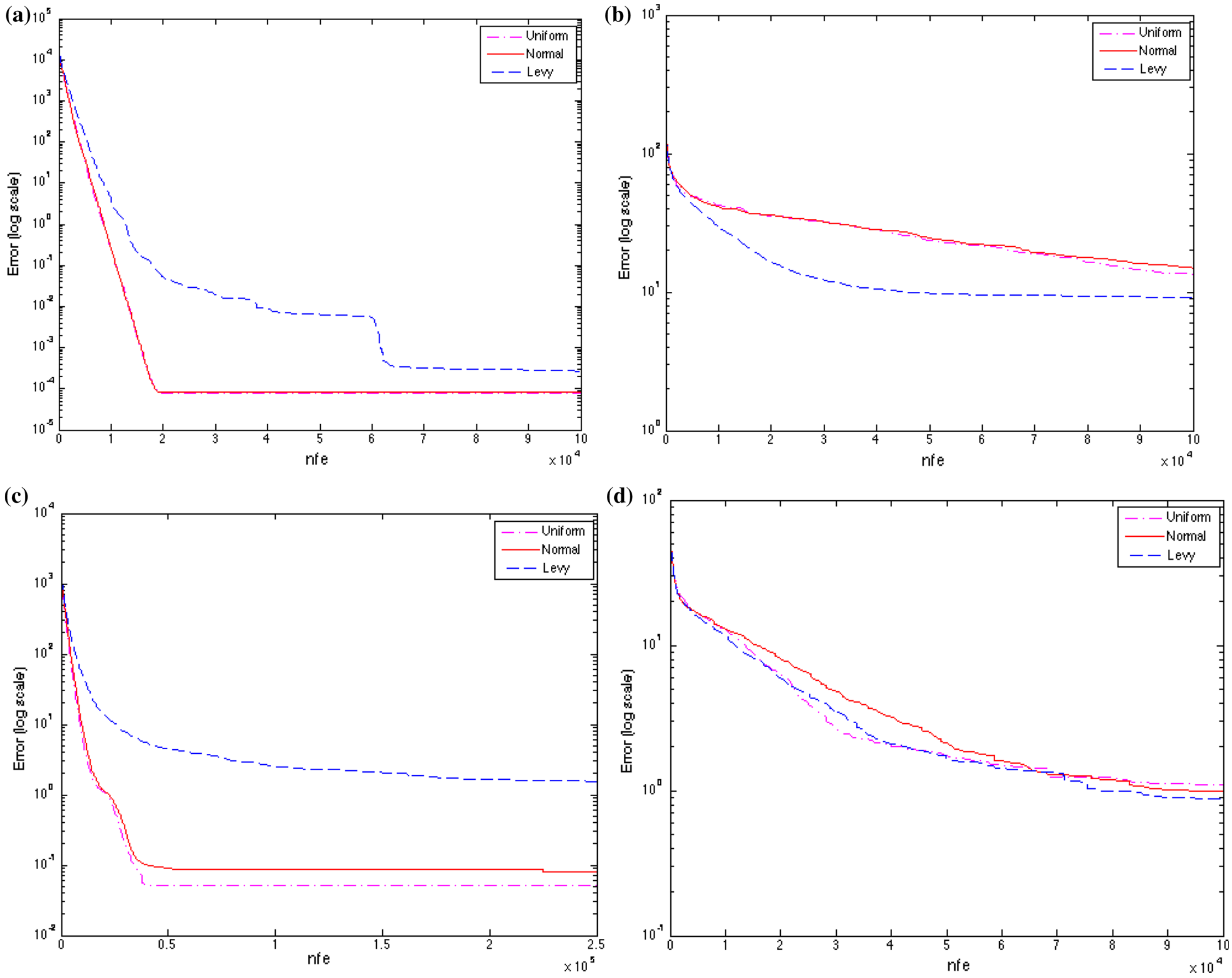

Fig. 3 Mean best error curves of SOS using different random distributions for selected problems. **a** CEC05 f5, **b** CEC05 f10, **c** CEC08 F5, **d** Frequency Modulation

