Advances in Intelligent Systems and Computing 1139

Atulya K. Nagar
Kusum Deep
Jagdish Chand Bansal
Kedar Nath Das *Editors*

Soft Computing for Problem Solving 2019

Proceedings of SocProS 2019, Volume 2

Springer

Advances in Intelligent Systems and Computing

Volume 1139

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within “Advances in Intelligent Systems and Computing” are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

**** Indexing: The books of this series are submitted to ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink ****

More information about this series at http://www.springer.com/series/11156

Atulya K. Nagar · Kusum Deep ·
Jagdish Chand Bansal · Kedar Nath Das
Editors

Soft Computing for Problem Solving 2019

Proceedings of SocProS 2019, Volume 2

Springer

Editors
Atulya K. Nagar
School of Mathematics,
Computer Science and Engineering
Liverpool Hope University
Liverpool, UK

Kusum Deep
Department of Mathematics
Indian Institute of Technology Roorkee
Roorkee, India

Jagdish Chand Bansal
Department of Mathematics
South Asian University
New Delhi, Delhi, India

Kedar Nath Das
Department of Mathematics
National Institute of Technology Silchar
Silchar, India

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-981-15-3286-3 ISBN 978-981-15-3287-0 (eBook)
https://doi.org/10.1007/978-981-15-3287-0

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

We are delighted that the 9th International Conference on Soft Computing for Problem Solving, SocProS 2019, took place at Liverpool Hope University, UK, during 02–04 September 2019. This was a particularly important event as it coincided with the 175th anniversary of the foundation of Liverpool Hope University (LHU). The SocProS conference series has a glorious history; the earlier editions of the conference have been organised in various prestigious institutions of India. For the very first time, this event in the series of the conference was hosted outside of India at Liverpool in the UK. This is significant because along with IIT Roorkee and South Asian University in India, LHU has been one of the key institutions that initiated this prestigious meeting. Continuing the trend, once again the 9th edition of this conference touched many milestones in terms of quality research papers and fruitful discussions. The theme of SocProS 2019 was "*Unlocking the Power and Impact of Artificial Intelligence*".

This proceedings as an outcome of the 9th meeting of the SocProS community includes a collection of selected high-quality articles on various topics related to soft computing and artificial intelligence and their applications. The book is being prepared in two volumes to cover the recent advances and challenges in the themes of machine learning, neural networks, scientific computing, and intelligent systems and includes several chapters addressing the problems arising in real-life applications comprising that of image classification, deep learning, fuzzy systems, flow shop scheduling, support vector machines, mobile robot path planning, P-systems, machine learning, and spiking neural networks, to name a few contributions. We have also tried to capture the impact aspects of research in this area, particularly impact beyond the academic world. We have made further efforts in this direction to embed impact as part of our conference series, and going forward we very much hope that, as was agreed at the conference, we will continue to mainstream impact in our work and intensify our efforts to reach out to non-academic beneficiaries and users to realise the impact from our research.

Highlighting theoretical perspectives and empirical research, it is hoped that this edited volume will prove to be a comprehensive reference source for researchers, practitioners, students, and professionals interested in the current advancements and efficient use of soft computing as well as in making the impact happen. We express our heartfelt gratitude to all the authors, reviewers, and Springer personnel for their motivation and patience.

Liverpool, UK
September 2019

Atulya K. Nagar
Kusum Deep
Jagdish Chand Bansal
Kedar Nath Das

Contents

About the Editors

Atulya K. Nagar holds the Foundation Chair as Professor of Mathematical Sciences, and is the Pro-Vice-Chancellor for Research and Dean of the Faculty of Science at Liverpool Hope University, United Kingdom. He is also the Head of the School of Mathematics, Computer Science and Engineering, which he established at the University. He is an internationally respected scholar working at the cutting edge of theoretical computer science, applied mathematical analysis, operations research, and systems engineering. He received a prestigious Commonwealth Fellowship to pursue his doctorate (DPhil) in Applied Nonlinear Mathematics, which he earned from the University of York (UK) in 1996. He holds a BSc (Hons), an MSc and MPhil (with distinction) in Mathematical Physics from the MDS University of Ajmer, India. His research expertise spans both applied mathematics and computational methods for nonlinear, complex, and intractable problems arising in science, engineering and industry.

Prof. Kusum Deep is a Professor at the Department of Mathematics, Indian Institute of Technology Roorkee. Her research interests include numerical optimization, nature inspired optimization, computational intelligence, genetic algorithms, parallel genetic algorithms, and parallel particle swarm optimization.

Dr. Jagdish Chand Bansal is an Assistant Professor at the South Asian University, New Delhi, India and visiting research fellow at Liverpool Hope University, UK. He has an excellent academic record and is a leading researcher in the field of swarm intelligence. He has published numerous research papers in respected international and national journals.

