REVIEW ARTICLE

An Intensive and Comprehensive Overview of JAYA Algorithm, its Versions and Applications

Raed Abu Zitar[1] · Mohammed Azmi Al-Betar[2,3] · Mohammed A. Awadallah[2,4] · Iyad Abu Doush[5,6] · Khaled Assaleh[2]

Abstract

In this review paper, JAYA algorithm, which is a recent population-based algorithm is intensively overviewed. The JAYA algorithm combines the survival of the fittest principle from evolutionary algorithms as well as the global optimal solution attractions of Swarm Intelligence methods. Initially, the optimization model and convergence characteristics of JAYA algorithm are carefully analyzed. Thereafter, the proposed versions of JAYA algorithm have been surveyed such as modified, binary, hybridized, parallel, chaotic, multi-objective and others. The various applications tackled using relevant versions of JAYA algorithm are also discussed and summarized based on several problem domains. Furthermore, the open sources code of JAYA algorithm are identified to provide enrich resources for JAYA research communities. The critical analysis of JAYA algorithm reveals its advantages and limitations in dealing with optimization problems. Finally, the paper ends up with conclusion and possible future enhancements suggested to improve the performance of JAYA algorithm. The reader of this overview will determine the best domains and applications used by JAYA algorithm and can justify their JAYA-related contributions.

✉ Mohammed Azmi Al-Betar
m.albetar@ajman.ac.ae

Raed Abu Zitar
raed.zitar@sorbonne.ae

Mohammed A. Awadallah
ma.awadallah@alaqsa.edu.ps

Iyad Abu Doush
idoush@auk.edu.kw

Khaled Assaleh
k.assaleh@ajman.ac.ae

1 Sorbonne University Center of Artificial Intelligence, Sorbonne University-Abu Dhabi, Abu Dhabi, UAE

2 Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman, UAE

3 Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Irbid, Jordan

4 Department of Computer Science, Al-Aqsa University, P.O. Box 4051, Gaza, Palestine

5 Computing Department, American University of Kuwait, Salmiya, Kuwait

6 Computer Science Department, Yarmouk University, Irbid, Jordan

1 Introduction

In the Artificial Intelligence, the search for the optimal solution in the search space is one of the common issues in almost all sub-fields such as machine learning, deep learning, knowledge representation, reasoning, and perception. The traditional search techniques depend on the heuristic concepts which are problem-dependent trying to search for approximate solutions often with less focus on the quality [1]. Recently, Metaheuristic-based algorithms attract the attentions of artificial intelligence research communities due to its impressive capability in manipulating the search space of optimization problems [2]. A Metaheuristic-based algorithm is a general optimization framework that can be adapted for many optimization problems. It has an iterative evolution strategy that utilizes learning mechanisms exploiting accumulative knowledge as well as exploring problem search space. It is usually controlled by specific parameters used to find good enough approximation solution for optimization problem in hand [3].

Metaheuristic-based algorithms are conventionally categorized in several ways [4]: Nature-inspired vs. non-nature inspired, population-based vs. single point search, dynamic

vs. static objective function, single vs. various neighborhood structures, and memory usage vs. memory-less. Recently, the research community combined these algorithms in categories based on the natural inspirations of the Metaheuristic algorithm such as evolutionary algorithms (EAs), Local Search (LSA), Swarm intelligence , physical-based , chemical-based, and human-based algorithms [5, 6].

The traditional evolutionary algorithm is genetic algorithm (GA) which was proposed in 1970 by Holland and his students to emulate the biological concepts of survival-of-the-fittest [7] . The initial population is evolved using recombination, mutation, and natural selection. Other EAs includes: Genetic Algorithm (GA) [7], Differential evolution (DE) [8], Evolutionary programming (EP) [9], Evolution strategies (ES) [10], Genetic Programming (GP) [11], Probability-based incremental learning (PBIL) [12], and Biogeography-Based Optimizer (BBO) [13].

The metaheuristic local search-based algorithms are initiated with a single solution. Iteratively, that solution is adjusted following a specific trajectory in the search space by moving to the neighbouring solution until a local optima is achieved. These algorithms are very strong in exploitation. The most popular local search algorithms are β-Hill Climbing [14], Simulated Annealing[15], Tabu Search [16], Iterated Local Search [17], Variable Neighborhood Search [18], and Greedy Randomized Adaptive Search Procedure (GRASP) [19].

The Swarm Intelligence (SI) are the third category of metaheuristic-based algorithms. This type of algorithms is stemmed by the idea of how the Swarm of animals survive, collaborate, communicate in their social behavior and follow the swarm leader. The earliest SI are Particle swarm optimization (PSO) [20] and Ant colony optimization (ACO) [21]. The reminder of SI includes Artifical bee colony (ABC) [22], Firefly algorithm (FA) [23], Bat algorithm (BA) [24], Flower pollination algorithm (FPA) [25], Grey wolf optimizer (GWO) [26], Krill herd algorithm (KHA) [27], Moth-flame optimization algorithm (MFO) [28], and JAYA algorithm [29].

The physical-based category of metaheuristic algorithms emulate the physical phenomena happening in the universe. Examples of physical-based metaheuristic are Central force optimization (CFO) [30], Gravitational search algorithm (GSA) [31], big bang–big crunch (BBBC) [32], electromagnetic field optimization (EFO) [33], water evaporation optimization (WEO) [34], Multi-verse optimizer (MVO) [35], and thermal exchange optimization (TEO) [36]. The chemical-based metaheuristic category is inspired by the chemical process.These include Artificial chemical process (ACP) [37], chemical reaction optimization (CRO) [38], artificial chemical reaction optimization (ACRO) [39], gases Brownian motion optimization (GBMO) [40], and chemotherapy science algorithm (CSA) [41]. Human-based metaheuristic

algorithms simulate the intelligent behaviour in solving the problems. The most famous Harmony Search Algorithm (HSA) [42], Coronavirus herd immunity optimizer (CHIO) [43], firework algorithm (FWA) [44], teaching–learning-based algorithm (TLBA) [45], and football game-inspired algorithm (FGIA) [46] Fig. 1.

A recent metaheuristic-based algorithm called JAYA algorithm combines the features of EA in terms of survivability for the fittest principle as well as SI in which the Swarm normally follow the leader during the search for optimal solution. JAYA algorithm is proposed by Rao [29] in 2016 and gained a considerable interest from a wide variety of research communities due to its impressive characteristics: It is simple in concepts and easy-to-use. It has no derivative information in the initial search. It is a parameter-less algorithm. It is adaptable, flexible, and sound-and-complete. Therefore, JAYA algorithm has been tremendously utilized for a plethora of optimization problems in different domains such as Optimal Power Flow [47, 48], parameters extraction of solar cells [49], knapsack problems [50], virtual machine placement [51], job shop scheduling [52, 53], permutation flow-shop scheduling problem [54], reliability–redundancy allocation problems [55], team formation problem [56], truss structures [57], facial emotion recognition [58], feature selection [59], Plate-fin heat exchanger [60], estimating Li-ion battery model parameters [61], etc.

Due to the complexity nature of some optimization problems and their rugged features in their search space, several researchers have modified the JAYA algorithm to improve their convergence behaviours. Some other researchers hybridize JAYA Algorithm with other powerful components borrowed from other optimization algorithms. Therefore, several variants of JAYA algorithm have been proposed such as binary JAYA [62]. self-adaptive JAYA algorithm [63, 64], elitism-based JAYA [65], elitism-based self-adaptive multi-population JAYA algorithm [66, 67], chaotic JAYA algorithm [64, 68], Neural network JAYA algorithm [69, 70], hybridization with evolutionary algorithms [71, 72], hybridization with swarm intelligence algorithms [73–76], hybridization with physical-based algorithms [55], and hybridization with other components [77, 78].

In this review paper, an intensive but not exhaustive overview of the JAYA algorithms from different sides is provided: initially, the theoretical background are discussed and analyzed. The modified, hybridized and adapted versions of JAYA algorithm are then overviewed. The applications tackled by JAYA algorithms are summarized based on several categories. The reader of this overview will determine the domains and applications that used JAYA algorithm and how to justify their JAYA-related contributions. Furthermore, the open sources code of JAYA algorithm are identified to provide enrich resources for JAYA research communities. Accordingly, this paper will provide critiques and critical
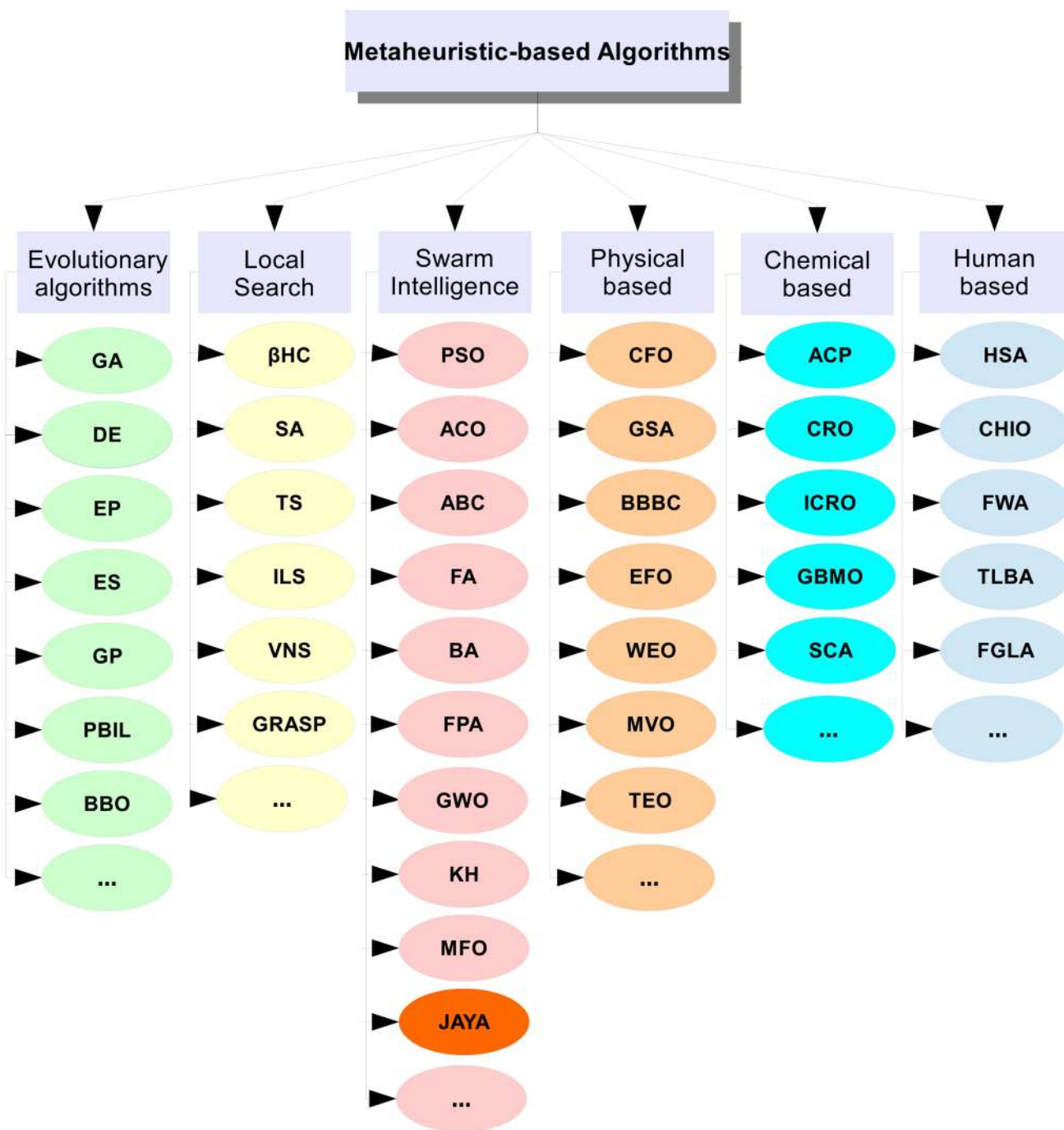
Metaheuristic-based Algorithms

Evolutionary algorithms	Local Search	Swarm Intelligence	Physical based	Chemical based	Human based
GA	βHC	PSO	CFO	ACP	HSA
DE	SA	ACO	GSA	CRO	CHIO
EP	TS	ABC	BBBC	ICRO	FWA
ES	ILS	FA	EFO	GBMO	TLBA
GP	VNS	BA	WEO	SCA	FGLA
PBIL	GRASP	FPA	MVO	...	...
BBO	...	GWO	TEO		
...		KH	...		
		MFO			
		JAYA			
		...			

Fig. 1 The categories of Metaheuristic-based algorithms

analyses about the main pros and cons of JAYA algorithm. Finally, the conclusion of the existing JAYA algorithm contributions and the possible future directions to improve the performance and the applicability of using JAYA algorithm is provided.

The rest parts of this overview paper is arranged as follows: The timeline and the growth of the JAYA algorithm developments are skimmed in Sect.2. The theoretical aspects of JAYA algorithm are provided in Sect.3. The latest developments in terms of adaptation, modifications, hybridization of JAYA algorithm variants are given in Sect. 4. The JAYA algorithm has been widely adapted for various real-world optimization applications which are extensively summarized in Sect. 5.The open source software prepared for JAYA algorithm is illustrated in Sect.6. Critical analysis about the convergence behaviour of JAYA algorithm is provided in

Fig. 2 The number of JAYA algorithm publications per publisher extracted from Google Scholar at 17-8-2020
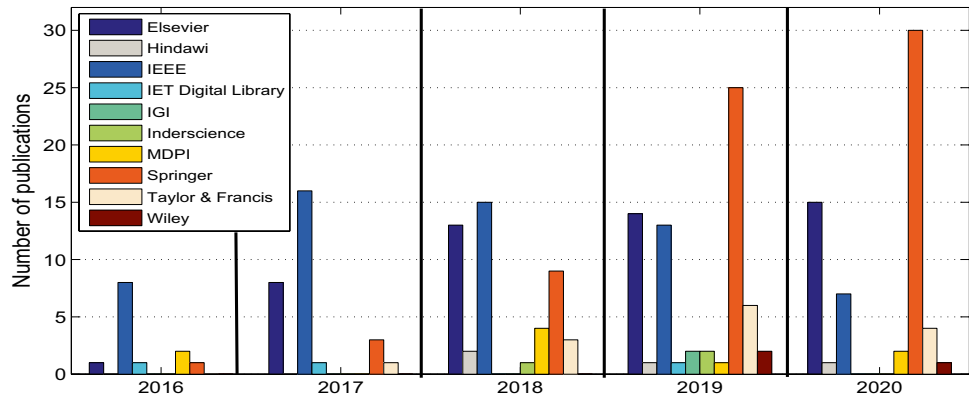
Fig. 3 The number of JAYA-based publications published by each subject (Source Scopus: 17-8-2020)
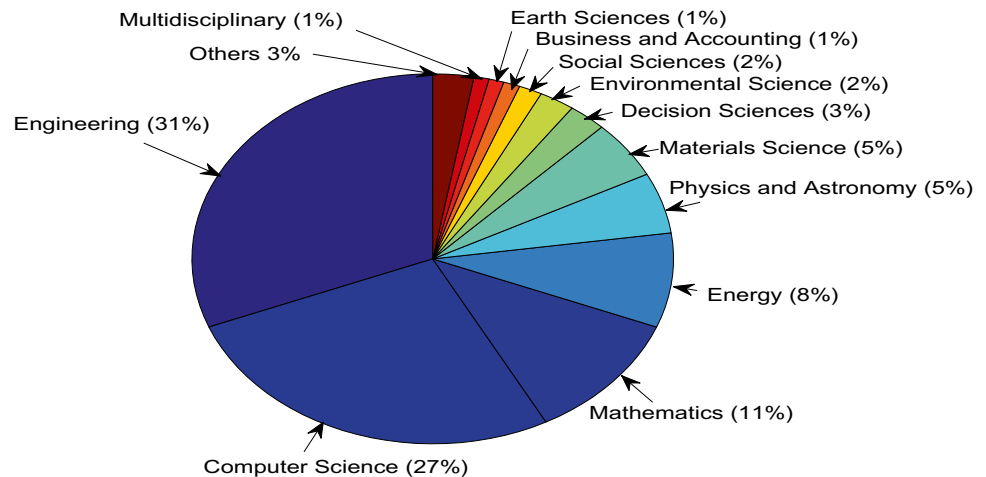
Sect. 7. Finally, the overview paper is concluded and some future points are highlighted in Sect. 8.

2 The Growth of JAYA Glgorithm in the Literature

The growth and progression of JAYA algorithm is presented in this section with different perspectives. The JAYA algorithm is initially established in 2016. Since that time, the interest of the JAYA algorithm has rapidly increased. It has been published in well reputed journals, conferences, book chapters with high-prestige publishers such as ScienceDirect, Springer-Link, IEEEExplorer, IET Digital Library. IGI, Inderscience, MDPI, Hindawi, Taylor & Francis, Wiley, and many others as shown in Fig. 2.

The growth of the JAYA algorithm is also analyzed based on the topic against the number of publications per topic. Apparently, the highest number of publications is published in the Engineering topics which is equal to 338. The computer science topic has also gained tremendous publications with a number equal to 292. In the mathematics topics, JAYA-based algorithms are used 122 times to solve

mathematical problems. Other topics used JAYA-based algorithms with large extension as shown in Fig. 3.

Also in this JAYA overview paper, the top 15 authors who published a number of documents using JAYA-based algorithms are given in Fig. 4. As can be noticed, the founder of JAYA algorithm has the most interest in solving optimization problems with 30 JAYA-based articles recorded in Scopus database.

The National Institutes of Technology in India has the most interest in JAYA-based algorithms with 47 published articles in different domains. The second-highest interest institutions which published 14 articles in JAYA-based algorithms are the University of Science and Technology Beijing at China and the Ton-Duc-Thang University at Vietnam . Other interested institutions are bar-charted in Fig. 5.