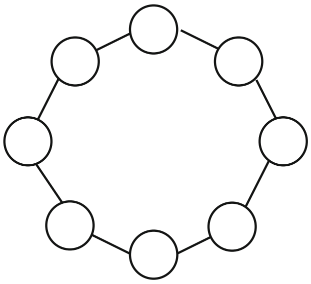

Fig. 4 Ring topology

1. *lbest*_SOS
   The original SOS uses the best organism in the ecosystem, $\boldsymbol{x}_{best}$, in the mutualism and commensalism phases. However, using this model (i.e., a fully connected network structure or a star topology [7] the propagation is very fast. All the individuals in the population will be affected by the best solution found in iteration t, immediately in iteration $t + 1$. This fast propagation may result in a premature convergence problem. However, using a ring topology, as depicted in Fig. 4, will slow down the convergence rate because the best solution found has to propagate through several neighborhoods before affecting all individuals in the population. This slow propagation will enable the individuals to explore more areas in the search space and thus decrease the chance of premature convergence. This is particularly useful for multimodal functions [10].
2. *p*best_SOS
   The best organism in the ecosystem, $\boldsymbol{x}_{\mathrm{best}}$, in Eqs. 1, 2 and 4 is replaced with, $\boldsymbol{x}^{p}_{best}$, which is randomly chosen as one of the top 100p % organisms in the current ecosystem with $p \in (0, 1]$ as in JADE. The p parameter

Table 10 Comparing SOS, *lbest*_SOS, *p*best_SOS and *eco_min_max*_SOS on the CEC05 functions ($D = 10$)

Functions (*p* value)	SOS	*lbest*_SOS	*p*best_SOS	*eco_min_max*_SOS
f1 (0.4505)				
Median	7.22e−05	7.50e−05	7.45e−05	7.49e−05
Mean (SD)	7.34e−05 (1.56e−05)	7.31e−05 (1.98e−05)	7.63e−05 (1.57e−05)	7.09e−05 (1.75e−05)
FEs (SR %)	9.87e+03 (100)	1.44e+04 (100)	1.05e+04 (100)	7.42e+03 (100)
f2 (0.0186)				
Median	**7.40e−05**	*8.91e−05*	7.79e−05	8.59e−05
Mean (SD)	7.52e−05 (1.42e−05)	8.70e−05 (8.99e−06)	7.71e−05 (1.74e−05)	8.16e−05 (1.52e−05)
FEs (SR %)	1.73e+04 (100)	4.13e+04 (100)	2.06e+04 (100)	1.48e+04 (100)
f3 (0.4862)				
Median	6.81e+04	4.45e+04	4.88e+04	3.76e+04
Mean (SD)	9.47e+04 (8.87e+04)	5.55e+04 (3.38e+04)	5.67e+04 (4.46e+04)	5.96e+04 (5.68e+04)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f4 (0.6962)				
Median	8.60e−05	8.70e−05	8.39e−05	8.33e−05
Mean (SD)	8.26e−05 (1.44e−05)	8.55e−05 (9.50e−06)	7.88e−05 (1.73e−05)	7.95e−05 (1.66e−05)
FEs (SR %)	2.10e+04 (100)	4.74e+04 (100)	2.36e+04 (100)	1.77e+04 (100)
f5 (0.6325)				
Median	8.30e−05	8.78e−05	8.51e−05	7.74e−05
Mean (SD)	8.29e−05 (1.26e−05)	8.32e−05 (1.50e−05)	8.13e−05 (1.42e−05)	7.73e−05 (1.38e−05)
FEs (SR %)	1.76e+04 (100)	3.63e+04 (100)	2.00e+04 (100)	1.45e+04 (100)
f6 (4.2271e−07)				
Median	*2.27e−01*	*2.62e+00*	7.69e−02	**9.43e−05**
Mean (SD)	1.09e+01 (4.89e+01)	3.06e+00 (1.71e+00)	6.22e−01 (1.39e+00)	9.82e−01 (2.11e+00)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	7.78e+04 (70)
f7 (2.9264e−04)				
Median	1.27e+03	*1.27e+03*	1.27e+03	**1.27e+03**
Mean (SD)	1.27e+03 (8.44e−14)	1.27e+03 (1.64e−13)	1.27e+03 (1.19e−13)	1.27e+03 (4.22e−14)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f8 (0.6869)				
Median	2.04e+01	2.04e+01	2.04e+01	2.03e+01
Mean (SD)	2.04e+01 (7.30e−02)	2.04e+01 (6.73e−02)	2.03e+01 (7.17e−02)	2.03e+01 (8.61e−02)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f9 (7.0706e−09)				
Median	*2.42e+00*	*2.83e+00*	*3.52e+00*	**9.72e−05**
Mean (SD)	2.85e+00 (1.67e+00)	2.83e+00 (1.66e+00)	3.29e+00 (2.12e+00)	5.97e−01 (8.10e−01)
FEs (SR %)	*x* (0)	*x* (0)	1.00e+05 (3.33)	6.45e+04 (60)
f10 (2.6463e−05)				
Median	*1.36e+01*	9.58e+00	*1.13e+01*	**6.96e+00**
Mean (SD)	1.43e+01 (6.03e+00)	1.09e+01 (5.49e+00)	1.29e+01 (5.18e+00)	7.46e+00 (3.34e+00)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f11 (0.6325)				
Median	2.10e+00	2.32e+00	2.47e+00	1.50e+00
Mean (SD)	2.50e+00 (1.31e+00)	2.43e+00 (1.37e+00)	2.18e+00 (1.27e+00)	1.98e+00 (1.46e+00)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
f12 (0.7149)				
Median	3.58e+03	2.95e+03	3.28e+03	3.03e+03
Mean (SD)	3.42e+03 (1.51e+03)	2.88e+03 (1.31e+03)	3.19e+03 (1.26e+03)	3.29e+03 (1.63e+03)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)