Dr. Kedar Nath Das is an Assistant Professor at the Department of Mathematics, National Institute of Technology, Silchar, Assam, India. Over the past 10 years, he has made substantial contributions to research on soft computing, and has published several papers in prominent national and international journals. His chief area of interest is evolutionary and bio-inspired algorithms for optimization.

Exponential Adaptive Strategy in Spider Monkey Optimization Algorithm

Apoorva Sharma, Nirmala Sharma, Harish Sharma, and Jagdish Chand Bansal

Abstract Spider monkey optimization algorithm (SMOA) is one of the powerful techniques in the arena of swarm intelligence (SI)-based strategies. This article proposes a modified variant of SMOA that is based on an exponential adaptive strategy for step size. During the search of the optimal solution, this exponential strategy is used to adjust the step size so that it can speed up the convergence ability of the swarm. The proposed algorithm is termed as exponential adaptive spider monkey optimization (EASMO) algorithm. This evinced algorithm is tested over 14 standard optimization problems to examine its authenticity. Further, the obtained results are compared with the artificial bee colony (ABC), differential evolution (DE), Gbest-guided artificial bee colony (GABC), particle swarm optimization (PSO), SMOA, and three of its momentous variants, namely levy flight SMOA (LFSMOA), modified limacon SMOA (MLSMOA), and power law-based local search in SMOA (PLSMOA). The analysis of the results proved the competence of EASMO in the field of SI-based strategies.

Keywords Population-based optimization · Spider monkey optimization algorithm · Swarm intelligence-based algorithms · Exponential adaptive strategy

A real one.

A. Sharma (✉) · N. Sharma · H. Sharma
Rajasthan Technical University, Kota, Rajasthan, India
e-mail: apoorvasharma2628@gmail.com

N. Sharma
e-mail: nsharma@rtu.ac.in

H. Sharma
e-mail: harish.sharma0107@gmail.com

J. Chand Bansal
South Asian University, New Delhi, India
e-mail: jcbansal@gmail.com

A. K. Nagar et al. (eds.), *Soft Computing for Problem Solving 2019*,
Advances in Intelligent Systems and Computing 1139,
https://doi.org/10.1007/978-981-15-3287-0_1

1 Introduction

Researchers have analyzed the behavior of several systems in nature that is able to cope efficiently with many optimization problems (OPs), and these systems could be used to resolve various complex optimization queries [2]. Population-based optimization algorithms are used to find near-optimal solutions for difficult OPs [3]. Evolutionary algorithms (EAs) and swarm intelligence (SI)-based algorithms are two essential categories of population-based optimization algorithms. EAs use mechanisms motivated by biological evolution, like selection, crossover, mutation, etc. [4]. SI is a metaheuristic method in the field of artificial intelligence (AI). To solve actual world OPs, SI-dependent algorithms are used because of its significant potential [5, 11, 12, 20–23]. Ant colony optimization (ACO) [2], particle swarm optimization (PSO) [8], bacterial foraging optimization (BFO) [10], artificial bee colony (ABC) optimization [7], etc. are some significant SI-based algorithms. Spider monkey optimization algorithm (SMOA) also come under the category of SI-based algorithms. Food searching behavior of spider monkeys is used as a basis for developing SMOA. With the ability to solve numerical OPs, it also has a inherit drawback like other population-based optimization algorithms such as converges to the local optima probabilistically and sometimes get stuck at local optima [1].

To maintain an appropriate consistency between exploration and exploitation, it is necessary for the algorithm to first use the variation process for few iterations and then use the selection process for the rest of the iterations. In order to reduce the chances to jump the optimal solution and enhance the exploitation proficiency of the SMOA, an exponential adaptive strategy is applied with SMOA. This adaptive strategy decreases the step size exponentially. The propound algorithm is titled as an exponential adaptive spider monkey optimization (EASMO) algorithm. The authenticity of the proposed EASMO is evaluated using 14 benchmark optimization functions. The obtained outcomes are analyzed and compared with eight state-of-the-art algorithms. The obtained results validate the proposed approach.

The arrangement of the remaining article is as follows: Sect. 2 discusses SMOA. Section 3 comprises exponential adaptive strategy-based SMOA (EASMO). Various numerical standard problems are used to investigate the performance of EASMO, narrated in Sect. 4. Conclusively, Sect. 5 winds up the work and assimilates the summary.

2 Spider Monkey Optimization Algorithm (SMOA)

This stochastic optimization method, proposed by Bansal et al. [1], is inspired by the social nature of spider monkeys (SMs). These SMs come in the class of fission–fusion social structure (FFSS)-based animals. In the food searching process, the FFSS-based animals separated themselves into subgroups. A female more often leads the group and if that female leader will not get plenteous food for that group, the group will be divided into small subgroups. Depending upon the accessibility of food sources, the members of the subgroups impart inside and outside [1].