In another perspective, the country of India has published JAYA-based algorithms and its variants more than 140 times while country of china utilized the JAYA-based algorithms and its versions more than 35 times. In Fig. 6, the usage of JAYA-based algorithms are bar-charted for 20 countries sorted in descending order based on the number of articles published. This data is extracted using Scopus database at 17-8-2020.
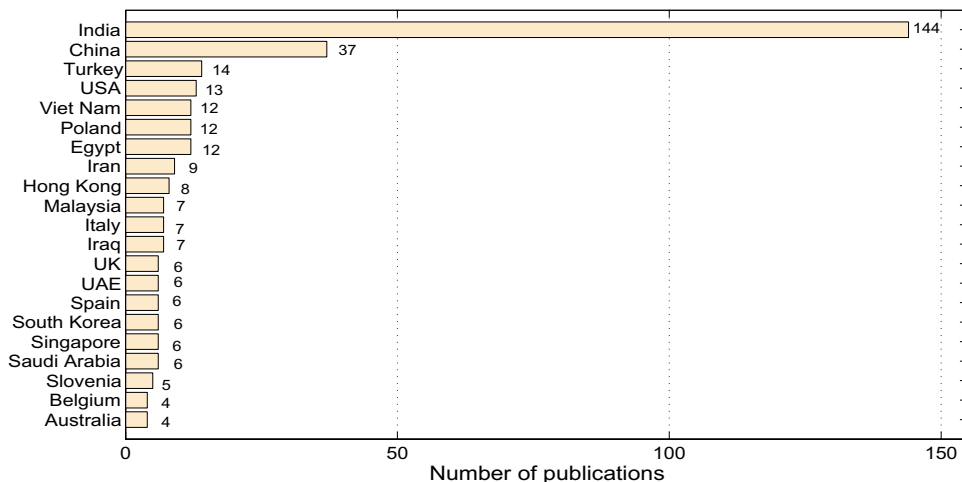
Fig. 4 The number of JAYA-based publications published The top 15 authors (Source Scopus)
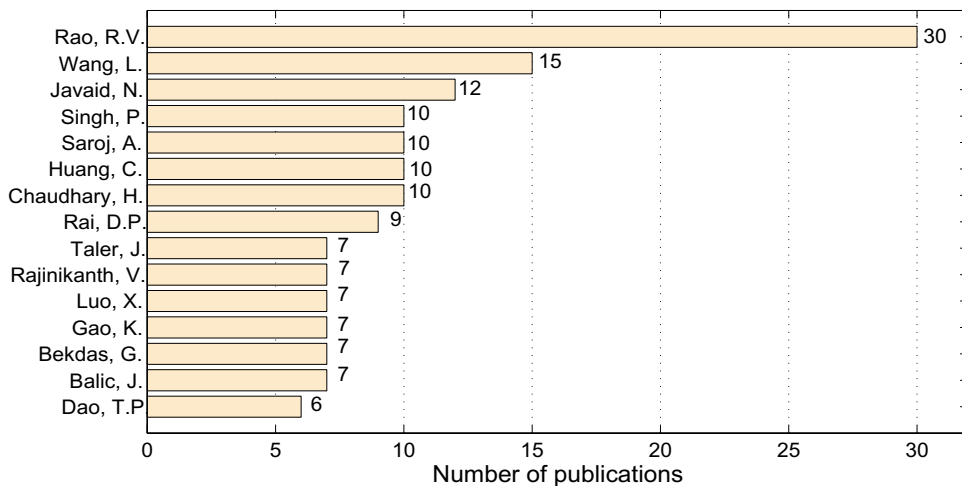
Fig. 5 The number of JAYA-based publications published in the top 10 afflictions (Source Scopus)
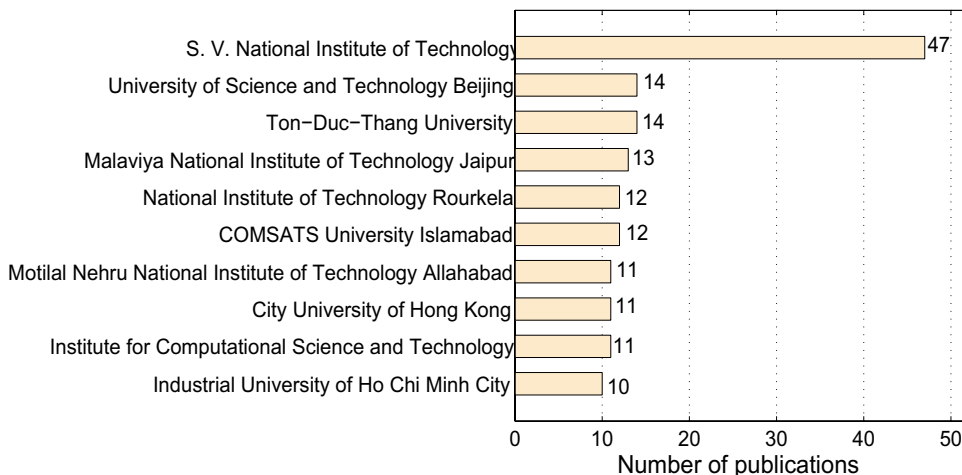
Fig. 6 The number of JAYA-based publications of each country (Source Scopus)

In order to provide a comprehensive overview for the growth of JAYA-based articles, the progression of the JAYA algorithms based on the number of articles published per year using most two popular databases, that are Scopus and Google Scholar, are bar-charted in Fig. 7a and b, consecutively. Apparently year after year, the interest growth of JAYA-based algorithm has gradually increased. This data is extracted using Scopus and Google Scholar database at 17-8-2020.
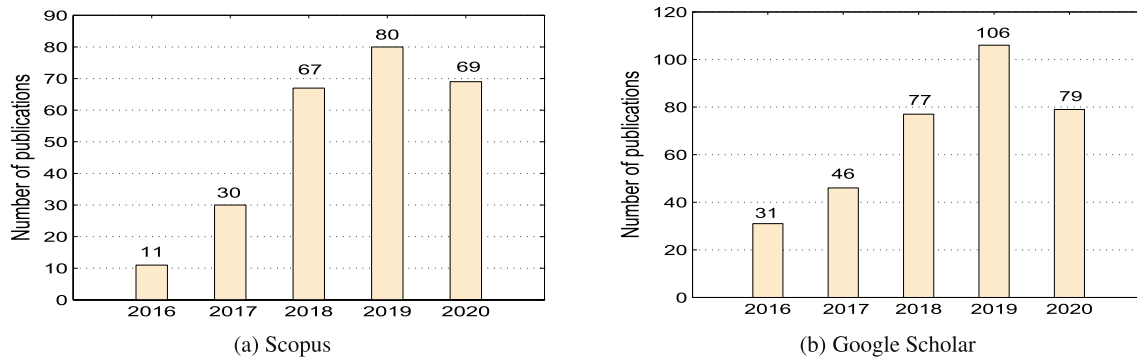
(a) Scopus

(b) Google Scholar

Fig. 7 The number of JAYA-based publications per year

3 Basic Concepts of JAYA Algorithm

JAYA algorithm is a recent population-based metaheuristic algorithm proposed by Rao [29]. This section presents and analyzes JAYA algorithm in different optimization perspectives. Initially the inspiration of JAYA algorithm is provided. Thereafter, the procedural steps of JAYA algorithm are given. Finally, the exploration and exploitation capabilities of such algorithm are discussed.

3.1 Inspiration of JAYA Algorithm

The JAYA algorithm was initially established to tackle constrained and unconstrained optimization functions. The JAYA term is Sanskrit in origin, it means "victory". This algorithm is a population-based metaheuristic combining the features of evolutionary algorithms and Swarm-based Intelligence. It is inspired by the natural behaviour of the "survival of the fittest" principle. This means that solutions in the JAYA population are being attracted toward the global best solutions and at the same time neglecting the worst solutions. In other words, the search process of the JAYA algorithm tries to get closer to success by reaching the global best solutions, and attempts to escape from failure by running out from the worst solutions. JAYA algorithm has several advantages over other population-based algorithms such as being easy to implement with no algorithm-specific parameters dependence (i.e., the population size, and maximum number of iterations) [57, 79].

3.2 Procedural Steps of JAYA Algorithm

In this overview paper, a new presentation based on the six procedural steps are presented for JAYA algorithm. These procedural steps can help the researchers of the optimization algorithms to use this algorithm easily. The procedural steps of JAYA algorithm are shown in flowchart of Fig. 8. Furthermore, the pseudo-code of JAYA algorithm is provided

in Algorithm 1. The procedure of JAYA algorithm is thoroughly discussed in following steps:

Step:1 Initialize the parameters of both JAYA algorithm and optimization problem. Initially the parameters of JAYA Algorithm are set in the initial stage of run. Interestingly, the JAYA algorithm has no control parameters. It has only two algorithmic parameters which are population size N and iteration numbers T. Normally the constrained problem modeled in the optimization context is as follows:

$$\min f(\boldsymbol{x})$$
$$\text{S.t.}$$
$$g_j(\boldsymbol{x}) = c_j \quad \forall j = (1, 2, \ldots, n)$$
$$h_k(\boldsymbol{x}) \leq d_k \quad \forall k = (1, 2, \ldots, m)$$

where $f(\boldsymbol{x})$ is the objective function used to calculate the fitness value of the solution $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)$ where x_i is a decision variable assigned by a value in the lower and upper bound range such that $x_i \in [X_i^{min}, X_i^{max}]$. g_j is the $j^t h$ equality constraints and h_k is the k inequality constraints. The problem variables, dimensions and related data are normally extracted for a benchmark dataset.

Step : 2 Constructing the initial population for JAYA. The initial solutions (or population) of JAYA algorithm are constructed and retained in the JAYA Memory (JM). Note that the *JM* is an augmented matrix of size $N \times D$ as shown in Eq. 2 where N is the number of solutions and D is the solution dimension. Conventionally, solution is randomly constructed: $JM_{i,j} = X_j^{min} + (X_j^{min} - X_j^{max}) \times rnd$, $\forall i \in (1, 2, \cdots, N) \bigwedge \forall j \in (1, 2, \cdots, D)$. *rnd* is a uniform function generates a random values between 0 and 1.

$$\mathbf{JM} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_D^1 \\ x_1^2 & x_2^2 & \cdots & x_D^2 \\ \vdots & \vdots & \cdots & \vdots \\ x_1^N & x_2^N & \cdots & x_D^N \end{bmatrix} \begin{bmatrix} f(\boldsymbol{x}^1) \\ f(\boldsymbol{x}^2) \\ \vdots \\ f(\boldsymbol{x}^N) \end{bmatrix} \quad (2)$$

Fig. 8 The flowchart of the JAYA algorithm
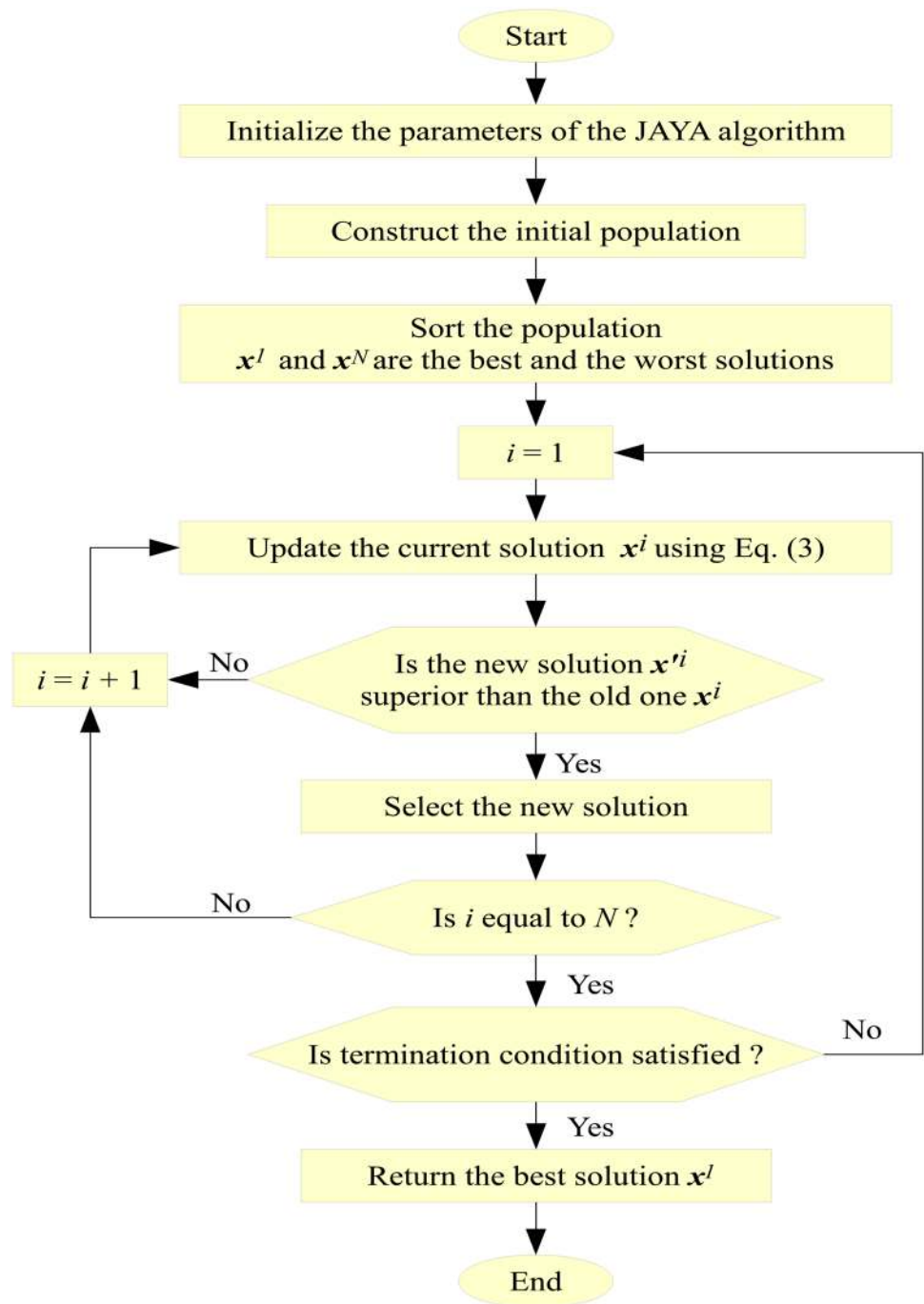
The objective function $f(x^i)$ for each solution is also calculated and the *JM* solutions are sorted in a ascending order based on their objective function values. Therefore, the best solution is x^1 while the worst solution is x^N.

Step: 3 JAYA Evolution process. Iteration by iteration, The decision variables of all solutions in the *JM* undergoes changes using JAYA operator formulated in Eq.3.

$$x_j'^i = x_j^i + r_1 \times (x_j^1 - |x_j^i|) - r_2 \times (x_j^N - |x_j^i|) \quad (3)$$

Note that $x_j'^i$ is the new updated solution; x_j^i is the current solution. $x_j'^i$ is the modified value of the decision value x_j^i. r_1 and r_2 are two uniform functions generates a random value in the range of [0,1]. These generated random numbers are used to achieve the right balance between the exploration and exploitation processes. Note that x_j^1 is the decision variable j in the best solution while x_j^N is the decision variable j in the worst solution. The distance between the decision variables of the best solution and

the current one and the distance between the decision variable of the wort solution and current one determine the diversity control of the JAYA algorithm. Closer distance means higher exploitation and higher distance means higher exploration.

*Step*4: Update JM . The JM solutions at every iteration will be updated. The objective function value of the new solution $f(x'^i)$ is calculated. The current solution x^i will be replaced by the new solution x'^i, if $f(x'^i) \leq f(x^i)$. This process will be repeated as many as N.

Step: 5 Stop rule. The JAYA algorithm repeats Step 3 and Step 4 until the stopping rule which is sometimes the maximum number of iterations T is reached.

Algorithm 1 The pseudo-code of JAYA algorithm

1: Initialize the parameters of both JAYA algorithm and optimization problem (N, T, etc.).
2: Initialize a population of N solutions randomly.
3: Calculate $f(X_i)$ $\forall i = 1, 2, \ldots, N$
4: Sort the population: (x^1 and x^N are the best and the worst solutions respectively).
5: t=1
6: **while** $(t \leq T)$ **do**
7:     **for** $i = 1, \cdots, N$ **do**
8:         **for** $j = 1, \cdots, D$ **do**
9:             Set $r_1 \in$[0,1]
10:            Set $r_2 \in$[0,1]
11:            $x'^i_j = x^i_j + r_1 \times (x^1_j - |x^i_j|) - r_2 \times (x^N_j - |x^i_j|)$
12:        **end for**
13:        **if** $f(x'^i) \leq f(x^i)$ **then**
14:            $x^i = x'^i$ {Update process}
15:        **end if**
16:    **end for**
17:    $t = t + 1$
18: **end while**

3.3 Exploration vs. Exploitation in JAYA Algorithm

It is worth to analyze the exploration and exploitation behavior of JAYA algorithm. Recall that the exploration and exploration are a common process in any metaheuristic-based algorithm. The success of any metaheuristic algorithm depends on its capability of achieving the right balance between wide range exploration of the search space regions and close near-by exploitation of the region to which explore. These two terms are contradictory in which the explorations refer to the ability of the algorithm to navigate not-visited search space regions while the exploitation refers to the ability of the algorithm to deeply search in the search space region to which it navigate. Normally, the exploration related to the random search while the exploitation related to the accumulative search.

In JAYA algorithm, there is only one operator called JAYA which is mathematically formulated in Eq. 3. The exploration and exploitation process are achieved through the distance between the current value of the decision variables and the corresponding variable in the best and worst solution. As far as the distance is large, the exploration is getting higher. Normally, because the random nature of the initial population is high, the distance will be also higher, thus the exploitation is lower. In the final stage of search, the distance is very minimal due to identical population situation. Thus the search is more concentrate on exploitation rather than exploration. In order to show a clear visualization plot to draw such concepts, Fig. 9 is an example of drawing the distance of the average value of the decision variables for 5 solutions in *JM* including best and worst. Each line represent the distance of one solution which is fluctuated in the initial search and all reach steady-state in the final course of runs. Note that the figure uses the Sphere mathematical function with $D = 30$.