Table 10 continued

Functions (*p* value)	SOS	*lbest*_SOS	*p*best_SOS	*eco_min_max*_SOS
f13 (3.0383e−07)				
Median	*8.22e−01*	*7.35e−01*	*9.88e−01*	**4.08e−01**
Mean (SD)	7.86e−01 (3.74e−01)	7.93e−01 (2.49e−01)	9.80e−01 (3.17e−01)	4.08e−01 (1.35e−01)
FEs (SR %)	4.78e+04 (3.33)	*x* (0)	*x* (0)	*x* (0)
f14 (0.0670)				
Median	3.05e+00	3.19e+00	3.18e+00	3.05e+00
Mean (SD)	3.01e+00 (2.88e−01)	3.18e+00 (1.93e−01)	3.13e+00 (2.11e−01)	3.04e+00 (2.71e−01)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
# of times the algorithm is best	0	1	0	5

Bold is statistically better than italic at 5 % significance level

Table 11 Comparing SOS, *lbest*_SOS, *p*best_SOS and *eco_min_max*_SOS on the CEC08 functions ($D = 50$)

Functions (*p* value)	SOS	*lbest*_SOS	*p*best_SOS	*eco_min_max*_SOS
F1 (0.5479)				
Median	9.50e−05	9.60e−05	9.45e−05	9.42e−05
Mean (SD)	8.90e−05 (1.61e−05)	9.51e−05 (3.99e−06)	9.25e−05 (6.46e−06)	9.07e−05 (1.41e−05)
FEs (SR %)	4.67e+04 (100)	7.35e+04 (100)	4.52e+04 (100)	2.95e+04 (100)
F2 (7.5463e−18)				
Median	*2.68e+01*	**1.69e+01**	*2.43e+01*	*4.69e+01*
Mean (SD)	2.69e+01 (2.93e+00)	1.75e+01 (2.96e+00)	2.48e+01 (2.25e+00)	4.71e+01 (5.11e+00)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
F3 (8.0532e−05)				
Median	8.91e+01	*1.52e+02*	8.11e+01	**7.16e+01**
Mean (SD)	9.69e+01 (6.63e+01)	1.62e+02 (9.28e+01)	7.90e+01 (3.43e+01)	6.10e+01 (3.75e+01)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
F4 (0.0027)				
Median	1.09e+02	*1.22e+02*	**1.05e+02**	**1.03e+02**
Mean (SD)	1.07e+02 (1.94e+01)	1.21e+02 (2.01e+01)	1.03e+02 (2.36e+01)	1.04e+02 (1.74e+01)
FEs (SR %)	*x* (0)	*x* (0)	*x* (0)	*x* (0)
F5 (0.0786)				
Median	8.63e−03	9.88e−05	9.82e−05	1.35e−02
Mean (SD)	2.68e−02 (3.29e−02)	4.07e−03 (7.15e−03)	1.30e−02 (2.44e−02)	3.80e−02 (4.86e−02)
FEs (SR %)	4.60e+04 (43.3)	8.51e+04 (70)	4.62e+04 (56.7)	2.93e+04 (36.7)
F6 (7.2428e−11)				
Median	*2.84e+00*	**1.32e+00**	1.95e+00	*2.37e+00*
Mean (SD)	2.88e+00 (4.63e−01)	1.34e+00 (4.73e−01)	1.93e+00 (3.46e−01)	2.66e+00 (1.44e+00)
FEs (SR %)	*x* (0)	2.50e+05 (3.33)	*x* (0)	*x* (0)
# of times the algorithm is best	0	2	1	2