At first, the population of N spider monkeys is uniformly distributed in the given search space as expressed into Eq. 1:

$$SM_{ij} = SM_{minj} + U(0, 1) \times (SM_{maxj} - SM_{minj}) \tag{1}$$

here each monkey SM_i $(i = 1, 2, \ldots, N)$ is a D-dimensional vector and SM_{ij} specify the jth dimension of the ith spider monkey, SM_{minj} and SM_{maxj} are the lower limit and upper limit of the ith spider monkey in jth dimension and $U(0, 1)$ is the evenly assigned arbitrary values specified in the interval [0, 1].

The SMOA comprises six stages, namely Local Leader Stage, Global Leader Stage, Local Leader Learning Stage, Global Leader Learning Stage, Local Leader Decision Stage, and Global Leader Decision Stage. Firstly, like the other SI-based algorithms, the swarm of the SMOA is initialized in the given search space. After that, SMOA stages evolve the swarm to reach the global optima.

2.1 Local Leader Stage (LLS)

During this stage, the position of every SM is updated relied upon the local leader's position and the position of randomly selected neighbor within the group. Then, the fitness of the new solution is calculated, and by applying the greedy selection, the best fitted SM is adopted. The position update equation for ithSM (kth local group member) is expressed by Eq. 2:

$$SMnew_{ij} = SM_{ij} + U(0, 1) \times (LL_{kj} - SM_{ij}) + U(-1, 1) \times (SM_{rj} - SM_{ij}) \tag{2}$$

where SM_{ij} is the jth dimension of the ith spider monkey, SM_{rj} is the jth dimension of the arbitrary choosen rth spider monkey within the group k such that $r \neq i$. LL_{kj} is the jth dimension of the the kth local group leader.

2.2 Global Leader Stage (GLS)

After the execution of the LLS, GLS is executed. In the GLS, all the SMs are updated as per the knowledge received from the global leader and the members of the local group. The position update process takes place as per Eq. 3:

$$SMnew_{ij} = SM_{ij} + U(0, 1) \times (GL_j - SM_{ij}) + U(-1, 1) \times (SM_{rj} - SM_{ij}) \tag{3}$$

where GL_j depicts the position of the global leader in the jth direction. The term $j \in \{1, 2, ..., D\}$ is the arbitrarily choosen index. In this stage, more chances are given to the favorable solutions to make them better and for this purpose, a probability factor $prob_i$ is introduced. Here, the position of the SMs is updated on the basis of probability. The $prob_i$ is calculated by using the following equation:

$$prob_i = 0.9 \times \frac{fitness_i}{max\_fitness} + 0.1 \tag{4}$$

here, $max\_fitness$ is the maximum fitness, and $fitness_i$ is the fitness of ith SM of the group.

2.3 Global Leader Learning Stage (GLLS)

Here, the greedy selection method is employed on the population for updating the global leader's position. If the global leader's position is not updated up to a predetermined number of times, then the GlobalLimitCount is raised by 1.

2.4 Local Leader Learning Stage (LLLS)

The global leader's position is updated by applying the greedy selection method. Furthermore, the LocalLimitCount is raised by 1 when the local leader is not updating its location.

2.5 Local Leader Decision Stage (LLDS)

If a group's local leader is not updating its position according to a predestined threshold called LocalLeaderLimit, then by doing arbitrary initialization or by using the global and local leader's information, the position of the individuals from that group is updated.

2.6 Global Leader Decision Stage (GLDS)

If the position of the global leader is not updating up to a predetermined threshold extent known as GlobalLeaderLimit, then the global leader takes a decision and the population is split in the form of subgroups. The limit for the maximum number of small groups is defined by MG which is defined in Sect. 4.

3 Exponential Adaptive Strategy-Based SMOA

For the efficient performance of any population-based optimization algorithm, exploration and exploitation should be well balanced [3, 6, 9, 18]. In Local Leader Phase (LLP), the new solution is updated by learning from its previous experience as well as learning from the local leaders and the local group members [1]. Here, the chances for skipping the true solutions and stuck into the local optima are high. So balancing of step size should be managed in a way so that the chances of getting success in achieving the global optima will become maximum. Therefore to enhance the exploitation capability of SMOA, an exponential adaptive strategy [17] is applied in the LLS of SMOA. Equation 2 of LLS is modified as per Eq. 5.

$$SMnew_{ij} = SM_{ij} + \alpha(iteration) \times (LL_{kj} - SM_{ij}) + U(-1, 1) \times (SM_{rj} - SM_{ij}) \tag{5}$$

and the step size $\alpha(iteration)$ during each iteration is calculated by using Eq. 6.

$$\alpha(iteration) = \alpha_{max}\left(\frac{itr_{max} - iteration + 1}{itr_{max}}\right)^{\theta} \tag{6}$$

$$\theta = \frac{log(\frac{\alpha_{min}}{\alpha_{max}})}{log(\frac{1}{itr_{max}})} \tag{7}$$

here, α_{max} and α_{min} are the maximum and minimum step size reduction coefficient, itr_{max} is the maximum generation number, and iteration is the current generation number of the algorithm. θ is a coefficient, and its value is calculated by using Eq. 7.