The convergence behaviour is also studied and analyzed using the chart of the Sphere function ($D = 5$) shown in Fig 10. The trends in the plot represent the convergence behaviour (i.e., objective function value) of one solution over 60 iterations. As can be noticed, the fitness value of each solution begins with extremely high value and sharply reduced to reach equilibrium state where the optimal solution is located.

Fig. 9 The Exploration and Exploitation of the JAYA algorithm
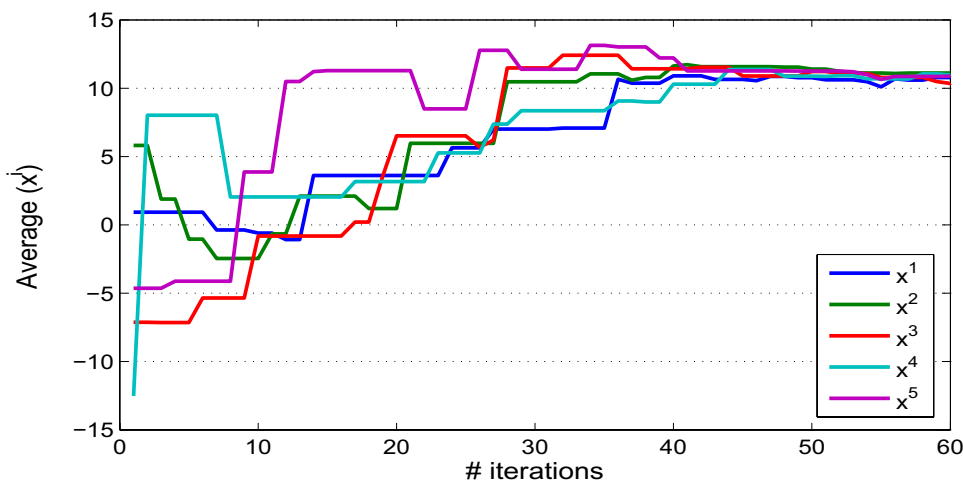
Fig. 10 The convergence of the JAYA algorithm
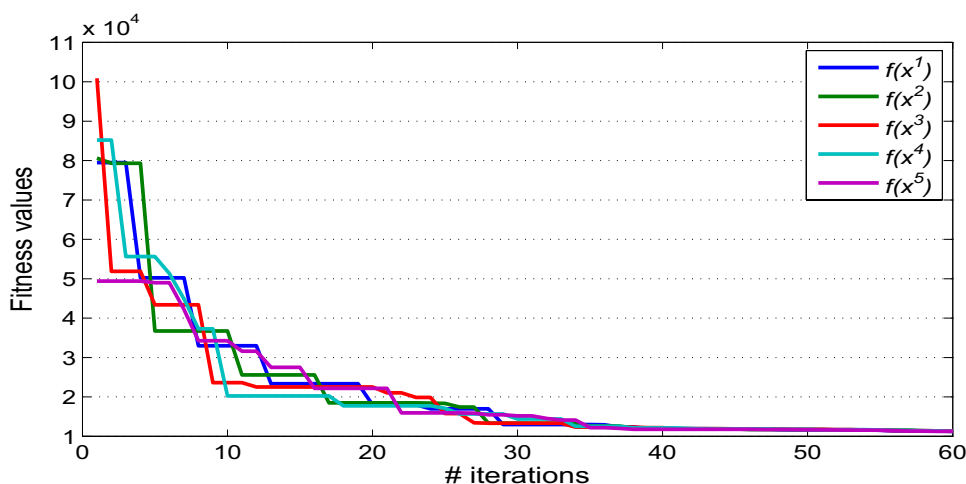
Fig. 11 The Hamming Distance of the JAYA algorithm
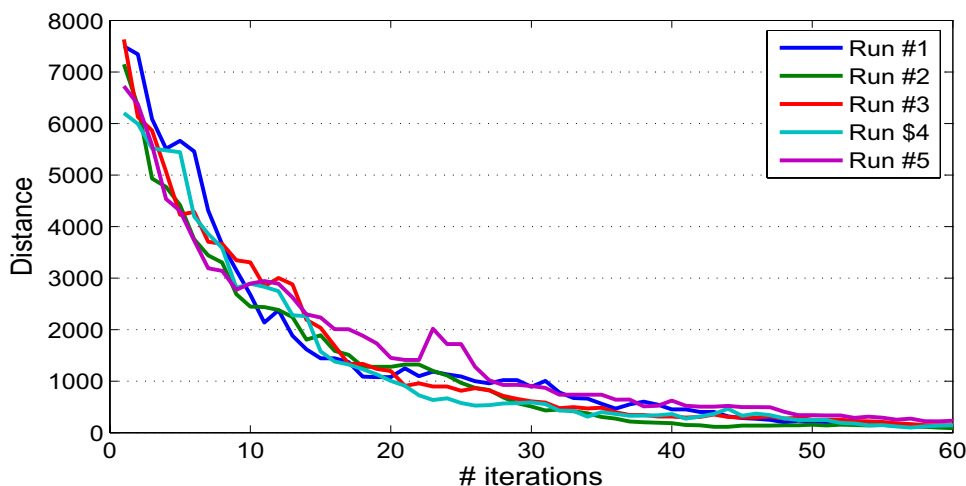
The hamming distance is normally calculated to show the distance between the JM solutions to measure the power of JAYA algorithm in controlling the diversity through the search process [80]. The Sphere function with $D = 30$ is again used in JAYA algorithm with $N = 5$. The hamming distance is calculated over 60 iterations using 5 runs. Each run is a trend in the chart and the behaviour of hamming distance for all run are converged to the stagnation point in the final stage of search. This show that how the JAYA

Table 1 Optimal results for Hartman test function using JAYA algorithm

#	Initial population				Population after one iteration				Population after 10 iterations			
	x_1	x_2	x_3	$f(x)$	x_1	x_2	x_3	$f(x)$	x_1	x_2	x_3	$f(x)$
1	0.0878	0.0525	0.6876	−0.2965	0.3611	0.2095	0.8664	−1.2607	0.0496	0.5724	0.8302	−3.8014
2	0.8828	0.1017	0.6692	−0.3757	0.6904	0.0869	0.8196	−0.5361	0.2279	0.5303	0.8231	−3.7588
3	0.8194	0.5268	0.3575	−0.147	0.719	0.5221	0.4476	−0.306	0.3139	0.5789	0.8285	−3.7606
4	0.4542	0.4939	0.7534	**−2.9553**	0.4542	0.4939	0.7534	**−2.9553**	0.1556	0.5581	0.8275	**−3.8034**
5	0.7246	0.6125	0.3515	−0.1927	0.6633	0.5802	0.5685	−0.9269	0.1282	0.5823	0.8137	−3.7011

#	Population after 20 iterations				Population after 30 iterations				Population after 39 iterations			
	x_1	x_2	x_3	$f(x)$	x_1	x_2	x_3	$f(x)$	x_1	x_2	x_3	$f(x)$
1	0.1816	0.5588	0.8455	−3.8549	0.1553	0.5803	0.8515	−3.8402	0.119	0.5547	0.8506	−3.8624
2	0.1652	0.5525	0.8445	−3.8549	0.1605	0.5434	0.8627	−3.8455	0.1007	0.5517	0.8531	−3.8621
3	0.1616	0.5486	0.8458	−3.8557	0.1629	0.5592	0.8511	**−3.8607**	0.1187	0.5554	0.8509	−3.8625
4	0.1501	0.5538	0.8447	**−3.8562**	0.1501	0.5538	0.8447	−3.8562	0.1186	0.5556	0.8514	**−3.8626**
5	0.159	0.5486	0.8436	−3.8528	0.0389	0.5495	0.8489	−3.8566	0.1052	0.5538	0.8518	**−3.8626**

Bold font refers to the best results

algorithm can control the diversity during the search (see Fig. 11).

3.4 Illustrative Example of JAYA Algorithm

In order to provide a clear picture of how the convergence behaviour of JAYA algorithm, HARTMANN 3-dimensional function is used. This function is as follows:

$$\min \quad f(\boldsymbol{x}) = - \sum_{i=1}^{4} c_i \exp\left(- \sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2 \right)$$

where $x_i \in [1, 3]$ and the global optima is −3.86. As shown in Table 1, the population size is 5. The five solutions together with their objective function values are recorded in the 1st, 2nd, 10th, 20th, 30th, 39th iterations. As can be noticed, the distance between the decision variables of the initial population is large. Iteration by iteration, this distance is gradually reduced until the global optima is reached at iteration #39. This proves that the JAYA Algorithm is tending to be explorer in the initial search and tends to be exploiter at the final stage of run.

4 Recent Variants of JAYA Algorithm

JAYA is a flexible algorithm that is simple in nature but very efficient in action [81]. It is a population-based search algorithm that is originally established for continuous search spaces. It has been modified to be workable as discrete or focuses on local search in addition to the global search. Many variants and hybridization of JAYA algorithm have been introduced to upgrade its efficiency and effectiveness.

Some applications require new variants of JAYA algorithm to be used. In the following sections, several variants of JAYA are discussed for different problems.

4.1 Modified Versions of JAYA Algorithm

The modified versions of JAYA are introduced. The summarized variants concerns with modifying some of the parameters or operators of JAYA. Those modifications do not constitute major change in the JAYA algorithm mechanism rather that minor changes. This is to add more ability to JAYA to achieve better balance between exploration and exploitation for the complex search spaces of different applications. The modified variants of JAYA include binary JAYA algorithm, Discrete JAYA algorithm, Chaotic JAYA algorithm, Adaptive JAYA algorithm, multi-population JAYA algorithm, parallel JAYA algorithm, Fuzzy-based JAYA algorithm, and others.

4.1.1 Binary JAYA Algorithm

As aforementioned, the JAYA algorithm is produced for continuous search space. The JAYA algorithm is modified to work on optimization problems with binary search space. There are several examples in literature that presents binary JAYA in one way or another. The binary JAYA could be blended with some other transfer functions to achieve the required results based on the applications being tackled [59, 62, 82–84].

In [62], authors manged to produce a genuine versions of binary JAYA algorithm called 'JAYAX'. The X-OR operator is used to discretize the decision variables into binary. The proposed method is evaluated against 15 benchmark

functions established in IEEE-CEC2015. The results were better than those produced by other comparative methods. In another study [83], a new binary version of JAYA algorithm is proposed for feature selection problems. Naive Bayes and Support Vector Machine are used to evaluate the resulting feature subsets using different text corpus data sets. Their proposed method proved its efficiency against other existing works.

For feature selection, a new variant of binary JAYA algorithm is proposed [59]. The JAYA algorithm is modified by using S-shape transfer method to normalize the continuous values into elements with [0,1]. The proposed method is evaluated using 22 data sets extracted from UCI data repository. The binary JAYA algorithm produced very fruitful results. A new version for feature selection problem to predict transient stability assessment of power systems is also tackled using binary JAYA algorithm [82]. The phasor measurement units (PMUs) data is used to evaluate their proposed 'binJAYA' algorithm and yield a very promising results.

In [84], commitment to economic unit is a large-scale mix-integer optimization problem that calls for strong and effective instrumentation. At the other hand, the environmental effect of power dependent generation is rapidly gaining attention because of the global warming phenomenon and its urgency. This calls for the production of renewable resources. In this article, the twofold economic and commitment to the emissions unit are made into a single objective issue. To solve this, a new binary JAYA optimization is proposed and incorporated with the process of lambda iteration. The proposed binary JAYA method is inspired by the JAYA evolution and generates binary bits from a v-shape transfer function. Numerical study demonstrates the significant improvement of the binary JAYA in regarding the convergence speed for solving unit commitment problem.

4.1.2 Discrete JAYA Algorithm

The JAYA algorithm is also modified to manipulate the discrete search space of different constrained and unconstrained real world applications with discrete decision variables [54, 85, 86].

The flexible job shop re-scheduling problem is tackled using a discrete version of JAYA algorithm and called DJaya. Two heuristics are used to build efficient initial solutions [86]. Thereafter, five local search-based operators are used to improve the convergence behaviour of DJaya. For evaluation process, seven real-world experimented cases with various scales from pump re-manufacturing are used and the results are very competitive when compared with some state-of-the-art algorithms.

Another paper with modified JAYA algorithm for dealing with discrete search space is proposed by Degertek et. al. [85]. It has been modified to solve truss structures under stress and displacement constraints with huge search space including many local minima. Their proposed algorithm is denoted as 'DAJA'. Instead of using the best and worst solution in JAYA algorithm, each candidate solution is updated using the descent directions in its neighborhood for evolution process. Their proposed method is measured using a collection of seven benchmark problems which showed the superiority of DAJA over other state-of-the-art metaheuristic algorithms.

In [54], another discrete version of JAYA algorithm is adapted for permutation flow-shop scheduling problem. The aim of this study was to minimize the make-span. The procedural steps of the solution method include: 1) a random priority is allocated to each job in a permutation sequence, and 2) the job priority vector is converted to job permutation vector using the Largest Order Value (LOV) concept. The performance of this JAYA algorithm is compared with other five algorithms using two benchmark datasets. The experimental results show that the JAYA algorithm is an efficient algorithm for solving permutation flow-shop scheduling problems.

Singh, Prem, and Himanshu Chaudhary [87] introduced a discreate version of JAYA algorithm for optimum two-plane balancing of the threshing drum. In their algorithm, the continuous solutions are converted to discrete solutions using discrete constraints of predefined values of design variables. The efficiency of their algorithm is tested using a model of the threshing drum supported on bearings. Simulation results showed that this algorithm is more efficient than GA and PSO in terms of solution quality and computational time.

A modified version of JAYA algorithm is presented by Singh, Prem, and Himanshu Chaudhary [88] for solving mixed-variable optimization problems. In their algorithm, the continuous variables remain in the continuous domains, while continues domains of discrete and integer variables are converted into discrete and integer domains. This is done by applying bound constraint of the middle point of corresponding two consecutive values of discrete and integer variables. The performance of the modified JAYA algorithm is evaluated using five data sets of mixed-variable optimization problems. Simulation results showed that the modified JAYA algorithm is more efficient and easier to use than other comparative methods.

4.1.3 Chaotic JAYA Algorithm

In chaotic map theory, the chaos is a random-orientation, deterministic strategy exists in dynamic systems with nonlinear in nature. This dynamic system is non-converging, non-period and bounded. The chaos is normally utilized as the source of randomness which works in as a simple deterministic dynamical system. Chaos use different chaotic

maps each with special mathematical equation to possess the sequence of elements regularly. By means of this, the chaos will help optimization methods to explore the problem search space more widely and dynamically. The theory of chaos is integrated in JAYA algorithm to improve its diversity control [64, 68, 89–91].

The authors in [91] introduces a novel chaotic JAYA algorithm for unconstrained numerical optimization, called C-Jaya algorithm. In C-Jaya, three new mutation strategies integrated with the original JAYA algorithm in order to enhance to the global and local search capabilities, and thus solving the problem of premature convergence. Experimentally, the performance of the C-Jaya is tested using sixteen benchmark functions with various dimensions. In comparison with the original JAYA algorithm, it can be seen that the performance of C-Jaya algorithm has faster global convergence, better solution quality, and most robust than the original JAYA algorithm. In comparison with other algorithms, the results show that the performance of the C-Jaya algorithm is superior than others in the convergence speed and solution quality in accordance to almost all the test functions.

Another chaotic JAYA algorithm is introduced by Yu et al. [89] for solving the economic load dispatch problems. Their algorithm is called multi-population based chaotic JAYA algorithm (MP-CJAYA). In, MP-CJAYA, the multi-population method and chaotic optimization algorithm are integrated with the original version of JAYA algorithm to guarantee the best solution of the problem. The population is divided into a set of sub-populations with the same size in order to enhance the exploration and exploitation abilities. The chaotic concept is adopted on each sub-population during the search to keep searching for the global optima. Five case studies are used for evaluation purposes. The experimental results show that the performance of the MP-CJAYA performs better than the original version of JAYA algorithm, as well as the chaotic version of JAYA algorithm in all case studies. In comparison with other algorithms in the literature, the performance of the MP-CJAYA algorithm outperforms the other algorithms in all case studies.

In [68], the authors presented a logistic chaotic JAYA algorithm (LCJAYA) for parameters identification of photovoltaic cell and module models. In LCJAYA algorithm, a logistic chaotic map strategy and a chaotic mutation strategy are introduced into the solution updating phase of JAYA algorithm. The aim of this modifications is to enhance the population diversity, as well as empower the balance of the exploration and exploitation abilities of the JAYA algorithm. The performance of the LCJAYA algorithm evaluated using two datasets with different PV models. The simulation results show that the performance of the LCJAYA algorithm outperforms the original version of the JAYA algorithm, and

other four algorithms from the literature in terms optimization accuracy and reliability.

In another study by Ravipudi et al. [64], they applied three versions of JAYA algorithm for to synthesize linear antenna arrays. These versions are the original version of JAYA algorithm. Self-adaptive Jaya algorithm (SJaya) and Chaotic-Jaya (CJaya) algorithm. In, SJaya, the size of the population is adopted automatically during the search in order to manipulate diversity. In CJaya, the Chaotic rule is integrated with the JAYA algorithm in order to improve the convergence speed and to solve the problem of getting stuck in local optima. The three JAYA-based algorithms are tested using two datasets with different formulations. The simulation results show that the performance of the three JAYA-based algorithms perform better than the other optimization algorithms.

4.1.4 Adaptive JAYA Algorithm

The adaptive parameter concept in the optimization context refers to as the automatically tuning parameters during the search process by the algorithm. In JAYA algorithm, there is no control parameters while it has two algorithmic parameters which are population size N and maximum number of iterations. Therefore, the adaptive versions of JAYA algorithm are proposed to adapt the population size during the search.

A self-adaptive JAYA algorithm is presented in [92]. In this algorithm, the size of the population is automatically adapted in order to enhance the convergence rate of the JAYA algorithm. This algorithm is used for optimal design of cooling tower from economic facets. Six different cases of mechanical draft cooling tower are used to evaluate this algorithm. The performance of the self-adaptive JAYA algorithm is compared with the original version of JAYA and three other comparative methods. The results illustrate that the self-adaptive JAYA algorithm outperforms others in terms of the solution quality, convergence rate, and CPU time. The same authors applied the same self-adaptive JAYA algorithm for optimal design of thermal devices [63]. The simulated results demonstrated that the self-adaptive JAYA algorithm is efficient for solving such problems by getting results better than six of the other comparative methods. In another study, the same self-adaptive JAYA algorithm is used for the design and optimization of plate-fin heat exchanger with superior results[60].