Bold is statistically better than italic at 5 % significance level

determines the greediness of the approach. In this paper, p is set to 0.1 that is within the [0.05,0.2] range suggested by Zhang and Sanderson [20].

3. *eco_min_max*_SOS

In the parasitism phase, the parasite vector, $\boldsymbol{x}_p$, is created in the search space by duplicating organism $\boldsymbol{x}_i$ and modifying randomly chosen dimensions using a random number, i.e.,

$$x_{p,d} = l_d + \epsilon_4[u_d - l_d]$$

Table 12 Comparing SOS, *lbest*_SOS, *p*best_SOS and *eco_min_max*_SOS on the engineering problems

Functions (*p* value)	SOS	*lbest*_SOS	*p*best_SOS	*eco_min_max*_SOS
Compression Spring (0.8964)				
Median	6.75e−05	6.07e−05	5.15e−05	6.66e−05
Mean (SD)	5.97e−05 (3.09e−05)	5.50e−05 (3.34e−05)	5.44e−05 (2.86e−05)	5.79e−05 (2.76e−05)
FEs (SR %)	1.66e+04 (100)	2.84e+04 (100)	1.78e+04 (100)	1.53e+04 (100)
Gear Train (0.1427)				
Median	1.70e−06	4.25e−06	1.70e−06	2.02e−06
Mean (SD)	1.07e−05 (2.20e−05)	1.62e−05 (2.31e−05)	1.13e−05 (2.23e−05)	1.23e−05 (2.24e−05)
FEs (SR %)	2.17e+02 (100)	2.17e+02 (100)	2.17e+02 (100)	2.17e+02 (100)
Pressure Vessel (0.6414)				
Median	4.36e−05	4.63e−05	5.39e−05	3.23e−05
Mean (SD)	4.85e−05 (3.04e−05)	4.93e−05 (3.08e−05)	5.20e−05 (2.84e−05)	3.85e−05 (3.02e−05)
FEs (SR %)	3.02e+04 (100)	4.57e+04 (100)	3.14e+04 (100)	1.45e+04 (100)
Frequency Modulation (0.1577)				
Median	6.89e−05	7.54e−05	8.25e−05	9.25e−05
Mean (SD)	6.69e−01 (2.22e+00)	4.96e−02 (2.51e−01)	1.79e−01 (9.79e−01)	3.15e+00 (4.57e+00)
FEs (SR %)	5.17e+04 (90)	6.39e+04 (90)	6.21e+04 (96.7)	1.55e+04 (60)
Lennard-Jones (0.0658)				
Median	8.94e−05	8.75e−05	9.32e−05	8.06e−05
Mean (SD)	4.27e−04 (1.86e−03)	1.88e−04 (5.86e−04)	1.20e−02 (5.25e−02)	7.71e−05 (1.59e−05)
FEs (SR %)	5.40e+04 (90)	6.93e+04 (96.7)	6.13e+04 (73.3)	4.46e+04 (100)
# of times the algorithm is best	0	0	0	0

Bold is statistically better than italic at 5 % significance level

where $\epsilon_4 \sim U(0,1)$, l_d and u_d are the lower and upper bounds of the *d*th decision variable. In the proposed variant, l_d and u_d are replaced with the minimum and maximum values of the *d*th decision variable in the current ecosystem.

Table 10 shows that the three proposed methods perform better than or equal to the original SOS on the CEC05 functions. On unimodal functions, there is generally no significant difference between the four methods. However, on the more interesting multimodal functions, *eco_min_max*_SOS is the best approach. The results show also that *eco_min_max*_SOS is the most efficient approach since it requires less number of function evaluations to reach the "optimal" solution. This is another advantage for this variant.