In the first few iterations, α enhance the exploration capabilities by having larger values. As iteration increases, α decreases. Hence, it will decrease the step size exponentially for further iterations and improves the exploitation capability as well as the convergence speed. In the developed strategy, only the LLS is modified, while other stages are kept the same as mentioned in the basic SMOA. The sensitivity analysis of α with respect to iteration is depicted in Fig. 1. The graph depicts that the value of α decreases with the increase in the number of iterations.

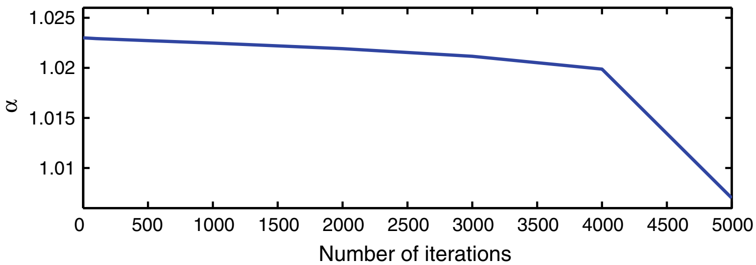

Fig. 1 Graph between α and the number of iterations

On the basis of the above clarification, modified LLS is depicted in Algorithm 1.

for $each\ member\ SM_i \in k\text{th}\ group$ **do**
  for $each\ j \in \{1, ..., D\}$ **do**
    if $U(0, 1) \geq pr$ **then**
      Calculate the value of θ as per equation 7
      Calculate the value of $\alpha(iteration)$ as per equation 6
      $SMnew_{ij} = SM_{ij} + \alpha(iteration) \times (LL_{kj} - SM_{ij}) +$
      $U(-1, 1) \times (SM_{rj} - SM_{ij})$
    else
      $SMnew_{ij} = SM_{ij}$
    end if
  end for
end for

Algorithm 1: Position Update Process in LLS

4 Discussion and Analysis

To investigate the pursuance of the propound EASMO, 14 distinctive global optimization problems (OP_1 to OP_{14}) are chosen as depicted in Table 1.

To approve the compatibility of the EASMO, a comparative analysis is carried out among EASMO, ABC [7], DE [18], GABC [24], PSO [19], SMOA [1], and three of its variant, namely LFSMOA [14], MLSMOA [15], and PLSMOA [13]. Experimental setup for the comparison is as follows:

- The numbers of simulations = 100,
- Population Size $N = 50$,
- $GlobalLeaderLimit = N$,
- $LocalLeaderLimit = D \times N$,
- $pr = 0.1 + (\frac{0.4}{Max\_iterations})$,
- $\alpha_{max} = 1.020$,
- $\alpha_{min} = 1.007$,
- $MG = N/10$,
- The parameter settings for SMOA [1], ABC [7], DE [18], GABC [24], PSO [19], LFSMOA [14], MLSMOA [15] and PLSMOA [13] are taken from the original research.
- To study the impact of the parameters α_{min} and α_{max} on the performance of EASMO, its susceptibility with respect to various values of α_{max} and α_{min} in the range of [0, 2], is examined in Figs. 2 and 3. Figures 2 and 3 show that the algorithm is exceptionally delicate toward α_{max} and α_{min} and their value 1.020 for α_{max} and 1.007 for α_{min} give comparatively better results.Therefore,$\alpha_{max} = 1.02$ and $\alpha_{min} = 1.007$ are choosed in this paper for the experiments.