In [64], the same self-adaptive JAYA algorithm is employed to synthesize linear antenna arrays. This algorithm is tested using five case studies of syntheses of linear antenna arrays. The results of the self-adaptive JAYA algorithm are compared with the original version of JAYA algorithm, chaotic JAYA algorithm, as well as eight other comparative methods. The simulation results show the

performance of the self-adaptive JAYA algorithm is similar to the performance of chaotic JAYA algorithm in almost all cases. Furthermore, the self-adaptive JAYA algorithm obtained superior results in comparison with other comparative methods in terms of the solution quality and convergence speed.

4.1.5 Fuzzy-Based JAYA Algorithm

Many variants of Jaya are integrated with fuzzy controllers for the purpose of optimizing different parameters of those fuzzy controllers. That resulted in better performance of those controllers as demonstrated in works such as [93]. For example, in [76], one of the most popular and flexible control algorithms in the design of motion control is the fuzzy logic system. In certain instances, the design of the fuzzy controller is focused on individual and professional experience. Trial and error is used to establish the membership roles and the rule base itself . In recent years, there has been a growing interest in optimizing the fuzzy logic controller with numerous methods influenced by meta-heuristics. This paper focuses on the optimisation of a fuzzy controller applied to nonlinear structures. In certain instances, the problem is developed on the basis of the assumed linear behaviour, although objective functions and output parameters are interpreted in relation to the model have nonlinear responses. The optimisation algorithm is based on ant-lion optimizer implementation and JAYA algorithm as a hybrid process.

4.1.6 Elitism-Based JAYA Algorithm

In order to control the diversity behaviour of JAYA algorithm and utilize the survival of the fittest principle of natural selection in more proper manner, the JAYA algorithm is modified to elite some strong solutions to drive the search toward optimality.

Raut et al. [65] introduced an improved elitist-based JAYA algorithm for simultaneous network reconfiguration with distributed generation (DG) allocation. In their algorithm, the elitist-based concept is integrated with the JAYA algorithm by replacing the worst solutions with elites in order to avoid local trapping. The update equation is modified based on neighbourhood search and a linear decreasing inertia weight in order to better balance between the exploitation and the exploration abilities of JAYA algorithm. The effectiveness of their algorithm is tested using 33-bus and 69-bus distribution systems on four different case studies. Simulation results show the superiority of their algorithm in comparison with the original version of JAYA algorithm, bit-shift operator based particle swarm optimization, fireworks algorithm, genetic algorithm, and harmony search

algorithm in terms of active power loss reduction, voltage profile improvement and loadability enhancement.

Rao et al. [94] presented an elitist version of JAYA algorithm for economic optimization of shell-and-tube heat exchanger (STHE) design. Their approach is named elitist-Jaya algorithm. In elitist-Jaya algorithm, the elitism concept is used during every loop iteration in which the worst solution is duplicated by the elite solution(s). The parameters of the elitist-Jaya algorithm are studied using different settings to find the best settings. The efficiency of their algorithm is evaluated using three different problems of STHE. The simulation results illustrated the efficiency of the elitist-Jaya algorithm against GA, PSO, ABC, ICA, CSA, FFA, ITHS, CI, and GSA algorithms in terms of the solution quality and the CPU time.

Another elitist version of JAYA algorithm is proposed in [95], named Elitist-Jaya algorithm.This algorithm is applied for design optimization of shell-and-tube and plate-fin heat exchangers. The Elitist-Jaya algorithm is designed to simultaneously optimize the total annual cost and effectiveness of the heat exchangers. This algorithm is evaluated using two case studies of heat exchange problem. The impact of the three parameters of the Elitist-Jaya algorithm (population size, maximum iterations, and elite size) are experimentally studied using different settings. The simulation results illustrated the power of this algorithm against the genetic algorithm, and the modified TLBO in getting the optimum solutions within few iterations.

4.1.7 Other Modified Versions of JAYA Algorithm

There are other trials for improving the performance of JAYA algorithm to be inline with the search space of the optimization problem on hand.

Rao et al. [96] introduced a new modified version of JAYA algorithm for wind farm layout optimization. This algorithm is called multi-team perturbation-guiding Jaya (MTPG-Jaya) algorithm. The MTPG-Jaya algorithm utilized multiple teams being guided by different perturbation equations on a single set of population in order to explore the problem search space. Each team is triggered to enhance the same set of the population. The perturbation equation of the worst performing team is modified using the knowledge of the best teams. This knowledge includes the solution fitness value and the boundary violations of solutions. The MTPG-Jaya algorithm is evaluated using optimization test functions and three cases of wind farm layout optimization. Simulation results showed that the performance of the MTPG-Jaya algorithm is better than or competitive to the original version of JAYA algorithm and other comparative algorithms. In another study, the adaptive version of the MTPG-Jaya algorithm is presented in [97, 98]. In this algorithm, the number

of teams is adapted in order to efficiently explore the search space based on convergence performance.

Yu et al. [99] proposed a performance-guided JAYA (PGJAYA) algorithm to estimate the parameters of PV cell and module models. In PGJAYA, the solution performance in the population is qualified based on probability. Then, each solution self-adaptively select different evaluation mechanisms that are designed to strike the balance between exploration and exploitation abilities during the search process. The qualified performance is used to select the exemplar to establish the promising search direction. Finally, the self-adaptive chaotic perturbation mechanism is utilized to explore better solutions around the global best for replacing it with the worst ones. This is to improve the quality of the whole population. Simulated results showed that the PGJAYA algorithm is an efficient method for extracting model parameters of PV cell and module.

Another modified version of JAYA algorithm is designed for optimal power flow incorporating renewable energy sources as in [100]. This algorithm is named MJAYA. In this algorithm, each individual in the population is enhanced using the features of the fittest individual, while the features of the worst individual are ignored. This modification leads to modify the position of each individual in the population to a place near to the best individual. The efficiency of the MJAYA algorithm is evaluated using five case studies of IEEE 30-bus system and IEEE 118-bus system. The simulation results illustrated the superiority of the MJAYA algorithm against the moth swarm algorithm, artificial bee colony algorithm, cuckoo search algorithm, grey wolf optimizer, and backtracking search optimization algorithm in terms of the solution quality and the number of evaluations.

Mane et al. [101] introduced another improved version of JAYA algorithm, called semi-steady-state JAYA (SJaya) Algorithm. In SJaya algorithm, the authors presented a new update strategy for the best and the worst individuals in the population. This is done by using a new criterion to accept a new individual and eliminate the old one. The efficiency of the SJaya algorithm is tested using twelve classical benchmark functions, as well as fuel cell stack design optimization. The simulation results illustrated the superiority of the SJaya against the original version of JAYA algorithm in terms of solution quality and speed of finding a near-optimal solution.

In [102], the authors presented a new modified version of the JAYA algorithm to optimize many-objective optimization problems (MaOPs). Their approach is named a many-objective JAYA (MaOJaya) algorithm. In MaOJaya, the current population and the modified population are combined into one population. The tournament selection scheme is used to select the individuals for the next iteration. This modification is designed to guide the algorithm search process towards the best individual. Finally, the Tchebycheff

as a decomposition method is used for solving the MaOPs. The MaOJaya algorithm is evaluated using DTLZ benchmark functions with three, five, eight and ten objectives. The experiment results showed that the MaOJaya algorithm obtained better results or compeitive results against the other comparative algorithms on solving the MaOPs.

a modified version of JAYA algorithm, called an oppositional Jaya (OJaya) algorithm is designed by Yu et al. [103]. In OJaya algorithm, the oppositional learning is integrated with the JAYA algorithm to expand the problem search space and enhance the population diversity. In addition, the distance-adaptive coefficient is utilized to determine the best and the worst individuals in order to guide the individuals in the populations towards the best solution and escaping from the worst solution. The efficiency of the OJaya algorithm is tested on 3-bus, 8-bus, 9-bus and 15-bus testing systems of directions over current relays coordination problem. The simulation results showed the superiority of the OJaya algorithm against the original version of JAYA algorithm and other comparative algorithms in terms of solution quality, convergence speed, robustness and computation efficiency.

4.2 Hybridized Versions of JAYA Algorithm

The other versions of JAYA algorithm depends on the hybridization concepts. JAYA is hybridized with deep learning algorithms, evolutionary based algorithms, swarm based algorithms and other components of different metaheuristic-based algorithms. The hybridization of JAYA algorithm with other algorithms is provided in this section.

4.2.1 Hybridization with Neural Network

The JAYA algorithm is combined with the neural network ensemble (NNE) for diagnosis of brain tumor using MRI images [104]. The NNE is used in classifications into normal and abnormal brains using MRI images. The JAYA algorithm is used for extraction of the abnormal portion of the brain. Finally, the NNE is used again for further classification into the possibility of having a benign or malignant tumor. Simulation results showed that the JAYA algorithm is performing better than the PSO and GA algorithms.

Ramesh et al. [70] combined deep neural network (DNN) with JAYA algorithm (JOA) for detecting the rice leaf diseases. Their algorithm is called DNN-JOA. In the acquisition of images phase, the images of rice plant are taken from the farm with different characteristics as normal, bacterial blight, brown spot, sheath rot and blast diseases. In the pre-possessing phase, the background of images is removed, the RGB images are converted to HSV images, then the images are converted to binary images to create a mask. The output of this phase to split into the diseased and non-diseased parts in the images. In the segmentation phase, the DNN-JOA

algorithm as a classifier is utilized in order to classify the rice plant diseases, while the best weights are selected by the JOA algorithm. The simulation results illustrated the superiority of the DNN-JOA algorithm against ANN, DAE and DNN classifiers in terms of accuracy, precision, F1-score, TNR, TPR, FPR, FNR, FDR and NPV.

Bansal et al. [105] presented a new technique based on the JAYA algorithm and the feed-forward neural network to determine the maintainability cost of the software. The significance of their algorithm is evaluated using user interface management system (UIMS) and quality evaluation system (UIMS) datasets in terms of the mean absolute error as the analysis parameter. The performance of their algorithm is compared with another version of JAYA algorithm, three different versions of the clonal selection algorithm, and traditional statistical approach. The experimental results demonstrated that their algorithm is able to reduce the mean absolute error in comparison with others.

Khatir et al. [106] integrated an improved artificial neural networks (ANN) with JAYA algorithm for crack identifications in plates. This algorithm is named ANN-Jaya. Six accelerometers are used for modal analysis for three scenarios based on different crack angles to collect the data and estimate the parameters of ANN. Dynamic and static datasets are introduced using eXtended IsoGeometric Analysis (XIGA) to improve the accuracy of the ANN-Jaya algorithm. Furthermore, the XIGA datasets of cracked plate are used to improve ANN technique for static and dynamic analyses. The crack length is predicated using XIGA model based on experimental analysis. Jaya algorithm is used to optimize the most important parameters of ANN. The simulation results showed that the ANN-Jaya algorithm is robust and accurate for crack identification in plates.

A new version of JAYA algorithm to train the artificial multilayer perceptron (MLP) neural NARX model is introduced in [69]. This algorithm is named Jaya-NNARX. This algorithm is utilized for nonlinear system identification. The efficiency of the Jaya-NNARX algorithm is tested using two typical nonlinear benchmark test functions, and the simulation results showed that the performance of the Jaya-NNARX algorithm is better than the BP-NNARX, DE-NNARX, and PSO-NNARX algorithms. In another study, the JAYA algorithm is integrated with the neural network model for wire electric discharge machining (WEDM) of machining maraging steel [107]. The neural network model is used to establish the correlation between parameters and machining characteristics. In addition, the JAYA algorithm is triggered in order to recommend the optimal settings of process parameters. This is to enhance the matching performance. Simulation results illustrated the power of this algorithm in solving such problem.

4.2.2 Hybridization with Evolutionary-Based Algorithms

Wu et al. [50] proposes a hybrid JAYA algorithm for solving set-union knapsack problems (SUKP), named DHJaya. In DHJaya algorithm, the JAYA algorithm is combined with the mutation and the crossover operators of the differential evolution algorithm to enhance the diversity of the population and the exploration ability. The performance of DHJaya algorithm is tested using three large-scale SUKP datasets. Experimentally, the DHJaya performs better than the original version of JAYA and the basic differential evolution algorithm in terms of solution quality and convergence speed. In addition, the DHJaya algorithm obtained superior results against those of five other comparative methods.

Similarly, the hybridization of the JAYA algorithm with the adaptive differential evolution to identify the Bouc–Wen hysteresis model of a piezoelectric actuator is introduced in [71]. This hybrid algorithm is called aDE-Jaya. In aDE-Jaya, the mutation operator of differential evolution and the JAYA operator are combined to control the balance between the exploration and exploitation abilities. The mutant factor, the crossover rate and the population size are adapted to enhance the convergence efficiency. The aDE-Jaya algorithm is evaluated using 8 classical benchmark functions. The simulation results show the efficiency of the aDE-Jaya algorithm against the DE, PSO, JAYA, and GA algorithms in terms of the solution quality and the speed of convergence. Finally, the aDE-Jaya algorithm is utilized to identify the Bouc–Wen hysteresis model of a piezoelectric actuator with good results.

Another hybridization algorithm between the JAYA algorithm and the self-adaptive differential evolution algorithm is introduced in [72]. This algorithm is named Jaya-jDE and it is used for addressing the problem of designing green LTE networks with the Internet of Things (IoT) nodes. The performance of the Jaya-jDE algorithm is compared with the original version of JAYA, the teaching-learning-optimization (TLBO), and the self-adaptive differential evolution algorithm. The simulation results show the efficiency of the Jaya-jDE algorithm by achieving a good trade-off even when it did not have the best performance.

In [73], the authors proposed a hybrid swarm intelligence algorithm to conduct structural damage identification by using vibration measurement data. In their algorithm, the JAYA algorithm is combined with the Tree Seeds Algorithm (TSA) and K-means clustering, named C-Jaya-TSA. The clustering strategy is used to replace solutions with low-quality objective values in the JAYA algorithm. Furthermore, the TSA is utilized in the C-Jaya-TSA algorithm in order to enhance the exploitation ability of the JAYA algorithm . The simulation results show the effectiveness of the C-Jaya-TSA algorithm when testing on classical mathematical benchmarks and benchmark structure.

The JAYA algorithm is integrated with the forest optimization algorithm (FOA) for gene selection [108]. This hybrid algorithm is known as EJFOA. Initially, the ANOVA as a statistical filter is used to select the relevant genes from the original dataset. Later on, the EJFOA algorithm is triggered to find the optimal set of genes from the previously selected genes, while the support vector machine is employed as a classifier to classify the micro-array data. Finally, the EJFOA algorithm is evaluated using seven data sets published in the Kent Ridge Biomedical Data set Repository. The EJFOA algorithm achieved competitive results with other algorithms in terms of selecting the minimum subset of genes with maximum accuracy rate.

4.2.3 Hybridization with Swarm-Based Algorithms

Azizi et al. [76] integrated the ant lion optimizer (ALO) with the JAYA algorithm to optimize a fuzzy logic controller. Their algorithm is called ALO–Jaya algorithm. In this algorithm, the JAYA algorithm is implemented in updating the position of ants and antlions in order to manipulate the exploration and exploitation abilities of ALO algorithm. The ALO–Jaya algorithm is evaluated using eight optimization functions. The results show that the performance of the ALO–Jaya algorithm performs better than the original version of JAYA and the original version ALO algorithms. For more validations, the ALO–Jaya algorithm is tested using two benchmark buildings with nonlinear behavior. The performance of the ALO–Jaya algorithm is compared with eight comparative algorithms using these data sets. The results show that the ALO–Jaya algorithm is more effective than others in reducing the building responses.

In another study, the authors in [74] introduced a new hybrid algorithm, called JAYA-IPSO algorithm. In this algorithm, the improved particle swarm optimization (IPSO) algorithm is combined with the JAYA algorithm in order to enhance the exploitation ability of the JAYA algorithm. The JAYA-IPSO algorithm is applied for solving the problem of synchronization control for chaotic maps based on parameter estimation. Their algorithm is evaluated using two classical chaotic maps and their fractional-order form. The experimental results show the effectiveness of the JAYA-IPSO algorithm compared to four other optimization algorithms.

In [109], the authors combined the JAYA algorithm with directed particle swarm optimization (DPSO) for delivering nano-robots to cancer areas. Their algorithm is named Directed Jaya (DJaya) algorithm. The aim of this combination is to speed up the process of Nano-robots delivery. In DJaya, the JAYA algorithm is triggered first, while the DPSO algorithm is triggered last during the process of Nano-robots delivery. The simulation results show that the DJaya algorithm is faster than JAYA and DPSO algorithms

in the release of drugs, and thus destroying the cancer cells with relatively small numbers of Nano-robots.

Similarly, the JAYA algorithm is integrated with the firefly algorithm as a hybrid algorithm for video copyright protection in [75]. The JAYA algorithm is utilized in this hybrid algorithm to solve the problem of getting stuck in local optima for the firefly algorithm. This hybrid algorithm is tested using nine test data sets. The simulation results show the efficiency of the present hybrid algorithm in comparison with other algorithms.

4.2.4 Hybridization with Physical-Based Algorithms

In [55], the JAYA algorithm is hybridized with the learning operator of the teaching–learning-based optimization (TLBO) algorithm and time-varying acceleration coefficients (TVACs), called the LJaya-TVAC algorithm. This hybrid algorithm is applied for solving standard real-parameter test functions and various types of reliability–redundancy allocation problems (RRAPs) in two stages of experiments. In the first stage of experiments, the LJaya-TVAC algorithm is evaluated using six unimodal and multi-modal test functions. The simulation results show that the LJaya-TVAC algorithm is able to get better solutions with higher convergence rates than the original version of JAYA algorithm. In the second stage of experiments, the LJaya-TVAC algorithm is tested on RRAPs test problems with good results in comparison with the original JAYA algorithm and other algorithms.