For high-dimensional CEC08 functions, Table 11 shows that on multimodal functions *eco_min_max*_SOS performs better on two out of the four functions. In addition, *eco_min_max*_SOS is the most efficient approach as in Table 10.

On the engineering problems, Table 12 shows that there is no significant difference between the four approaches. However, as in Tables 10 and 11, *eco_min_max*_SOS is the most efficient approach.

Based on the results reported in Tables 10, 11 and 12, it seems that *eco_min_max*_SOS is generally a viable and efficient alternative to the original SOS.

Figure 5 shows the convergence behavior of SOS and its variants on four representative problems. It is clear from the figure that SOS exhibits faster convergence.

Therefore, we can conclude from the results reported in this section that $\boldsymbol{l}$ and $\boldsymbol{u}$ may give exceedingly "loose" bounds on the decision variables, and hence, the parasitism phase (and consequently SOS) can be improved by a process that successively revises $\boldsymbol{l}$ and $\boldsymbol{u}$ based on solution history.

8 Conclusions

In this paper, the symbiosis organisms search (SOS) algorithm was empirically studied on a wide range of problems. First, SOS has been compared to three recent metaheuristic algorithms. The results showed that SOS suffered when used to solve high-dimensional problems. However, it worked well on the five engineering problems used in this study. Second, the effect of the only SOS parameter, *eco_size*, is investigated. The results suggested that as the problem's dimension increases, *eco_size* should be increased. Hence, an adaptive formula could be used to automatically tune the *eco_size* parameter. Third, different random walks have then been used in the commensalism phase. In general, it seems that uniform and normal