Table 1 OP: Optimization problem, D: Dimension, AE: Acceptable error

Objective functions	Search range	Optimum value	D	AE
$OP_1(x) = \sum_{i=1}^{D}(100(x_{i+1} - x_i^2)^2 + (x_{i-1})^2)$	[–30, 30]	$OP(\mathbf{1}) = 0$	30	1.0E−02
$OP_2(x) = \sum_{i=1}^{D} \lvert x_i \sin x_i + 0.1x_i \rvert$	[–10, 10]	$OP(0) = 0$	30	1.0E−05
$OP_3(x) = \sum_{i=1}^{D} {x_i}^2 + (\sum_{i=1}^{D} \frac{ix_i}{2})^2 + (\sum_{i=1}^{D} \frac{ix_1}{2})^4$	[–5.12, 5.12]	$OP(0) = 0$	30	1.0E−02
$OP_4(x) = -\sum_{i=1}^{D-1}\left(\exp\left(\frac{-(x_i^2+x_{i+1}^2+0.5x_ix_{i+1})}{8}\right) \times I\right)$ where, I $= cos\left(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5x_ix_{i+1}}\right)$	[–5, 5]	$OP(0) = -D+1$	10	1.0E−05
$OP_5(x) = [1.5 - x_1(1-x_2)]^2 + [2.25 - x_1(1-x_2^2)]^2 + [2.625 - x_1(1-x_2^3)]^2$	[–4.5, 4.5]	$OP(3, 0.5) = 0$	2	1.0E−05
$OP_6(x) = 100[x_2 - x_1^2]^2 + (1-x_1)^2 + 90(x_4 - x_3^2)^2 + (1-x_3)^2 + 10.1[(x_2-1)^2 + (x_4-1)^2] + 19.8(x_2-1)(x_4-1)$	[–10, 10]	$OP(1) = 0$	4	1.0E−05
$OP_7(x) = \sum_{i=1}^{D} z_i^2 + f_{bias}, \quad z = x - o \quad ,x = [x_1, x_2, ...x_D], o = [o_1, o_2, ...o_D]$	[–100, 100]	$OP(o) = OP_{bias} = -450$	10	1.0E−05
$OP_8(x) = -\cos x_1 \cos x_2 e^{(-(x_1-\pi)^2 - (x_2-\pi)^2)}$	[–100, 100]	$OP(\pi, \pi) = -1$	2	1.0E−13
$OP_9(x) = 10^5 x_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + 10^{-5}(x_1^2 + x_2^2)^4$	[–20, 20]	$OP(0, 15) = F(0, -15) = -24777$	2	5.0E−01
$OP_{10}(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - \frac{3}{2}x_1 + \frac{5}{2}x_2 + 1$	$-1.5 \leq x_1 \leq 4, -3 \leq x_2 \leq 3$	$OP(-0.547, -1.547) = -1.9133$	2	1.0E−04
$OP_{11}(x) = -\sum_{i=1}^{5} i\cos((i+1)x_1 + 1)\sum_{i=1}^{5} i\cos((i+1)x_2 + 1)$	[–10, 10]	$OP(7.0835, 4.8580) = -186.7309$	2	1.0E−05
$OP_{12}(x) = -[A\prod_{i=1}^{D}\sin(x_i - z) + \prod_{i=1}^{D} sin(B(x_i - z))]$, $A = 2.5, B = 5, z = 30$	[0, 180]	$OP(90+z) = -(A+1)$	10	1.0E−02
$OP_{13}(x) = 4\varepsilon[(\frac{\sigma}{\gamma})^{12} - (\frac{\sigma}{\gamma})^6] = \varepsilon[(\frac{\gamma m}{\gamma})^{12} - 2(\frac{\gamma m}{\gamma})^6]$	[–2, 2]	−9.103852	15	1.0E−03
$OP_{14}(x) = -\sum_{i=1}^{D} x_i sin(\sqrt{\lvert x_i \rvert})$	[–100, 100]	$OP(0) = 0$	30	1.0E−03

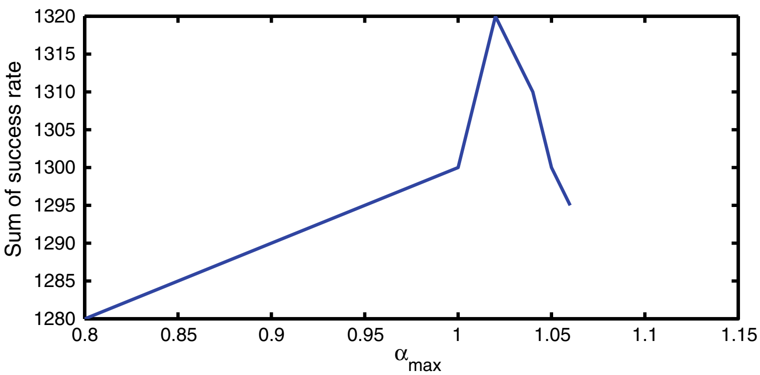

Fig. 2 Effect of parameter α_{max} on success rate

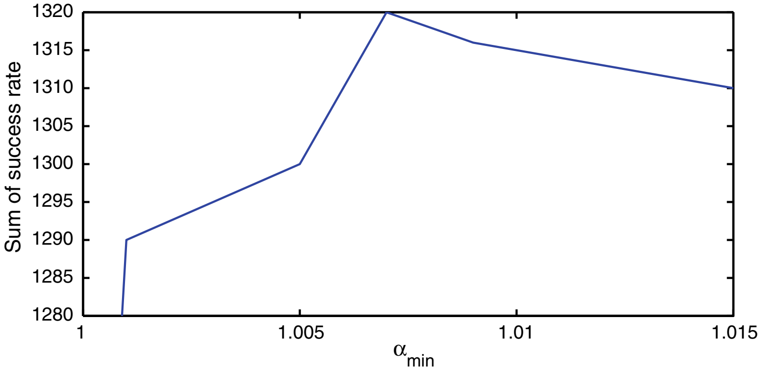

Fig. 3 Effect of parameter α_{min} on success rate

The experimental outcomes for the considered algorithms are exhibited in Table 2. Table 2 represents the results in the form of standard deviation (SD), average count of function evaluations ($AFEs$), mean error (ME) along with the success rate (SR). Results in Table 2 reveal that the EASMO outplays in the form of efficiency, accuracy as well as reliability in comparison with the considered algorithms.