4.2.5 Hybridization with Other Components

Luo et al. [77] they presented a hybrid JAYA algorithm by combining the JAYA algorithm and Nelder-Mead (NM) algorithm for identifying photovoltaic cell model parameters. This algorithm is called Jaya-NM algorithm. The NM algorithm is used as a local search component to enhance the exploitation capability of the JAYA algorithm. Jaya-NM algorithm is evaluated using a real dataset collected for this research. Experimentally, the performance of the Jaya-NM algorithm performs better than six other algorithms from the literature.

In [83], the binary JAYA algorithm as a wrapper-based technique is integrated with the filter-based normalized difference measure (NDM) for reducing the high-dimensional feature space of text classification problems. This algorithm is named NDM-BJO. In NDM-BJO algorithm, the digitization process is utilized to convert the continues values to binary. The Naive Bayes and Support Vector Machine classifiers are used to nominate the optimal subset of features which are generated by the NDM-BJO algorithm. This algorithm is tested using four well-known benchmark text corpus datasets. The performance of the NDM-BJO algorithm is compared with five filter-based algorithms and two

wrapper-based algorithms. From experimental results, it can be seen that the NDM-BJO-algorithm performs better than the other algorithms in term of accuracy.

In another study, the Lévy Flight (LF) technique is integrated with the JAYA algorithm for the non-linear channel equalization problem in [78]. This algorithm is known as JAYALF algorithm. The Lévy Flight is used to solve the shortcomings of the JAYA algorithm like the weak exploration capability and trapping into local minima due to insufficient diversity of population. The JAYALF algorithm is tested using seventeen well-known unimodal and multi-modal benchmark functions and three wireless communication channels with two different nonlinearities. Experimentally, the JAYALF algorithm outperforms the state-of-the-art algorithms in terms of solution quality, convergence speed, and robustness. The authors mentioned that JAYALF algorithm has a better exploration ability and converges quickly without getting stuck in local optima.

An improved version of the JAYA algorithm is presented for flexible job shop scheduling and rescheduling problems [110]. In this algorithm, the JAYA algorithm is integrated with two local search heuristics in order to improve the exploitation ability of JAYA algorithm. For evaluation purposes, this algorithm is tested using ten different-scale real-world datasets. This algorithm achieved superior results in comparison with evolutionary algorithms, tabu search, artificial bee colony, and ant colony optimization by stratifying the balance between instability and makespan in a rescheduling phase.

Li et al. [53] they proposed an improved JAYA (IJaya) algorithm for solving the flexible job shop scheduling problem. In the IJaya algorithm, seven local search heuristics are integrated with the JAYA algorithm in order to enhance the exploitation ability. Furthermore, the acceptance criteria of the simulated annealing algorithm is employed in order to enhance the exploration ability of the JAYA algorithm. Thirty instances with different scales as a benchmark dataset are used for evaluation purposes. The simulation results show the effectiveness of the IJaya algorithm on solving the considered problem.

Another improved JAYA algorithm called IJMSO is presented in [56]. The IJMSO algorithm is used for solving team formation problems. In IJMSO algorithm, the single-point crossover is combined within the Jaya algorithm to accelerate the search process, while the swap heuristic as a local search procedure is integrated with the JAYA algorithm to verify the consistency of the capabilities and the required skills to carry out the task. The IJMSO algorithm is tested using two real-world datasets and it is compared with four other algorithms. The experimental results show that the IJMSO algorithm is able to achieve better results faster than the other comparative algorithms.

An improved version of the JAYA algorithm for structural damage identification is introduced in [111], called I-Jaya algorithm. The low-quality objective values are replaced with a cluster strategy, while the solution update equation is modified using the best-so-far solution. The aim of these modifications to improve the global search ability. Furthermore, the sparse regularization and Bayesian inference are considered in the traditional objective function to perform the damage identification and enhance the robustness of the JAYA algorithm. Experimentally, the I-Jaya algorithm is able to provide accurate and reliable damage identification in comparison with other comparative algorithms.

Nayak et al. [112] developed a modified version of the JAYA algorithm to detect sensorial hearing loss through brain MR images. In their algorithm, the mutation operator is combined with the JAYA algorithm to enhance the global search ability of the JAYA algorithm. In addition, the extreme learning machine (ELM) is integrated with the JAYA algorithm to perform classification to ensure better generalization performance than its counterparts. Their algorithm is named MJaya-ELM. Their algorithm is tested using a well-studied hearing loss dataset. The performance of the MJaya-ELM is compared with the other comparative algorithms such as PSO-ELM, DE-ELM, and Jaya-ELM. The experimental results show the superiority of the MJaya-ELM algorithm in terms of accuracy and other performance measures. It should be noted that the authors mentioned that their algorithm was evaluated using a small dataset, and the performance of their algorithm may be affected when tested using large-scale dataset.

4.3 Multipupulation JAYA Algorithm

An elitism-based self-adaptive multipopulation JAYA algorithm for the design optimization of heat pipes is introduced in [67]. This algorithm is named SAMPE-Jaya algorithm. In SAMPE-Jaya, the population is divided into a set of subpopulations based on the quality of the solutions in order to manipulate the diversity of the JAYA search mechanism. Furthermore, the number of subpopulations is changeable during the search based on the change strength of the problem. The worst solutions in the inferior subpopulation are exchanged with the elite solutions in the superior subpopulation. The JAYA algorithm is triggered to update the solutions on each subpopulation separately. At the end of each iteration, the entire subpopulations are combined, and then check if the global best solution is enhanced or not. If the global best solution is enhanced then the number of subpopulations is increased by 1, otherwise the number of subpopulations will be decreased by 1. The performance of the SAMPE-Jaya is evaluated using the single- and multi-objective optimization cases of heat pipes problem. Experimentally, the SAMPE-Jaya algorithm achieved better results with

less computational times compared to other comparative algorithms. It should be noted that the same SAMPE-Jaya algorithm is applied for solving the constrained economic optimization of shell-and-tube Heat exchangers in [113]. In addition, the same SAMPE-Jaya algorithm is used to solve CEC 2015 benchmark functions in [66, 114].

Another self-adaptive multipopulation JAYA algorithm is presented in [115]. This algorithm is called MO-SAMP-JAYA algorithm and it is implemented for optimization of thermal devices and cycles. In MO-SAMP-JAYA algorithm, the population is divided into a set of subpopulations based on the nondominance rank and crowding distance of solutions. The best solution in the population has the highest rank, while the worst solution has the lowest rank. If two or more solutions have the same rank, the solution with the highest value of crowding distance is selected as the best solution. The JAYA algorithm runs on each subpopulation separately. The number of subpopulations is adapted based on the improvements on the global best solution at each iteration. The simulation results show the superiority of the MO-SAMP-JAYA algorithm in comparison with the latest algorithms in the literature.

4.4 Parallel JAYA Algorithm

Migallon et al. [116, 117] introduced a modified version of the JAYA algorithm based on parallel structure. In their algorithm, the population is divided into a set of subpopulations. The number of the subpopulations is fixed during the search. Furthermore, the JAYA algorithm is running on each subpopulation at one of the processors separately. It should be noted that the migration process is not allowed between subpopulations. Their algorithm is evaluated using 30 unconstrained benchmark functions with good results. The same authors presented another parallel JAYA algorithm in [118]. In this algorithm, the chaotic 2D map is combined with the parallel JAYA algorithm in order to speed up the convergence. Their algorithm is tested using 18 classical benchmark functions and two real-world engineering design problems.

In another study, a parallel version of the JAYA algorithm is implemented. The graphics processing unit (GPU) is used to estimate parameters of the Li-ion battery model [61]. This algorithm is named GPU-Jaya. In this algorithm, the iterative update of solution candidates, the computation of the solution quality, and the selection of the globally best and worst solutions are all executed in parallel on the GPU. The global memory and the shared memory are employed in the GPU-Jaya algorithm. The global memory is utilized in the solution update and the computation of the solution quality, while the shared memory is applied to select the best and worst solutions. The simulation results show that the GPU-Jaya algorithm is able to estimate battery model parameters

with less computational time opposed to the other comparative algorithms.

4.5 Multi-Objective JAYA algorithm

Rao et al. [119] introduced an efficient multi-objective JAYA algorithm for tackling the respective machining process optimization problems. Their algorithm is named MO-Jaya. The effectiveness of their algorithm is tested using four case studies including wire-electric discharge machining process, laser cutting process, electrochemical machining process and focused ion beam micro-milling process. The simulation results illustrated that the performance of the MO-Jaya algorithm is better than GA, NSGA, NSGA-II, BBO, NSTLBO, PSO, SQP and Monte Carlo simulations. The same authors applied the same MO-Jaya algorithm for solving the input process parameters optimization problems of abrasive waterjet machining process in [120, 121]. The experimental results showed the efficiency of the MO-Jaya algorithm against the simulated annealing, the particle swam optimization, the firefly algorithm, the cuckoo search algorithm, the black hole algorithm and the biogeography based optimization in terms of the solution quality and the computational time. Similarly, the MO-Jaya algorithm is applied for optimization of modern machining processes in [122]. The plasma arc machining, electro-discharge machining, and micro electro-discharge machining processes are case studies used to evaluate the MO-Jaya algorithm. Simulation results showed the efficiency of the MO-Jaya algorithm on real manufacturing environment.

Mishra et al. [123] presented a multi-objective JAYA algorithm for solving permutation flow shop scheduling problem. The minimization of the total tardiness cost and makespan are the two objectives considered when running their algorithm to solve such problem. Simulation results showed the efficiency of their algorithm compared with total enumeration method and simulated annealing algorithm in terms of solution quality and CPU time.

The efficiency of the multi-objective JAYA (MOJaya) algorithm for tackling optimal power flow in investigated in [124]. The MOJaya algorithm tries to satisfy different objectives like fuel cost minimization, voltage profile improvement, voltage stability enhancement, active power losses reduction and system security enhancement. The MOJaya algorithm is evaluated on IEEE 30-bus test system. Five cases as a single-objective and eleven cases, considered as multi-objectives, are used as case studies. Simulation results showed the superiority of the MOJaya algorithm compared to other methods in the literature in case studies with one, two, and three objectives.

Rao et al. [125] applied the JAYA algorithm for the single- and multi-objective design optimization of plate-fin heat exchangers (PFHEs). Their algorithm is applied for PFHEs

in order to minimize the total surface area of heat transfer, total annual cost, and total pressure drop of the system and maximizing the effectiveness in four design case studies. Simulation results showed the superiority of their algorithm in comparison with GA, ICA PSO, BBO and TLBO algorithms in terms of the solution quality and computational time. In another study, the efficiency of the single- and multi-objectives JAYA algorithm is investigated on optimization of casting processes in [126]. These casting process such as squeeze casting process, continuous casting process, pressure die casting process and green sand casting process. This algorithm is named QO-Jaya. The performance of QO-Jaya algorithm is compared with SA, GA, PSO, TLBO and aguchi methods. It can be seen from the the experimental results that the QO-Jaya performs better or similar to other methods.

In [127], the authors presented a multi-objective JAYA algorithm for optimal scheduling of distributed generators in distribution system sectionalized into multi-microgrids. Their algorithm used to optimize multiple-objective cases such as i) minimizing the total operating cost, ii) minimizing the system loss, and iii) minimizing of voltage deviation. In addition, their algorithm is investigated to satisfy the multi-objectives such as: i) minimizing the total cost and system loss, ii) minimizing the total cost and voltage deviation, and iii) minimizing the system loss and voltage deviation. Their algorithm was evaluated using a dataset with 33 bus network distribution. Simulation results illustrated the effectiveness of their algorithm against with genetic algorithm.

Venkaiah et al. [128] introduced a multi-objective JAYA algorithm to find the optimal location and sizing of distributed generation in a radial distribution system. Their algorithm is known as MOJAYA. The MOJAYA is used to optimize multiple objectives such as reducing power loss, improving voltage profile and stability. The MOJAYA algorithm is tested on IEEE 33 bus radial distribution system. The MOJAYA algorithm outperforms the MOPSO algorithm in terms of solution quality and computational time.

Barakat et al. [129] presented an enhanced version of JAYA algorithm for solving multi-objective optimal reactive power dispatch (ORPD) problem. This algorithm is called EJOA. The EJOA is designed in order to minimize the power losses and voltage deviation in power system. The multi-objective concept is utilized by weighted sum method with fuzzy logic. The EJOA algorithm is tested using two standard datasets (IEEE 30-bus and WDRN). Experimentally, the EJOA algorithm perform better than PSO and GA in terms of the solution quality and speed of convergence.

5 Applications of JAYA Algorithms

The viability and efficiency of JAYA-based algorithms make it successfully applied in solving different real-world applications. These applications include mathematical functions [88, 101], feature selection [59, 82], image processing [130, 131], energy [99, 132], design proportional-integral-derivative (PID) controller [133, 134], economic dispatch [135, 136], communication [72, 137], electrical and power system [137, 138], fracture mechanics [139, 140], environmental engineering [96, 141] , manufacturing industries [120, 121], structural design [111, 142], truss structures [57, 85], planing and scheduling [50, 87], and others [105, 143].

The applications of JAYA algorithm and its variations are listed in Table 2. The applications are classified in terms of the application domain, the name of applied JAYA algorithm, the name of tackled problem, the variant of JAYA algorithm used (i.e., original, modified, hybrid, multi-objective, or parallel), and the type of data representation of the undertaken problem (e.g., continuous or binary).

The following is a summary of some of the main applications that are solved by JAYA algorithm and its variations. Feature selection is a problem of selecting the most promising features to represent the whole data. The feature selection is used to solve a wide variety of real-world applications such as bioinformatics, pattern recognition, and classifications. JAYA algorithm and its variations [59, 82, 144] have been used for feature selection applications.

Image processing means recognizing images for classification purposes. JAYA algorithm and its variations have been used for image processing of MRI scans [104, 130, 131, 145, 146], ultrasound image [147], and emotion recognition [58], and other tasks [70, 147, 148].

Energy optimization means using available resources efficiently. JAYA algorithm has been used widely to tackle such problems. Photovolatic cells optimization is a crucial application due to its importance in providing clean energy. Several researchers used JAYA algorithm [77, 99, 132, 149, 150] for parameter estimation of photovoltaic cell models. Other state-of-the-art JAYA algorithms are used for li-ion batteries optimization [61] and optimal power flow sharing [100, 151].

The communication domain includes different types of problems that are tackled by JAYA algorithm such as load balancing in cloud computing [137], synthesis of linear antenna arrays [64, 64], non-linear Channel equalization [78], and simultaneous network reconfiguration and distributed generation (dg) allocation [65].

Electrical and power system optimization problems are tackled by JAYA algorithm and its modifications. This includes optimal power flow solution [47, 48, 138], reactive power dispatch [129], directional over current relays

Table 2 Application of JAYA algorithm on different domains

Domain	Algorithm	Problem	Variant	Type	Ref.
Mathematical function	SJaya	Engineering and benchmark functions	Modified	Continuous	[101]
	Parallel Jaya	Engineering and benchmark functions	Parallel	Continuous	[88]
	MaOJaya	Many-objective benchmark functions	Modified	Continuous	[102]
	Jaya	Mixed-variable problems	Modified	Mixed	[88]
	SAMP-Jaya	Numerical unconstrained and constraint problems	Multi-population	Continuous	[114]
	Jaya GPU	Unconstrained functions	Parallel	Continuous	[158]
Feature selection	BJAM	Feature selection	Modified	Binary	[59]
	BinJaya	Real-Time Transient Stability Assessment Using PMU Data	Modified	Binary	[82]
	FSJaya	Supervised classification	Modified	Binary	[144]
Image Processing	TS-Jaya	Identification of Alcoholism using MRI scans	Modified	Binary	[130]
	Jaya	Segment Tumor from Brain MRI	Original	Continuous	[131]
	MJaya	Multiple sclerosis identification from Brain MRI	Modified	Binary	[145]
	Jaya	Intelligent facial emotion recognition	Original	Binary	[58]
	Jaya	Fetal head periphery extraction from ultrasound image	Original	Binary	[147]
	FF-Jaya	Video copyright protection	Hybrid	Binary	[147]
	NN-Jaya	Diagnosis of Brain Tumor Using MRI Images	Hybrid	Binary	[104]
	ML-Jaya	Smart pathological brain detection Using MRI Images	Hybrid	Binary	[146]
	NN-Jaya	Recognition and classification of paddy leaf diseases	Hybrid	Binary	[70]
	Jaya	Tea category identification	Original	Binary	[148]
Energy	PGJAYA	Parameter estimation of photovoltaic cell models	Modified	Continuous	[99]
	EO-Jaya	Parameter estimation of photovoltaic cell models	Modified	Continuous	[132]
	H-Jaya-NM	Parameter estimation of photovoltaic cell models	Modified	Continuous	[77]
	Jaya	Parameter estimation of photovoltaic cell models	Original	Continuous	[149]
	IJAYA	Parameter estimation of photovoltaic cell models	Modified	Continuous	[150]
	GPU-Jaya	Li-ion batteries optimization	Parallel	Continuous	[61]
	MJAYA	Parameters extraction of solar cells	Modified	Continuous	[49]
	Jaya	Power sharing between the battery and the UC	Original	Continuous	[151]
	MJaya	Optimal power flow incorporating renewable energy sources	Modified	Continuous	[100]
Design PID controller	Jaya	optimized PID controller for Auto-matic Voltage Regulator	Original	Continuous	[133]
	Jaya	Optimization of FUZZY controller	Original	Continuous	[134]
	Jaya	2-DOF FOPID controller design for maglev system	Original	Continuous	[159]
	Jaya	Design and performance analysis of PID	Original	Continuous	[160]
	Jaya	Fuzzy-PID controller with derivative filter	Original	Continuous	[161]
	Jaya	Optimized Integral-Proportional Derivative (I-PD)	Original	Continuous	[162]
Economic dispatch	Jaya	Economic emission dispatch	Original	Continuous	[135]
	Jaya	Optimal reactive power dispatch	Original	Continuous	[136]
	Jaya	Economic Dispatch of Wind-Thermal Power	Original	Continuous	[153]
	ES-JAYA	estimation of proton exchange membrane fuel cells	Modified	Continuous	[154]
	Jaya	Optimization of Plate-Fin Heat Exchanger	Original	Continuous	[60]
	JAYA-IPSO	Economic emission dispatch	Modified	Continuous	[74]
	Chaotic JAYA	Economic emission dispatch	Original	Continuous	[89]
	Improved Jaya	Proton Exchange Membrane Fuel Cell Stack Design Optimization	Modified	Continuous	[163]