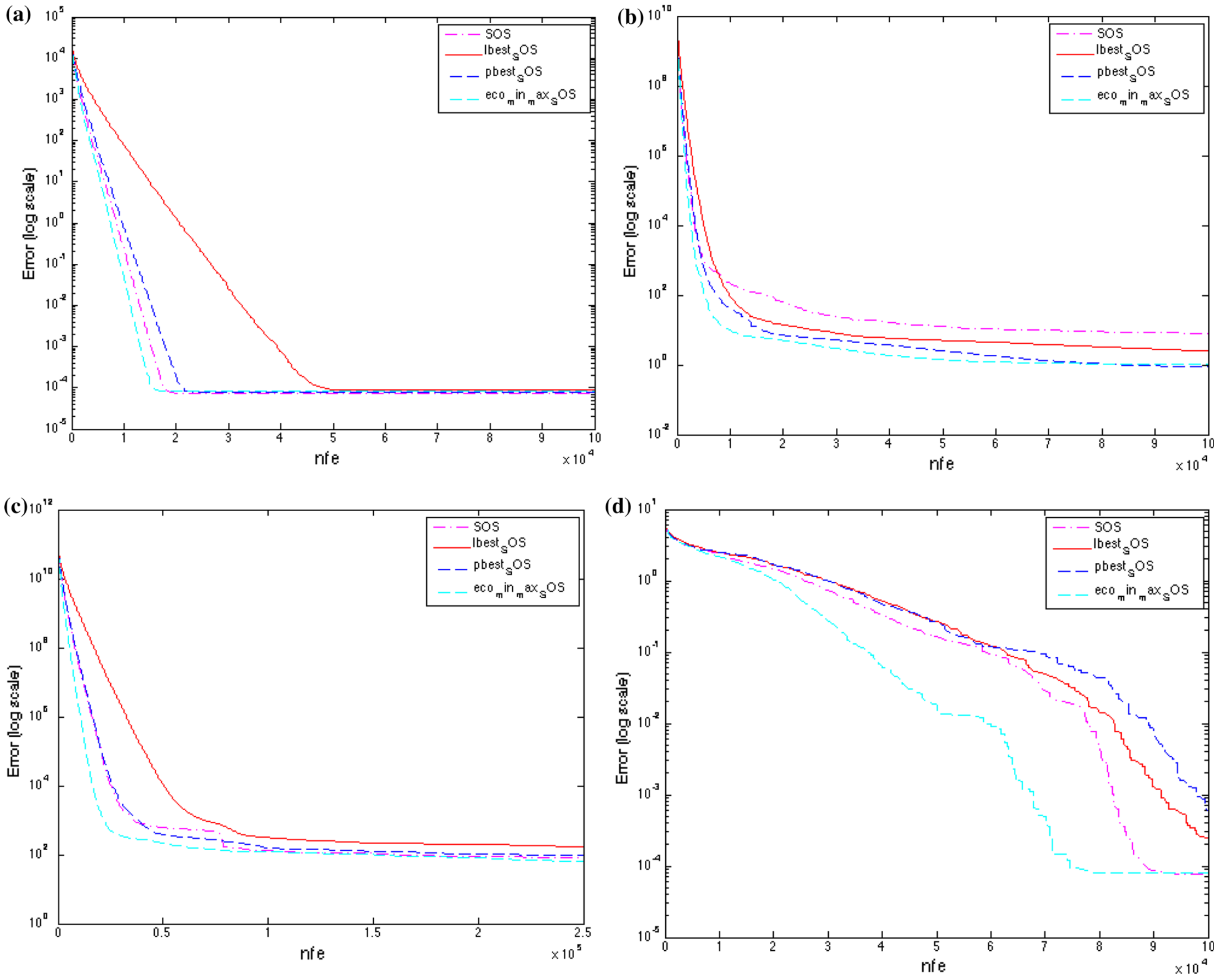

Fig. 5 Mean best error curves of SOS, *lbest*_SOS, *p*best_SOS and *eco_min_max*_SOS for selected problems. **a** CEC05 f2, **b** CEC05 f6, **c** CEC08 F3, **d** Lennard-Jones

distributions are better than the Lévy distribution except for the low-dimensional multimodal functions. Finally, three simple SOS variants have been proposed and the results suggest that *eco_min_max*_SOS is a promising alternative to SOS.

Acknowledgments The authors would like to thank the anonymous reviewers for their constructive and helpful comments and suggestions.

References

1. Cheng M, Prayogo D (2014) Symbiotic organism search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112
2. Civicioglu P (2013) Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput 219(15):8121–8144
3. Das S, Suganthan P (2010) Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Technical Report, Jadavpur University, Nanyang Technological University
4. Garcia S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. J Heuristics 15(6):617–644
5. Jones D (2010) Good practice in (pseudo) random number generation for bioinformatics applications. Technical Report, UCL Bioinformatics Group
6. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department
7. Kennedy J (1999) Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In: Proceedings of IEEE congress on evolutionary computation, Washington DC, USA, pp 1931–1938
8. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the IEEE international joint conference on neural networks. IEEE Press, pp. 1942–1948
9. Matsumoto M, Nishimura T (1998) Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Trans Model Comput Simul 8(1):3–30

10. Peer E, Van den Bergh F, Engelbrecht A (2003) Using neighborhoods with the guaranteed convergence PSO. In: Swarm intelligence symposium, Piscataway, New Jersey, USA, IEEE Service Center, pp. 235–242
11. Sandgren E (1990) Non linear integer and discrete programming in mechanical design optimization. J Mech Des 112(2):223–229
12. Simon D (2013) Evolutionary optimization algorithms. Wiley, New York
13. Sorensen K (2015) Metaheuristics—the metaphor exposed. Int Trans Oper Res 22(1):3–18
14. Sorensen K, Sevaux M, Glover F (2016) History of metaheuristics. Handbook of heuristics. Springer, New York
15. Storn R, Price K (1995) Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA
16. Suganthan P, Hansen N, Liang J, Deb K, Chen Y, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC2005 special session on real-parameter optimization. Technical Report, Nanyang Technology University, Singapore
17. Tang K, Yao X, Suganthan PN, MacNish C, Chen YP, Chen CM, Yang Z (2008) Benchmark functions for the CEC'2008 special session and competition on large scale global optimization. Technical Report, Nature Inspired Computation and Applications Laboratory, USTC, China
18. Yang X (2012) Flower pollination algorithm for global optimization. Lect Notes Comput Sci 7445:240–249
19. Yang X (2014) Nature-inspired optimization algorithms. Elsevier, Amsterdam
20. Zhang J, Sanderson A (2009) JADE: adaptive differential evolution with optional external archive. IEEE Trans Evol Comput 13(5):945–958
21. Zhan Z, Zhan J, Li Y, Chung H (2009) Adaptive particle swarm optimization. IEEE Trans Syst Man Cybern Part B 39(6):1362–1381
22. Zhou A, Sun J, Zhang Q (2015) An estimation of distribution algorithm with cheap and expensive local search methods. IEEE Trans Evol Comput 19(6):807–822