Table 2 Collation of the outcome of optimization problem, OP: Optimization problem

OP	Algorithm	SD	ME	AFEs	SR
OP_1	EASMO	1.28E+05	2.66E+04	327757.89	51
	SMOA	2.77E+01	2.04E+01	507418.68	5
	ABC	2.59E+00	9.58E−01	240192.00	10
	GABC	9.73E+00	4.48E+00	243403.08	6
	DE	6.24E+01	3.62E+01	250050.00	0
	PSO	4.75E+01	4.07E+01	250050.00	0
	LFSMOA	4.36E+01	4.81E+01	200069.05	0
	MLSMOA	2.18E+03	1.41E+03	200495.00	0
	PLSMOA	3.70E+01	3.43E+01	200056.07	0
OP_2	EASMO	2.40E−06	8.12E−06	69252.84	100
	SMOA	6.19E−07	9.44E−06	83959.22	100
	ABC	1.80E−06	8.30E−06	76266.50	100
	GABC	2.18E−06	7.97E−06	57913.04	100
	DE	9.26E−07	9.30E−06	63978.50	100
	PSO	4.00E−07	9.58E−06	94751.50	100
	LFSMOA	3.38E−04	1.23E−04	161894.07	60
	MLSMOA	3.83E−01	4.55E−01	200495.00	0
	PLSMOA	5.22E−04	1.21E−04	139888.24	68

(continued)

Table 2 (continued)

OP	Algorithm	SD	ME	AFEs	SR
OP_3	EASMO	8.89E−04	9.29E−03	91321.87	100
	SMOA	8.03E−04	9.33E−03	132125.82	100
	ABC	1.44E+01	9.10E+01	250000.00	0
	GABC	1.76E+01	9.16E+01	250000.00	0
	DE	1.19E−03	9.10E−03	71045.50	100
	PSO	5.37E−04	9.58E−03	207048.00	100
	LFSMOA	7.84E−04	9.32E−03	161360.01	100
	MLSMOA	1.21E−02	3.48E−02	200495.00	0
	PLSMOA	5.65E−04	9.61E−03	115965.19	100
OP_4	EASMO	3.67E−06	5.58E−06	45487.03	100
	SMOA	1.70E−06	8.06E−06	79609.01	100
	ABC	1.60E−04	2.43E−05	91044.53	97
	GABC	2.42E−06	7.10E−06	44735.63	100
	DE	6.48E−01	9.74E−01	219967.00	16
	PSO	6.40E−01	1.32E+00	243330.00	5
	LFSMOA	1.08E−01	1.33E−02	85488.54	95
	MLSMOA	8.60E−01	6.17E−01	199695.00	2
	PLSMOA	9.59E−02	1.34E−02	69924.43	97
OP_5	EASMO	2.99E−06	4.55E−06	1054.35	100
	SMOA	2.97E−06	4.80E−06	1478.07	100
	ABC	1.62E−06	8.73E−06	16843.86	100
	GABC	2.88E−06	5.51E−06	9062.85	100
	DE	6.24E−07	4.28E−06	1871.00	100
	PSO	3.11E−06	4.88E−06	2807.50	100
	LFSMOA	2.58E−06	7.55E−06	1345.29	100
	MLSMOA	2.42E−06	1.89E−06	49019.78	100
	PLSMOA	2.98E−06	4.95E−06	1418.78	100
OP_6	EASMO	1.35E−04	9.04E−04	21736.95	100
	SMOA	2.39E−04	7.57E−04	53552.63	100
	ABC	9.34E−02	1.45E−01	250151.98	0
	GABC	1.08E−02	1.25E−02	250045.38	1
	DE	4.77E−01	1.09E−01	40128.00	86
	PSO	1.58E−04	8.39E−04	49970.50	100
	LFSMOA	3.66E−05	9.80E−04	17656.19	100
	MLSMOA	6.84E−02	1.00E−01	196488.07	3
	PLSMOA	1.70E−04	8.68E−04	18280.78	100

(continued)

Table 2 (continued)