Table 2 (continued)

Domain	Algorithm	Problem	Variant	Type	Ref.
Communication	Jaya	Load Balancing in Cloud Computing	Original	Continuous	[137]
	HJaya-DE	5G Massive MIMO and NB-IoT GreenNetworks	Hybrid	Continuous	[72]
	Jaya	Synthesis of linear antenna arrays	Original	Continuous	[64]
	Jaya	Non-linear Channel Equalization	Original	Continuous	[78]
	IEJAYA	Simultaneous network reconfiguration and distributed generation (dg) allocation	Modified	Continuous	[65]
	MO-Jaya	Scheduling of distributed generators in Distribution System	Multi-Objective	Continuous	[127]
	Jaya, SJaya, and CJaya	Synthesize linear antenna arrays	Hybrid	Continuous	[64]
Electrical and Power System	Q-O M-Jaya	Multi-objective optimal power flow solution	Modified	Continuous	[138]
	EJOA	Reactive Power Dispatch	Multi-objective	Continuous	[129]
	OJaya	Directional over current relays coordination problem	Modified	Continuous	[103]
	Jaya	Power flow	Original	Continuous	[47]
	Jaya	Coordination of Overcurrent Relays in Single-and Multi-Loop Distribution System	Original	Continuous	[152]
	MJaya	Thermal performance optimization of the underground power cable system	Modified	Continuous	[164]
	Jaya	Automatic Generation Control Scheme for Interconnected Solar-Thermal Power System	Original	Continuous	[165]
	Jaya	Automatic Generation Control of Two-Area Thermal Power System	Original	Continuous	[166]
	SA-Jaya	Optimum configuration of chosen thermal equipment	Modified	Continuous	[63]
	MOSAMP	Dimensions of three thermal systems optimisation	Multi-objective	Continuous	[115]
	AMTPG-Jaya	Optimal power flow	Original	Continuous	[48]
	Jaya	Transient response and stability enhancement of regulator system	Original	Continuous	[167]
Fracture Mechanics	POD-RBF-Jaya	Fracture mechanics inverse problems	Hybrid	Continuous	[139]
	FEM-Jaya	Crack identification in plate structures	Hybrid	Discrete	[140]
Environmental Engineering	Jaya	Estimation of the soil water retention	Original	Continuous	[141]
	MTP-Jaya	Optimization of wind farm layout	Original	Continuous	[96]
	Jaya	Optimal points of PV-based DG injection	Original	Continuous	[168]
	MJaya	Optimal design and analysis of mechanical draft cooling tower	Modified	Continuous	[92]
Manufacturing industries	MO-Jaya	Optimization of Abrasive Waterjet Machining Process	Multi-objective	Continuous	[120]
	MO-Jaya	Optimization of Abrasive Waterjet Machining Process	Multi-objective	Continuous	[121]
	Hybrid Jaya	Optimization of proton exchange membrane fuel cells	Hybrid	Continuous	[154]
	Jaya	Crack identification in plates	Original	Continuous	[106]
	aDE-Jaya	Parameters identification of Bouc–Wen hysteresis model for piezoelectric actuators	Hybrid	Continuous	[71]
	Hybrid Jaya	Optimization of radioactive sources	Hybrid	Continuous	[169]
	QO-Jaya	Optimization of Selected Casting Processes	Modified	Continuous	[170]
	JA	Optimization of magnetic abrasive finishing	Original	Continuous	[171]
	MJA	Intelligent identification of permanent magnet synchronous machine parameters	Modified	Continuous	[155]
	Jaya	Maintenance of the multi-unit series system	Modified	Discrete	[172]
	Hybrid JAYA	Carbon fiber-reinforced polymer	Hybrid	Continuous	[156]
	Modified JAYA	Glass fiber reinforced polymer	Modified	Continuous	[93]
	QO-Jaya	Optimization of submerged arc welding process parameters	Modified	Continuous	[170]
	QO-Jaya	Optimisation of welding processes	Modified	Continuous	[173]
	QO-Jaya	Optimization of Casting Processes	Multi-objective	Continuous	[126]
	Hybrid JAYA	Wire electric discharge machining	Hybrid	Continuous	[107]

Table 2 (continued)

Domain	Algorithm	Problem	Variant	Type	Ref.
Structural Design	I-Jaya	Structural damage identification	Modified	Continuous	[111]
	Jaya	Structural damage identification	Modified	Continuous	[142]
	Jaya	Optimum design of a reinforced concrete counterfort retaining wall	Modified	Continuous	[174]
	ALO–Jaya	Optimum design of fuzzy logic controller	Hybrid	Continuous	[76]
Truss Structures	JA	Sizing and layout optimization of truss structures	Modified	Continuous	[57]
	DAJA	Sizing and layout optimization of truss structures	Modified	Discrete	[85]
	JAYA	Braced dome structures	Original	Continuous	[175]
	Hybrid JAYA	Optimization of squeeze casting process parameters	Hybrid	Continuous	[176]
Planing and scheduling	DHJaya	Set-union knapsack problem	Hybrid	Binary	[50]
	Modified Jaya	Threshing drum	Modified	Discrete	[87]
	JA	Optimization of Reservoir Operation	Modified	Discrete	[157]
	JAYA	Job shop scheduling	Modified	Discrete	[52]
	JAYA	Virtual Machine Placement	Modified	Discrete	[51]
	JAYA	Permutation flow-shop scheduling problem	Modified	Discrete	[54]
	JAYA	Permutation flow shop scheduling problem	Modified	Discrete	[123]
	LJaya-TVAC	Reliability–redundancy allocation problems	Hybrid	Discrete	[55]
	Improved Jaya	Flexible Job Shop Rescheduling Problems	Modified	Discrete	[110]
	DJaya	Flexible Job-Shop Rescheduling Problems	Modified	Discrete	[86]
	IJaya	Flexible Job-Shop Rescheduling Problems	Modified	Discrete	[53]
	IJMSO	Team formation problem	Hybrid	Discrete	[56]
	JAYA	Damage assessment in plate-like structures	Modified	Discrete	[177]
Other	NN-Jaya	Object-Oriented Software Quality Analysis	Hybrid	Continuous	[105]
	MS-Jaya	Software Module Clustering Problem	Modified	Continuous	[143]
	ELM-Jaya	Estimating currency exchange value	Hybrid	Continuous	[178]
	Jaya-ELM	Hearing loss detection	Hybrid	Continuous	[112]
	Jaya	Nano-robots deliver drugs to cancer area	Original	Continuous	[109]
	Jaya-BF	Gene selection for cancer classification	Modified	Binary	[108]

coordination problem [103, 152], and thermal systems optimisation.

Solving combined economic emission dispatch means scheduling generation at various interconnected generating plants to meet the required load demand while keeping the operating cost at a minimum level. JAYA algorithm has been used to solve such problem by different researchers such as [60, 74, 89, 135, 136, 153, 154].

Manufacturing has diverse kinds of optimization problems that are tackled by JAYA algorithm such as optimization of abrasive waterjet machining process [120, 121], crack identification in plates [106], parameters identification of Bouc–Wen hysteresis model for piezoelectric actuators [71], intelligent identification of permanent magnet synchronous machine parameters [155], carbon fiber-reinforced polymer [156], and glass fiber reinforced polymer [93].

Planning and scheduling tasks are combinatorial NP problems in which a set of resources are arranged within a set of constraints to reach to a specific goal. Several researchers are motivated to tackle scheduling problems such as optimization of reservoir operation [157], job shop scheduling [52], virtual machine placement [51], permutation flow-shop scheduling problem [54, 123], flexible job shop rescheduling problem [53, 86, 110], and team formation problem [56].

6 Open Source Software of JAYA Algorithm

In order to provide a comprehensive review of JAYA algorithm, open source software available to the JAYA research communities is discussed. Due to simplicity of the optimization framework, JAYA received a tremendous interest and became popular in a very short time. Since JAYA is parameter-free and one operator drives the search process, JAYA is implemented in few lines of code. Therefore, there are many trials that provide the code of JAYA algorithm to be available for the researchers in interest.

Since the first inception, the JAYA algorithm is coded in Matlab which is released as a public code[1][29]. The code tackled some standard benchmark functions which contributed in the JAYA popularity. This code is also uploaded in the GitHub Integration Platform to increase its publicity.[2]

There are some researchers tackled their optimization problems using JAYA algorithm and their code is made publicly available. In power systems, minimization of active power losses is one the main interest which is implemented by Orlando Ramirez Barron and the source code is provided[3]. Furthermore, JAYA algorithm is programmed in Java language to be incorporated with WEKA software, the workbench for machine learning[4]. The multi objective version of JAYA algoithm is also coded in Python programming language. The source code is made publicly available[5] by Berrouk et. al. [124]. Another open source of JAYA is coded in Python programming language[6]. Finally, the Matlab source code of JAYA algoirhm is released for unconstrained Rosenbrock[7] and constrained Himmelblaw[8] benchmark functions

7 Critical Analysis of JAYA Algorithm Theory

As summarized in the variants and applications of JAYA algorithm, JAYA has gained considerable attentions geared toward a vast variations of optimization problems. This is due to its impressive characteristics: Simple in concepts, easy-to-use, parameter-less, derivative-free, and sound-and-complete. This is the main reason of success and why the research communities are boosted to adopt this algorithm. However, as other metaheuristic algorithms, JAYA algorithm suffers from few limitations and inescapable drawbacks.

The main drawback of this algorithm is related to the No Free Lunch (NFL) Theorem of optimization [179]. In NFL, there is no Superior optimization algorithm that can excel over all competitors for the entire variants of optimization problems or even for all instances of the same optimization problem. Therefore, the convergence behaviour of JAYA

algorithm has tight connection with the nature of the problem search space on hand. Therefore, the JAYA algorithm need to be modified or hybridized to cope with the problem search space nature.

The second drawback is related to the problem domain to be manipulated. As aforementioned, the first proposal of JAYA is established for optimization problems with single-objectives, unconstrained objective functions and continuous domains [29]. However, JAYA needs to expand its applicability to be efficient for other problem domains and nature such as discrete, binary, combinatorial, dynamic, many objective and multi objective.

The third drawback of JAYA algorithm is related to its population-based behaviour. During the search process, JAYA algorithm is able to explore a wide range of search space regions through its unique operator which is able to evolve the search based on the inherited values attracted to the global best solution and move closely to better areas in the search space by expanding the distance with worse solutions. By means of this way, the search has a maneuver behaviour in its navigation. However, its operator, does not focus on each search space region to which it converge, thus degrading the exploitation process especially for the problems with multi-modal landscape. As a result, JAYA algorithm should be hybridized with other local search-based algorithm to enrich its exploitation capability. The hybridization of JAYA with local search techniques enables it to stay focused on potential regions of the search space.

As other optimization methods, the performance of JAYA algorithm is tightly related to the problem of dimensionality. As the number of the problem variables grows , the performance of the algorithm is degenerated. Therefore, dimensionality reduction in machine learning is a proper solution for some problem to yield more efficient results. Other scaling and transformation mechanisms should be considered. That is why in some applications, as stated in the previous sections, the GWO algorithm outperformed JAYA. The GWO is less susceptible to the dimensionality problem and may outperform JAYA in problems with high dimensionality.

Other common problem is the dependence of JAYA on stochastic operators that are really pseudo random. The pseudo randomness has an artificial cyclic nature which means, in the long run, some patterns will be repeating itself and consequently the search pattern will adapt also a cyclic behavior that might end up in an endless loop of search. However, the behavior of JAYA which is somewhat similar to Nelder-Mead and Simplex algorithms except that JAYA has stochastic components, that hinder its ability to span multi dimensional search spaces without any cyclic behavior.

Finally, there is a concur dilemma in JAYA algorithm related to the control of the diversity. The convergence

---

1 https://sites.google.com/site/jayaalgorithm/

2 https://github.com/sunmayra/Jaya-Algorithm

3 https://uk.mathworks.com/matlabcentral/fileexchange/71220-optimum-power-flow (access date: 28/08/2020)

4 https://github.com/sunmayra/Jaya-Algorithm (access date: 28/08/2020)

5 https://ch.mathworks.com/matlabcentral/fileexchange/72330-multiobjective-jaya-algorithm-mojaya (access date: 28/08/2020)

6 https://github.com/eadomench/pyjaya/find/master (access date: 28/08/2020)

7 https://drive.google.com/file/d/0B96X2BLz4rx-OHlBeGF3VjdKcjg/view?usp=sharing (access date: 28/08/2020)

8 https://drive.google.com/file/d/0B96X2BLz4rx-by01U09BUDB2OHM/view?usp=sharing (access date: 28/08/2020)

behaviour of JAYA algorithm tends to be stagnated to the local minima quickly due to the lose of diversity in the initial course of run. This is because the unique operator of JAYA algorithm considers only two solutions (i.e., the best and the worst) at each generation and do not considered the other attributes of other solutions in the population. It will be attracted to the best solution and escape the features of the worst solution. In optimization, it is conventionally known that the search direction to the global optima is not necessarily the strike to the best solution. Therefore, the loss of diversity exists due to neglecting the majority of other solutions in the population and immaturity occurs. Some works suggest dividing the population into sub-populations and dealing with each sub-population with isolated JAYA to control the diversity. It is also fond that using parallel JAYA can help is speeding up its convergence dramatically without much affecting the quality of solutions.

8 Conclusion and Future Trends

JAYA algorithm in its simple form and in its variants have demonstrated outstanding ability to solve multi-objective, multi-variable, and complex optimization problems. The majority of successful applications were engineering applications due to the ability of JAYA algorithm to handle numerical, continuous, and binary search spaces with well defined objective functions that could come either with constraints or without constraints. In many cases there was a need to add variants to the standard JAYA algorithm to increase the capability of JAYA for solving the problems under scope. As a matter of fact, there are three ways to increase the effectiveness of JAYA; adding variants, hybridization with other optimization techniques, or integrating it with other AI paradigms. Adding the variants to utilize a the other components of the standard JAYA algorithm proved to be very efficient in upgrading its performance and enabling it to solve the multi-objective optimization problems being tackled. The variants did not change the general shape of JAYA but made some of its operations much more efficient and rewarding. Those variants could be implemented on the parameter level, the population level, and/or operators level. Some of those variants are simply heuristics that are added to the algorithm, others constituted minor changes to the steps of the algorithm. However, they were all necessary to meet the demands of the problems being optimized. Those problems are sometimes multi-objective with abnormal nature of inputs and specific requirement for the output.

On the other hand, hybridization could be implemented in conjunction with some local search techniques, or with others global search techniques that have different nature than JAYA. Hybridization has been proposed to enhance the exploitation capability that JAYA has in its quest to find

optimum solutions. As a matter of fact, JAYA is integrated with other AI paradigms in an attempt to increase the efficiency of those paradigms. Its worth noting that, in general, the hybridization did not add considerable complexity to those paradigms due to JAYA natural simplicity, at the same time, it elevated the quality of the outcomes of those paradigms. Examples of those paradigms are neural networks, fuzzy controllers, features selection mechanisms, pattern recognition, and many other methods.

JAYA algorithm also was incorporated with other intelligent paradigms such as neural networks, it has a multi-purpose capabilities including training the neural networks, input groups selection, activation functions tuning and optimizing neural networks structures. In fuzzy systems it is used in tuning the fuzzy membership functions and other parameters of fuzzy controllers. In features selection, JAYA and its variants proved to be efficient in features optimization. The simplicity of JAYA and its variants made it preferable to be integrated with other much more complex systems.

As the JAYA algorithm provide a successful outcomes for different research domains, there are other venues of future enhancements that can be suggested. These possibilities can be summarized as follows:

Structured population The JAYA algorithm tends to lose diversity during its search due the fast convergence problem. One of the best ways to deal with controlling the diversity is the structured population. The population is divided into several sub-population and each sub-population is used as an initial input to independent JAYA version. The sub-populations exchange their solution periodically. The most structured population can be used for JAYA algorithm is Island model [180–182], Cellular Automata [183, 184], hierarchical model [185], and other reported in [186].

Selection mechanisms The JAYA algorithm only focus on the best and the worst solutions in the evolution loop. This can lead the algorithm to neglect several important directions which can lead to the global optimal solution. Therefore, instead of using only best and worst solution, the selection mechanisms such as proportional selection, tournament selection, linear and exponential rank selection can be utilized in to improve its diversity control and thus improve performance.

Adaptive Parameters The process of adapting parameters during the search helps the algorithm to find the optimal parameter configurations for the better performance [187]. Although the JAYA algorithm is parameter-free, the algorithmic parameters such as number of solutions in the initial population and the number of iterations are in many cases sufficient to reach local or global optima. Adaptive parameters can be used to autonomously tune those parameters during evolution.

Memetic Strategy As conventionally known, the population based algorithms such as JAYA algorithm are very powerful in exploring a different regions of search space at the same time. However, they are poor in exploiting each region to which they converge. On the other hand, local search-based algorithms are very powerful in exploiting each region and finding the local optima in that region, yet their disadvantage is in the exploration part. Consequently, hybridizing a local search-based within the evolution step of population-based algorithm can complement the powerful feature of both types of algorithms. This type of hybridization methods is called memetic strategy. Ultimately, JAYA algorithm can be hybridized with a local search to enrich its exploitation capabilities.

Dynamic Optimization The dynamic optimization concern where dynamic problems optimal solution is changing over the time. There is very few work that tackle dynamic algorithms using JAYA-based algorithm. This is indeed a very challenging type of problems that require special capability in navigating several search space regions with dynamic behaviour.

Combinatorics optimization Normally, the real-world optimization problems are not trivial since they have a combinatorial optimization nature. This type of problems is constrained and the search space is not smooth and rugged. Therefore, the best constraint handling technique should be investigated to deal with the combinatorial optimization problems such as scheduling and planning domains. Normally, this type of problems can work only on feasible search space regions.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Burke Edmund K, Gendreau Michel, Hyde Matthew, Kendall Graham, Ochoa Gabriela, Ozcan Ender, Rong Qu (2013) Hyper-heuristics: a survey of the state of the art. J Operational Res Soc 64(12):1695–1724

2. Hussain Kashif, Salleh Mohd Najib Mohd, Cheng Shi, Shi Yuhui (2019) Metaheuristic research: a comprehensive survey. Artif Intell Rev 52(4):2191–2233

3. Osman Ibrahim H, Laporte G (1996) Metaheuristics: a bibliography. Ann Oper Res 63:511–623

4. Blum Christian, Roli Andrea (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput Surv (CSUR) 35(3):268–308

5. Fausto Fernando, Reyna-Orta Adolfo, Cuevas Erik, Andrade Angel G, Perez-Cisneros Marco (2020) From ants to whales: metaheuristics for all tastes. Artif Intell Rev 53(1):753–810

6. Meraihi Y, Ramdane-Cherif A, Acheli D, Mahseur M (2020) Dragonfly algorithm: a comprehensive review and applications. Neural. Comput. Appl. 32:16625–16646

7. Holland John H (1992) Genetic algorithms. Scientific Am 267(1):66–73

8. Storn Rainer, Price Kenneth (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Global Optimization 11(4):341–359

9. Yao Xin, Liu Yong, Lin Guangming (1999) Evolutionary programming made faster. IEEE Transactions Evolut Comput 3(2):82–102

10. Beyer Hans-Georg, Schwefel Hans-Paul (2002) Evolution strategies-a comprehensive introduction. Natural computing 1(1):3–52

11. Koza John R, Koza John R (1992) Genetic programming: on the programming of computers by means of natural selection, vol 1. MIT press

12. Baluja S (1994) Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning. Technical report Technical report CMU-CS-94-163, Carnegie Mellon University

13. Simon Dan (2008) Biogeography-based optimization. IEEE Transactions Evolut Comput 12(6):702–713

14. Azmi Al-Betar Mohammed (2017) β-hill climbing: an exploratory local search. Neural Comput Appl 28(1):153–168

15. Kirkpatrick Scott, Gelatt C Daniel, Mario P VecchiVecchi Mario P VecchiMario P (1983) Optimization by simulated annealing. Science 220(4598):671–680

16. Glover F., Laguna M. (1998) Tabu Search. In: Du DZ, Pardalos PM (eds) Handbook of Combinatorial Optimization. Springer, Boston, MA. https://doi.org/10.1007/978-1-4613-0303-9_33

17. Lourenco Helena R, Martin Olivier C, Stutzle T (2003) Iterated local search. Handbook of metaheuristics. Springer, Newyork, pp 320–353

18. Mladenovic Nenad, Hansen Pierre (1997) Variable neighborhood search. Comput & Oper Res 24(11):1097–1100

19. Feo Thomas A, Resende Mauricio GC (1995) Greedy Randomized Adaptive Search Procedures. J Glob Optim 6:109–133. https://doi.org/10.1007/BF01096763

20. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN95-International Conference on Neural Networks. 4: 1942–1948. IEEE

21. Dorigo M, Caro G Di (1999) Ant colony optimization: a new meta-heuristic. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406), volume 2, pages 1470–1477. IEEE

22. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer

23. Yang Xin-She et al (2008) Firefly Algorithm. Nature-Inspired metaheuristic Algorithms 20:79–90

24. Yang Xin-She, Gandomi Amir Hossein (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput 29(5):464–483

25. Xin-She, Y (2012) Flower pollination algorithm for global optimization. In: International conference on unconventional computing and natural computation, pages 240–249. Springer

26. Mirjalili Seyedali, Mirjalili Seyed Mohammad, Lewis Andrew (2014) Grey wolf optimizer. Adv Engineering Softw 69:46–61

27. Gandomi Amir Hossein, Alavi Amir Hossein (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numerical Simul 17(12):4831–4845

28. Mirjalili Seyedali (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowledge-based Syst 89:228–249

29. Rao R (2016) Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int J Indus Eng Comput 7(1):19–34

30. Formato Richard A (2008) Central force optimization: a new nature inspired computational framework for multidimensional search and optimization. Nature inspired cooperative strategies for Optimization (NICSO 2007). Springer, Newyork, pp 221–238

31. Rashedi Esmat, Nezamabadi-Pour Hossein, Saryazdi Saeid (2009) Gsa: a gravitational search algorithm. Information Sci 179(13):2232–2248

32. Erol Osman K, Eksin Ibrahim (2006) A new optimization method: big bang-big crunch. Adv Eng Softw 37(2):106–111

33. Abedinpourshotorban Hosein, Shamsuddin Siti Mariyam, Beheshti Zahra, Jawawi Dayang NA (2016) Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. Swarm Evolut Comput 26:8–22

34. Kaveh A, Bakhshpoori T (2016) Water evaporation optimization: a novel physically inspired optimization algorithm. Comput & Struct 167:69–85

35. Mirjalili Seyedali, Mirjalili Seyed Mohammad, Hatamlou Abdolreza (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513

36. Kaveh A, Dadras A (2017) A novel meta-heuristic optimization algorithm: thermal exchange optimization. Adv Eng Softw 110:69–84

37. Irizarry Roberto (2004) Lares: an artificial chemical process approach for optimization. Evolut Comput 12(4):435–459

38. Lam Albert YS, Li Victor OK (2009) Chemical-reaction-inspired metaheuristic for optimization. IEEE Transactions Evolut Comput 14(3):381–399

39. Alatas Bilal (2011) Acroa: artificial chemical reaction optimization algorithm for global optimization. Expert Syst Appl 38(10):13170–13180

40. Abdechiri Marjan, Meybodi Mohammad Reza, Bahrami Helena (2013) Gases brownian motion optimization: an algorithm for optimization (gbmo). Appl Soft Comput 13(5):2932–2946

41. Salmani MH, Eshghi K (2017) A metaheuristic algorithm based on chemotherapy science: Csa. J Optim 2017: 3082024

42. Geem Zong Woo, Kim Joong Hoon, Loganathan Gobichettipalayam Vasudevan (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68

43. Al-Betar MA, Alyasseri Zaid Abdi A, Awadallah Mohammed A, Doush Iyad A (2021) Coronavirus herd immunity optimizer (CHIO). Neural Comput Appl 33:5011–5042

44. Tan Y, Zhu Y (2010) Fireworks algorithm for optimization. In: International conference in swarm intelligence, pages 355–364. Springer

45. Venkata Rao R, Savsani Vimal J, Vakharia DP (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput-Aided Des 43(3):303–315

46. Fadakar E, Ebrahimi M (2016) A new metaheuristic football game inspired algorithm. In: 2016 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC), pages 6–11. IEEE

47. Warid Warid, Hizam Hashim, Mariun Norman, Abdul-Wahab Noor Izzri (2016) Optimal power flow using the jaya algorithm. Energies 9(9):678

48. Warid W (2020) Optimal power flow using the amtpg-jaya algorithm. Appl Soft Comput 91:106252

49. Luu The Vinh, Nguyen Ngoc Son (2020) Parameters extraction of solar cells using modified jaya algorithm. Optik 203:164034

50. Congcong Wu, He Yichao (2020) Solving the set-union knapsack problem by a novel hybrid jaya algorithm. Soft Comput 24(3):1883–1902

51. Amarendhar Reddy M, Ravindranath K (2020) Virtual machine placement using jaya optimization algorithm. Appl Artif Intell 34(1):31–46

52. Maharana D, Kotecha P (2019) Optimization of job shop scheduling problem with grey wolf optimizer and jaya algorithm. Smart Innovations in Communication and Computational Sciences. Springer, New York, pp 47–58

53. Li J-Q, Deng J-W, Li C-Y, Han Y-Y, Tian J, Zhang B, Wang C-G (2020) An improved jaya algorithm for solving the flexible job shop scheduling problem with transportation and setup times. Knowl-Based Syst 200:106032

54. Mishra A, Shrivastava D (2020) A discrete jaya algorithm for permutation flow-shop scheduling problem. Int J Indus Eng Comput 11(3):415–428

55. Ghavidel Sahand, Azizivahed Ali, Li Li (2018) A hybrid jaya algorithm for reliability-redundancy allocation problems. Eng Optimization 50(4):698–715

56. WalaaH El-Ashmawi, AhmedF Ali, Slowik A (2020) An improved jaya algorithm with a modified swap operator for solving team formation problem. Soft Comput 24:16627–16641

57. Degertekin SO, Lamberti L, Ugur IB (2018) Sizing, layout and topology design optimization of truss structures using the jaya algorithm. Appl Soft Comput 70:903–928

58. Wang Shui-Hua, Phillips Preetha, Dong Zheng-Chao, Zhang Yu-Dong (2018) Intelligent facial emotion recognition based on stationary wavelet entropy and jaya algorithm. Neurocomputing 272:668–676

59. MohammedA Awadallah, MohammedA Al-Betar, Hammouri AI, Alomari OA (2020) Binary jaya algorithm with adaptive mutation for feature selection. Arab J Sci Eng 45:10875–10890

60. More Kiran C, Rao RV (2020) Design optimization of plate-fin heat exchanger by using modified jaya algorithm. Advanced engineering optimization through intelligent techniques. Springer, NewYork, pp 165–172

61. Wang Long, Zhang Zijun, Huang Chao, Tsui Kwok Leung (2018) A gpu-accelerated parallel jaya algorithm for efficiently estimating li-ion battery model parameters. Appl Soft Comput 65:12–20

62. Aslan Murat, Gunduz Mesut, Kiran Mustafa Servet (2019) Jayax: jaya algorithm with xor operator for binary optimization. Appl Soft Comput 82:105576

63. Rao RV, More KC (2017) Design optimization and analysis of selected thermal devices using self-adaptive jaya algorithm. Energy Convers Manag 140:24–35

64. Ravipudi Jaya Lakshmi, Neebha Mary (2018) Synthesis of linear antenna arrays using jaya, self-adaptive jaya and chaotic jaya algorithms. AEU-Int J Electron Commun 92:54–63

65. Raut Usharani, Mishra Sivkumar (2019) An improved elitist-jaya algorithm for simultaneous network reconfiguration and dg allocation in power distribution systems. Renew Energy Focus 30:92–106

66. Venkata Rao R, Saroj Ankit (2019) An elitism-based self-adaptive multi-population jaya algorithm and its applications. Soft Comput 23(12):4383–4406

67. Venkata Rao R, Saroj Ankit, Bhattacharyya Souvik (2018) Design optimization of heat pipes using elitism-based self-adaptive multipopulation jaya algorithm. J Thermophys Heat Transfer 32(3):702–712

68. Jian Xianzhong, Weng Zhiyuan (2020) A logistic chaotic jaya algorithm for parameters identification of photovoltaic cell and module models. Optik 203:164041

69. Son Nguyen N, Chinh Tran M, Anh Ho Pham H (2020) Uncertain nonlinear system identification using jaya-based adaptive neural network. Soft Comput 24:17123–17132

70. Ramesh S, Vydeki D (2020) Recognition and classification of paddy leaf diseases using optimized deep neural network with jaya algorithm. Information Process Agric 7(2):249–260

71. Son Nguyen Ngoc, Van Kien Cao, Anh Ho Pham Huy (2020) Parameters identification of bouc-wen hysteresis model for piezoelectric actuators using hybrid adaptive differential evolution and jaya algorithm. Eng Appl Artif Intell 87:103317

72. Goudos Sotirios K, Deruyck Margot, Plets David, Martens Luc, Psannis Kostas E, Sarigiannidis Panagiotis, Joseph Wout (2019) A novel design approach for 5g massive mimo and nb-iot green networks using a hybrid jaya-differential evolution algorithm. IEEE Access 7:105687–105700

73. Ding Zhenghao, Li Jun, Hao Hong (2020) Non-probabilistic method to consider uncertainties in structural damage identification based on hybrid jaya and tree seeds algorithm. Eng Struct 220:110925

74. Peng Yuexi, Sun Kehui, He Shaobo (2020) Synchronization for the integer-order and fractional-order chaotic maps based on parameter estimation with jaya-ipso algorithm. Eur Phys J Plus 135(3):331

75. Alotaibi SS (2020) Optimization insisted watermarking model: hybrid firefly and jaya algorithm for video copyright protection. Soft Comput 24:14809–14823

76. Azizi Mahdi, Ghasemi Seyyed Arash Mousavi, Ejlali Reza Goli, Talatahari Siamak (2020) Optimum design of fuzzy controller using hybrid ant lion optimizer and jaya algorithm. Artif Intell Rev 53(3):1553–1584

77. Luo Xiong, Cao Longpeng, Wang Long, Zhao Zihan, Huang Chao (2018) Parameter identification of the photovoltaic cell model with a hybrid jaya-nm algorithm. Optik 171:200–203

78. Ingle Kishor KisanKishor Kisan, Jatoth Ravi Kumar (2020) An efficient jaya algorithm with levy flight for non-linear channel equalization. Expert Syst Appl 145:112970

79. Rao RV, More KCmKCm, Taler J, Oclon P (2016) Dimensional optimization of a micro-channel heat sink using jaya algorithm. Appl Therm Eng 103:572–582

80. Morrison Ronald W, Jong Kenneth A De (2001) Measurement of population diversity. In: International Conference on Artificial Evolution (Evolution Artificielle). Springer, pp 31–41

81. Pandey Hari M (2016) Jaya a novel optimization algorithm: what, how and why? In: 2016 6th International Conference-Cloud System and Big Data Engineering (Confluence), IEEE, pp 728–730.

82. Li Yang, Yang Zhen (2017) Application of eos-elm with binary jaya-based feature selection to real-time transient stability assessment using pmu data. IEEE Access 5:23092–23101

83. Thirumoorthy K, Muneeswaran K (2020) Optimal feature subset selection using hybrid binary jaya optimization algorithm for text classification. Sadhana 45(1):1–13

84. Yang Z, Guo Y, Niu Q, Ma H, Zhou Y, Zhang L (2018) A novel binary jaya optimization for economic/emission unit commitment. In: 2018 IEEE Congress on Evolutionary Computation (CEC), 1–6

85. Degertekin SO, Lamberti L, Ugur IB (2019) Discrete sizing/layout/topology optimization of truss structures with an advanced jaya algorithm. Appl Soft Comput 79:363–390

86. Gao Kaizhou, Yang Fajun, Zhou MengChu, Pan Quanke, Suganthan Ponnuthurai Nagaratnam (2018) Flexible job-shop rescheduling for new job insertion by using discrete jaya algorithm. IEEE Transactions Cybern 49(5):1944–1955

87. Singh P, Chaudhary H (2019) Optimum discrete balancing of the threshing drum using jaya algorithm. Mech Based Des Struct Mach 1–15. https://doi.org/10.1080/15397734.2019.1701489

88. Singh Prem, Chaudhary Himanshu (2018) A modified jaya algorithm for mixed-variable optimization problems. J Intell Syst 29(1):1007–1027

89. Jiangtao Yu, Kim Chang-Hwan, Wadood Abdul, Khurshiad Tahir, Rhee Sang-Bong (2018) A novel multi-population based chaotic jaya algorithm with application in solving economic load dispatch problems. Energies 11(8):1946

90. Chen Feixin, Ding Zhenghao, Zhongrong Lu, Zeng Xiangkun (2018) Parameters identification for chaotic systems based on a modified jaya algorithm. Nonlinear Dyn 94(4):2307–2326

91. Farah Anouar, Belazi Akram (2018) A novel chaotic jaya algorithm for unconstrained numerical optimization. Nonlinear Dyn 93(3):1451–1480

92. Rao RV, More KC (2017) Optimal design and analysis of mechanical draft cooling tower using improved jaya algorithm. Int J Refrigeration 82:312–324

93. Kumar Abhishek V, Kumar Rakesh, Datta Saurav, Mahapatra Siba Sankar (2019) An integrated multi-response optimisation route combining principal component analysis, fuzzy inference system, nonlinear regression and jaya algorithm: a case experimental study on machining of gfrp (epoxy) composites. Int J Indus Syst Eng 32(4):497–525

94. Venkata Rao R, Saroj Ankit (2017) Constrained economic optimization of shell-and-tube heat exchangers using elitist-jaya algorithm. Energy 128:785–800

95. Venkata Rao R, Saroj Ankit (2018) Multi-objective design optimization of heat exchangers using elitist-jaya algorithm. Energy Syst 9(2):305–341

96. Venkata Rao R, Keesari Hameer Singh (2018) Multi-team perturbation guiding jaya algorithm for optimization of wind farm layout. Appl Soft Comput 71:800–815

97. Venkata Rao R, Keesari Hameer Singh, Oclon Pawel, Taler Jan (2020) An adaptive multi-team perturbation-guiding jaya algorithm for optimization and its applications. Eng Comput 36(1):391–419

98. Venkata Rao R, Keesari Hameer Singh, Oclon P, Taler Jan (2019) Improved multi-objective jaya optimization algorithm for a solar dish stirling engine. J Renew Sustain Energy 11(2):025903

99. Kunjie Yu, Boyang Qu, Yue Caitong, Ge Shilei, Chen Xu, Liang Jing (2019) A performance-guided jaya algorithm for parameters identification of photovoltaic cell and module. Appl Energy 237:241–257

100. Elattar Ehab E, ElSayed Salah K (2019) Modified jaya algorithm for optimal power flow incorporating renewable energy sources considering the cost, emission, power loss and voltage profile improvement. Energy 178:598–609

101. Chakraborty Uday K (2020) Semi-steady-state jaya algorithm for optimization. Appl Sci 10(15):5388

102. Mane S, Narsingrao M, Patil V (2018) A many-objective jaya algorithm for many-objective optimization problems. Decis Sci Lett 7(4):567–582

103. Jiangtao Yu, Kim Chang-Hwan, Rhee Sang-Bong (2019) Oppositional jaya algorithm with distance-adaptive coefficient in solving directional over current relays coordination problem. IEEE Access 7:150729–150742