OP	Algorithm	SD	ME	AFEs	SR
OP_7	EASMO	1.60E−06	7.61E−06	4740.12	100
	SMOA	1.92E−06	7.16E−06	5936.04	100
	ABC	2.61E−06	6.66E−06	9004.50	100
	GABC	2.24E−06	6.93E−06	5587.00	100
	DE	1.31E−06	7.68E−06	10895.00	100
	PSO	1.50E−06	8.37E−06	15822.50	100
	LFSMOA	1.70E−06	7.79E−06	7461.81	100
	MLSMOA	3.02E−04	4.34E−04	200495.00	0
	PLSMOA	1.75E−06	7.61E−06	6141.89	100
OP_8	EASMO	2.77E−14	4.48E−14	6835.17	100
	SMOA	2.92E−14	4.63E−14	12299.03	100
	ABC	3.49E−05	1.11E−05	243317.32	14
	GABC	5.37E−09	5.40E−10	45017.81	99
	DE	1.29E−14	4.95E−14	5203.00	100
	PSO	2.70E−14	5.35E−14	9867.00	100
	LFSMOA	2.93E−14	5.36E−14	12942.91	100
	MLSMOA	3.44E−06	2.16E−06	200495.00	0
	PLSMOA	2.85E−14	5.22E−14	12459.74	100
OP_9	EASMO	5.55E−03	4.89E−01	955.35	100
	SMOA	4.98E−03	4.90E−01	1192.95	100
	ABC	5.54E−03	4.89E−01	1432.56	100
	GABC	5.22E−03	4.89E−01	789.01	100
	DE	2.09E−03	4.90E−01	2738.50	100
	PSO	5.54E−03	4.92E−01	4723.50	100
	LFSMOA	5.66E−03	4.91E−01	900.26	100
	MLSMOA	3.93E−03	4.85E−01	44207.90	100
	PLSMOA	5.48E−03	4.90E−01	1214.17	100
OP_{10}	EASMO	6.49E−06	8.62E−05	548.46	100
	SMOA	6.53E−06	8.72E−05	766.26	100
	ABC	6.46E−06	8.87E−05	1193.07	100
	GABC	6.70E−06	8.88E−05	624.50	100
	DE	2.18E−06	8.78E−05	1419.50	100
	PSO	6.39E−06	8.87E−05	1507.50	100
	LFSMOA	6.53E−06	8.90E−05	695.81	100
	MLSMOA	2.73E−08	7.71E−05	40198.00	100
	PLSMOA	7.48E−06	8.81E−05	714.54	100

(continued)

Table 2 (continued)

OP	Algorithm	SD	ME	AFEs	SR
OP_{11}	EASMO	5.39E−06	4.68E−06	2362.14	100
	SMOA	5.68E−06	4.95E−06	4313.43	100
	ABC	5.55E−06	4.86E−06	4633.30	100
	GABC	5.77E−06	5.14E−06	2360.72	100
	DE	2.69E−06	4.40E−06	9562.50	100
	PSO	2.11E−03	4.29E−04	103221.50	71
	LFSMOA	5.94E−06	5.26E−06	2300.62	100
	MLSMOA	2.53E−05	1.08E−05	113565.31	86
	PLSMOA	5.93E−06	5.34E−06	3865.29	100
OP_{12}	EASMO	2.41E−03	7.99E−03	143305.00	95
	SMOA	4.22E−03	9.34E−03	236620.15	81
	ABC	1.96E−03	8.06E−03	54974.74	92
	GABC	2.35E−03	7.90E−03	49857.62	99
	DE	2.43E−01	3.13E+00	248696.00	1
	PSO	2.83E−01	3.74E−01	212422.00	25
	LFSMOA	3.61E−03	8.77E−03	83137.93	90
	MLSMOA	2.36E−03	6.03E−03	103953.43	98
	PLSMOA	1.27E−03	8.64E−03	27219.35	100
OP_{13}	EASMO	1.69E−04	8.20E−04	38951.83	100
	SMOA	5.98E−04	1.09E−03	274879.26	74
	ABC	1.25E−04	8.75E−04	70489.94	100
	GABC	4.11E−04	1.01E−03	116653.35	78
	DE	1.99E−04	7.13E−04	68619.50	100
	PSO	2.85E−01	1.69E−01	172325.00	54
	LFSMOA	4.37E−04	9.42E−04	57499.78	98
	MLSMO	1.87E−01	2.63E−02	54630.67	97
	PLSMOA	8.81E−05	9.17E−04	32449.10	100
OP_{14}	EASMO	1.45E−01	3.55E−02	215547.40	77
	SMOA	3.54E−01	8.22E−02	249974.06	75
	ABC	7.78E−02	2.18E−01	250000.21	0
	GABC	2.33E−03	9.89E−03	250000.00	0
	DE	4.71E+00	1.16E+01	250050.00	0
	PSO	3.11E−02	2.67E−02	250050.00	0
	LFSMOA	1.09E+00	6.74E−01	188933.23	22
	MLSMOA	0.00E+00	9.10E+00	200495.00	0
	PLSMOA	1.75E+00	1.15E+00	180908.97	26

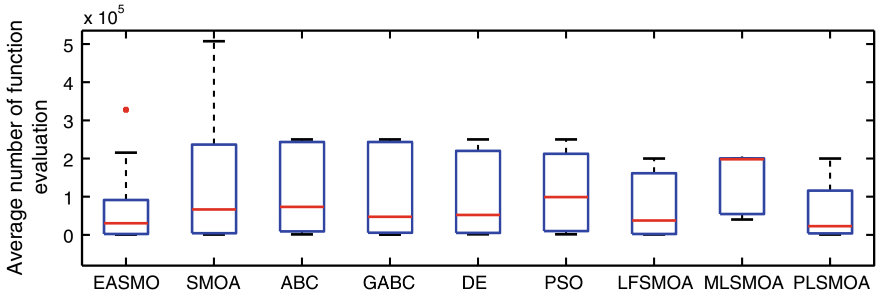

Fig. 4 Boxplot graph for average count of $function$ evaluation

To measure the efficiency of the EASMO, boxplot (bp) analysis [16] is carried out on the basis of AFEs as shown in Fig. 4. The results reveal that the interquartile range and medians of EASMO are comparably low to the considered algorithms.

Further, acceleration rate (AR) is calculated for the considered algorithms in comparsion with EASMO using Eq. 8. The calculated AR is shown in Table 3. It is clear from Table 3 that the EASMO is comparatively fast than the other considered algorithms. AR is calculated as:

$$AR = \frac{AFE_{ALGO}}{AFE_{EASMO}} \tag{8}$$

Table 3 Acceleration rate of EASMO compared to the SMOA, ABC, GABC, DE, PSO, LFSMOA, MLSMOA, and PLSMOA

Test problems	SMOA	ABC	GABC	DE	PSO	LFSMOA	MLSMOA	PLSMOA
fn_1	1.54	0.73	0.74	0.76	0.76	0.61	0.61	0.61
fn_2	1.21	1.10	0.83	0.92	1.36	2.33	2.89	2.01
fc_3	1.44	2.73	2.73	0.77	2.26	1.76	2.19	1.26
fn_4	1.75	2.00	0.98	4.83	5.34	1.87	4.39	1.53
fn_5	1.40	15.97	8.59	1.77	2.66	1.27	46.49	1.34
fn_6	2.46	11.50	11.50	1.84	2.29	0.81	9.03	0.84
fn_7	1.25	1.89	1.17	2.29	3.33	1.57	42.29	1.29
fn_8	1.79	35.59	6.58	0.76	1.44	1.89	29.33	1.82
fn_9	1.24	1.49	0.82	2.86	4.94	0.94	46.27	1.27
fn_{10}	1.39	2.17	1.13	2.58	2.74	1.26	73.29	1.30
fn_{11}	1.8	1.96	0.99	4.04	43.69	0.97	48.07	1.63
fn_{12}	1.65	0.38	0.34	1.73	1.48	0.58	0.72	0.18
fn_{13}	7.05	1.80	2.99	1.76	4.42	1.47	1.40	0.83
fn_{14}	1.15	1.15	1.15	1.16	1.16	0.87	0.93	0.83

Table 4 Collation based on AFE and the MVUR sum test at ('+' indicates EASMO is better, '−' shows EASMO is worse, and '=' shows that there is no noticeable distinction)

Function no	EASMO versus SMOA	EASMO versus ABC	EASMO versus GABC	EASMO versus DE	EASMO versus PSO	EASMO versus LFSMOA	EASMO versus MLSMOA	EASMO versus PLSMOA
OP_1	+	−	−	−	−	−	−	−
OP_2	+	+	−	−	+	+	+	+
OP_3	+	+	+	−	+	+	+	+
OP_4	+	+	−	+	+	+	+	+
OP_5	+	+	+	+	+	+	+	+
OP_6	+	+	+	+	+	−	+	−
OP_7	+	+	+	+	+	+	+	+
OP_8	+	−	+	−	+	+	+	+
OP_9	+	+	−	+	+	−	+	+
OP_{10}	+	+	+	+	+	+	+	+
OP_{11}	+	+	−	+	+	−	+	+
OP_{12}	+	−	−	+	+	−	−	−
OP_{13}	+	+	+	+	+	+	+	−
OP_{14}	+	+	+	+	+	−	−	−
Total number of '+' sign	14	11	8	10	13	8	11	9

Mann–Whitney U rank (MWUR) sum test based on mean function evaluations at $\alpha = 0.005$ significance level is also done as shown in Table 4. Table 4 depicts that for most of the standard problems, EASMO perform outstanding. While analyzing the results of Table 4, it can be stated that the EASMO performs well on all the 14 optimization problems as compared to SMOA, on 11 problems ($OP_2 - OP_7$, $OP_9 - OP_{11}$, $OP_{13} - OP_{14}$) as compared to ABC, on 8 problems (OP_3, $OP_5 - OP_8$, OP_{10}, $OP_{13} - OP_{14}$) as compared to GABC, on 10 problems ($OP_4 - OP_7$, $OP_9 - OP_{14}$) as compared to DE, on 13 problems ($OP_2 - OP_{14}$) as compared to PSO, on 8 problems ($OP_2 - OP_5$, $OP_7 - OP_8$, OP_{10}, OP_{13}) as compared to LFSMO, on 11 problems ($OP_2 - OP_{11}$, OP_{13}) as compared to MLSMO, and on 9 problems ($OP_2 - OP_5$, $OP_7 - OP_{11}$) as compared to PLSMO.

5 Conclusion

In this article, a new variant of SMOA is developed, namely exponential adaptive spider monkey optimization (EASMO) algorithm. In the Local Leader Stage, an exponential adaptive strategy is applied to balance the step size with respect to the iterations, which enhance the exploitation ability of the SMOA. Further, the EASMO