104. Kaur Kanwarpreet, Walia Gurjot Kaur, Kaur Jaspreet (2018) Neural network ensemble and jaya algorithm based diagnosis of brain tumor using mri images. J Inst Eng (India): Series B 99(5):509–517

105. Bansal Mukesh, Agrawal C (2018) Jaya algorithm and artificial neural network based approach for object-oriented software quality analysis. Int J Intell Eng Syst 11(4):275–282

106. Khatir S, Boutchicha D, Le Thanh C, Tran-Ngoc H, Nguyen TN, Abdel-Wahab M (2020) Improved ann technique combined with jaya algorithm for crack identification in plates using xiga and experimental analysis. Theor Appl Fract Mech 107:102554

107. Sen Ruma, Choudhuri Bikash, Barma John Deb, Chakraborti Prasun (2018) Experimental investigation and optimisation of

wedm process for machining maraging steel using neural network based jaya algorithm. Int J Mach Mach Mater 20(4):387–399

108. Baliarsingh Santos Kumar, Vipsita Swati, Dash Bodhisattva (2020) A new optimal gene selection approach for cancer classification using enhanced jaya-based forest optimization algorithm. Neural Comput Appl 32(12):8599–8616

109. Ezzat D, Amin S, Shedeed Howida A, Tolba Mohamed F (2020) Directed jaya algorithm for delivering nano-robots to cancer area. Comput Methods Biomech Biomed Eng 23(16):1306–1316

110. Gao Kaizhou, Yang Fajun, Li Junqing, Sang Hongyan, Luo Jianping (2020) Improved jaya algorithm for flexible job shop rescheduling problem. IEEE Access 8:86915–86922

111. Ding Zhenghao, Li Jun, Hao Hong (2019) Structural damage identification using improved jaya algorithm based on sparse regularization and bayesian inference. Mech Syst Signal Process 132:211–231

112. Nayak Deepak Ranjan, Zhang Yudong, Das Dibya Sundar, Panda Subinita (2019) Mjaya-elm: a jaya algorithm with mutation and extreme learning machine based approach for sensorineural hearing loss detection. Appl Soft Comput 83:105626

113. Rao RV, Saroj A (2018) Constrained economic optimization of shell-and-tube heat exchangers using a self-adaptive multipopulation elitist-jaya algorithm. J Therm Sci Eng Appl. 10(4):041001

114. Venkata Rao R, Saroj Ankit (2017) A self-adaptive multi-population based jaya algorithm for engineering optimization. Swarm Evolut Comput 37:1–26

115. Rao Ravipudi V, Saroj Ankit, Oclon Pawel, Taler Jan, Lakshmi Jaya (2018) A posteriori multiobjective self-adaptive multipopulation jaya algorithm for optimization of thermal devices and cycles. IEEE Access 7:4113–4134

116. Migallon Hector, Jimeno-Morenilla Antonio, Sanchez-Romero Jose-Luis, Rico H, Rao R VenkataR Venkata (2019) Multipopulation-based multi-level parallel enhanced jaya algorithms. J Supercomput 75(3):1697–1716

117. Migallon Hector, Jimeno-Morenilla Antonio, Sanchez-Romero Jose-Luis (2018) Parallel improvements of the jaya optimization algorithm. Appl Sci 8(5):819

118. Migallon H, Jimeno-Morenilla A, Sanchez-Romero JL, Belazi A (2020) Efficient parallel and fast convergence chaotic jaya algorithms. Swarm Evolut Comput 56:100698

119. Rao RV, More KC, Coelho LS, Mariani VC (2017) Multi-objective optimization of the stirling heat engine through self-adaptive jaya algorithm. J Renew Sustain Energy 9(3):033703

120. Venkata Rao R, Rai Dhiraj P, Balic Joze (2018) Optimization of abrasive waterjet machining process using multi-objective jaya algorithm. Mater Today: Proc 5(2):4930–4938

121. Venkata Rao R, Rai Dhiraj P, Balic Joze (2019) Multi-objective optimization of abrasive waterjet machining process using jaya algorithm and promethee method. J Intell Manuf 30(5):2101–2127

122. Venkata Rao R, Rai DP, Ramkumar J, Balic J (2016) A new multi-objective jaya algorithm for optimization of modern machining processes. Adv Prod Eng & Manag 11(4):271

123. Mishra Aseem K, Shrivastava D, Bundela B, Sircar S (2020) An efficient jaya algorithm for multi-objective permutation flow shop scheduling problem. Advanced engineering optimization through intelligent techniques. Springer, Singapore, pp 113–125

124. Berrouk F, Bouchekara HREH, Chaib AE, Abido MA, Bounaya K, Javaid MS (2018) A new multi-objective jaya algorithm for solving the optimal power flow problem. J Electrical Syst 14(3):165–181

125. Rao Ravipudi Venkata, Saroj Ankit, Oclon Pawel, Taler Jan, Taler Dawid (2018) Single-and multi-objective design optimization of plate-fin heat exchangers using jaya algorithm. Heat Transfer Eng 39(13–14):1201–1216

126. Rao Ravipudi V (2019) Single-and multi-objective optimization of casting processes using jaya algorithm and its variants. Jaya: an advanced optimization algorithm and its engineering applications. Springer, Cham, pp 273–289

127. Srinivasarthnam C, Yammani Chandrasekhar, Maheswarapu Sydulu (2019) Multi-objective jaya algorithm for optimal scheduling of dgs in distribution system sectionalized into multi-microgrids. Smart Sci 7(1):59–78

128. Venkaiah C, Jain Rohan V (2017) Multi-objective jaya algorithm based optimal location and sizing of distributed generation in a radial distribution system. In 2017 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), pages 1–6. IEEE

129. Barakat AF, El-Sehiemy Ragab A, Elsayd Mohamed I, Osman E (2019) An enhanced jaya optimization algorithm (ejoa) for solving multi-objective orpd problem. In: 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), pp 479–484. IEEE

130. Wang S-H, Muhammad K, Lv Y, Sui Y, Han L, Zhang Y-D (2018) Identification of alcoholism based on wavelet renyi entropy and three-segment encoded jaya algorithm. Complexity 2018:3198184

131. Satapathy SC, Rajinikanth V (2018) Jaya algorithm guided procedure to segment tumor from brain mri. J Optim 2018:3738049

132. Wang Long, Huang Chao (2018) A novel elite opposition-based jaya algorithm for parameter estimation of photovoltaic cell models. Optik 155:351–356

133. Gong C (2018) Jaya algorithm-optimized pid controller for avr system. In: International Conference on Intelligent and Interactive Systems and Applications. pp 382–393. Springer

134. Hari Mr Jadhav Vaibhav, Patil MA (2017) Optimization of fuzzy controller by jaya algorithm. Int Res J Eng Technol (IRJET) 4(7):2182–2186

135. Phulambrikar Sudhir (2016) Solving combined economic emission dispatch solution using jaya optimization algorithm approach. Int Res J Eng Tech 3:501–512

136. Das T, Roy R (2018) Optimal reactive power dispatch using jaya algorithm. In: 2018 Emerging trends in electronic devices and computational techniques (EDCT). IEEE pp 1–6

137. Mohanty Subhadarshini, Patra Prashanta Kumar, Ray Mitrabinda, Mohapatra Subasish (2019) An approach for load balancing in cloud computing using jaya algorithm. Int J Information Technol Web Eng (IJITWE) 14(1):27–41

138. Warid Warid, Hizam Hashim, Mariun Norman, Wahab Noor Izzri Abdul (2018) A novel quasi-oppositional modified jaya algorithm for multi-objective optimal power flow solution. Appl Soft Comput 65:360–373

139. Khatir SamirSamir, Wahab Magd Abdel (2019) Fast simulations for solving fracture mechanics inverse problems using pod-rbf xiga and jaya algorithm. Eng Fract Mech 205:285–300

140. Khatir Samir, Wahab Magd Abdel (2019) A computational approach for crack identification in plate structures using xfem, xiga, pso and jaya algorithm. Theoretical Appl Fract Mech 103:102240

141. Wang Long, Huang Chao, Huang Lingmiao (2018) Parameter estimation of the soil water retention curve model with jaya algorithm. Comput Electron Agric 151:349–353

142. Dinh-Cong Du, Vinh Ho-Huu, Trung Vo-Duy, Quyen Ngo-Thi Hong, Trung Nguyen-Thoi (2018) Efficiency of jaya algorithm for solving the optimization-based structural damage identification problem based on a hybrid objective function. Eng Optimization 50(8):1233–1251

143. Zamli Kamal Z, Alsewari Abdulrahman, Ahmed Bestoun S (2018) Multi-start jaya algorithm for software module clustering problem. Azerbaijan J High Perform Comput 1:87–112

144. Das H, Naik B, Behera HS (2020) A jaya algorithm based wrapper method for optimal feature selection in supervised classification. J King Saud University-Comput Information Sci 1–13. https://doi.org/10.1016/j.jksuci.2020.05.002

145. Wang Shui-Hua, Cheng Hong, Phillips Preetha, Zhang Yu-Dong (2018) Multiple sclerosis identification based on fractional fourier entropy and a modified jaya algorithm. Entropy 20(4):254

146. Zhang Yu-Dong, Zhao Guihu, Sun Junding, Xiaosheng Wu, Wang Zhi-Heng, Liu Hong-Min, Govindaraj Vishnu Varthanan, Zhan Tianmin, Li Jianwu (2018) Smart pathological brain detection by synthetic minority oversampling technique, extreme learning machine, and jaya algorithm. Multimedia Tools Appl 77(17):22629–22648

147. Rajinikanth V, Dey N, Kumar R, Panneerselvam J, Raja NSM (2019) Fetal head periphery extraction from ultrasound image using jaya algorithm and chan-vese segmentation. Procedia Comput Sci 152:66–73

148. Zhang Yudong, Yang Xiaojun, Cattani Carlo, Rao Ravipudi Venkata, Wang Shuihua, Phillips Preetha (2016) Tea category identification using a novel fractional fourier entropy and jaya algorithm. Entropy 18(3):77

149. Huang Chao, Wang Long, Yeung Ryan Shun-Cheung, Zhang Zijun, Chung Henry Shu-Hung, Bensoussan Alain (2017) A prediction model-guided jaya algorithm for the pv system maximum power point tracking. IEEE Transactions Sustain Energy 9(1):45–55

150. Kunjie Yu JJ, Liang BYQu, Chen Xu, Wang Heshan (2017) Parameters identification of photovoltaic models using an improved jaya optimization algorithm. Energy Convers Manag 150:742–753

151. Demircali Akif, Koroglu Selim (2020) Jaya algorithm-based energy management system for battery-and ultracapacitor-powered ultralight electric vehicle. Int J Energy Res 44(6):4977–4985

152. Wadood A, Farkoush SG, Khurshaid T, Yu J-T, Kim C-H, Rhee S-B (2019) Application of the jaya algorithm in solving the problem of the optimal coordination of overcurrent relays in single- and multi-loop distribution systems. Complexity 2019:5876318. https://doi.org/10.1155/2019/5876318

153. Patel Jignesh P, Patel H (2020) An improved optimal economic dispatch of wind-thermal power using jaya algorithm. J Electrical Eng 20:1–8

154. Shuhui Xu, Wang Yong, Wang Zhi (2019) Parameter estimation of proton exchange membrane fuel cells using eagle strategy based on jaya algorithm and nelder-mead simplex method. Energy 173:457–467

155. Khanh Pham Q, Anh HPH (2019) Advanced intelligent identification of pmsm parameter using modified jaya optimization algorithm. J of Sci Technol 8–16. https://doi.org/10.15625/vap.2019.0002

156. Kumar Abhishek V, Kumar Rakesh, Datta Saurav, Mahapatra Siba Sankar (2017) Application of jaya algorithm for the optimization of machining performance characteristics during the turning of cfrp (epoxy) composites: comparison with tlbo, ga, and ica. Eng Comput 33(3):457–475

157. Kumar Vijendra, Yadav SM (2018) Optimization of reservoir operation with a new approach in evolutionary computation using tlbo Algorithm and jaya algorithm. Water Resour Manag 32(13):4375–4391

158. Jimeno-Morenilla Antonio, Sanchez-Romero Jose-Luis, Migallon Hector, Mora-Mora H (2019) Jaya optimization algorithm with gpu acceleration. J Supercomput 75(3):1094–1106

159. Sain Debdoot, Swain Subrat Kumar, Kumar Tapas, Mishra Sudhansu Kumar (2020) Robust 2-dof fopid controller design for maglev system using jaya algorithm. IETE J Res 66(3):414–426

160. Singh Sugandh P, Prakash Tapan, Singh VP, Babu M Ganesh (2017) Analytic hierarchy process based automatic generation control of multi-area interconnected power system using jaya algorithm. Eng Appl Artif Intell 60:35–44

161. Debnath Manoj Kumar, Sinha Sayantan, Mallick Ranjan Kumar (2017) Application of fuzzy-pidf controller for automatic generation control using jaya algorithm. Int J Pure Appl Math 114(9):51–61

162. Sain Debdoot, Swain Subrat Kumar, Mishra Sudhansu Kumar (2018) Real time implementation of optimized i-pd controller for the magnetic levitation system using jaya algorithm. IFAC-Papers OnLine 51(1):106–111

163. Chakraborty Uday K (2019) Proton exchange membrane fuel cell stack design optimization using an improved jaya algorithm. Energies 12(16):3176

164. Oclon Pawel, Cisek Piotr, Rerak Monika, Dawid Taler R, Rao Venkata, Vallati Andrea, Pilarczyk Marcin (2018) Thermal performance optimization of the underground power cable system by using a modified jaya algorithm. Int J Therm Sci 123:162–180

165. Bhongade S, Parmar Vishnu Prasad (2020) Jaya algorithm-based optimized automatic generation control scheme for interconnected solar-Thermal power system. IUP J Electrical Electron Eng 13(1):29–41

166. Bhongade Sandeep (2018) Automatic generation control of two-area stthermal power system using jaya algorithm. Int J Smart Grid-ijSmartGrid 2(2):99–110

167. Jumani Touqeer A, Mustafa Mohd W, Hussain Z, Rasid Madihah Md, Saeed MS, Memon Mehran M, Khan I, Nisar KS (2020) Jaya optimization algorithm for transient response and stability enhancement of a fractional-order pid based automatic voltage regulator system. Alex Eng J 59(4):2429–2440

168. Chowdhury A, Roy R, Mandal KK, Bhattacharya S, Biswas P, Nandy S (2020) Comparison of improvement in technical, commercial and environmental benefits by optimal single-point, twin-point and triple-point deterministic pv-based dg injection by jaya algorithm. Comput Adv Commun Circuits Syst. Springer, Newyork, pp 25–36

169. Roshani GH, Karami A, Khazaei A, Olfateh A, Nazemi E, Omidi M (2018) Optimization of radioactive sources to achieve the highest precision in three-phase flow meters using jaya algorithm. Appl Radiat Isotopes 139:256–265

170. Venkata Rao R, Rai Dhiraj P (2017) Optimization of submerged arc welding process parameters using quasi-oppositional based jaya algorithm. J Mech Sci Technol 31(5):2513–2522

171. Anjaneyulu K, Venkatesh G (2021) Optimization of process parameters of magnetic abrasive finishing using jaya algorithm. Mater Today Proc 41:1035–1040

172. Mishra Aseem K, Shrivastava Divya, Vrat Prem (2020) An opportunistic group maintenance model for the multi-unit series system employing jaya algorithm. OPSEARCH 57(2):603–628

173. Venkata Rao R, Rai Dhiraj P (2017) Optimisation of welding processes using quasi-oppositional-based jaya algorithm. J Exp & Theoretical Artif Intell 29(5):1099–1117

174. T Ozturk Hasan, T Dede, E Turker (2020) Optimum design of reinforced concrete counterfort retaining walls using tlbo, jaya algorithm. Structures 25:285–296

175. Dede T, Grzywinski M, Rao RV (2020) Jaya: a new meta-heuristic algorithm for the optimization of braced dome structures. Advanced engineering optimization through intelligent techniques. Springer, Newyork, pp 13–20

176. Natrayan L, Kumar M Senthil (2020) Optimization of squeeze casting process parameters on aa2024/al2o3/sic/gr hybrid composite using taguchi and jaya algorithm. Int J Control Automation 13(2s):95–104

177. Dinh-Cong D, Vo-Duy T, Ho-Huu V, Nguyen-Thoi T (2019) Damage assessment in plate-like structures using a two-stage method based on modal strain energy change and jaya algorithm. Inverse Problems Sci Eng 27(2):166–189

178. Das Smruti Rekha, Mishra Debahuti, Rout Minakhi (2019) A hybridized elm using self-adaptive multi-population-based jaya algorithm for currency exchange prediction: an empirical assessment. Neural Comput Appl 31(11):7071–7094

179. Wolpert David H, Macready William G (1997) No free lunch theorems for optimization. IEEE Transactions Evolut Comput 1(1):67–82

180. Al-Betar Mohammed Azmi, Awadallah Mohammed A (2018) Island bat algorithm for optimization. Expert Syst Appli 107:126–145

181. Al-Betar Mohammed Azmi, Awadallah Mohammed A, Doush Iyad Abu, Hammouri Abdelaziz I, Mafarja Majdi, Alyasseri Zaid Abdi Alkareem (2019) Island flower pollination algorithm for global optimization. J Supercomput 75(8):5280–5323

182. Awadallah Mohammed A, Al-Betar Mohammed Azmi, Bolaji Asaju Laaro, Doush Iyad Abu, Hammouri Abdelaziz I, Mafarja Majdi (2020) Island artificial bee colony for global optimization. Soft Comput 24:13461–13487

183. Chao Lu, Gao Liang, Yi Jin (2018) Grey wolf optimizer with cellular topological structure. Expert Syst Appl 107:89–114

184. Al-Betar MA, Khader Ahamad T, Awadallah Mohammed A, Alawan MH, Zaqaibeh B (2013) Cellular harmony search for optimization problems. J Appl Math 2013:139464

185. Tardivo Maria Laura, Caymes-Scutari Paola, Bianchini German, Mendez-Garabetti Miguel (2017) Hierarchical parallel model for improving performance on differential evolution. Concurrency Comput: Pract Exp 29(10):e4087

186. Lim Ting Yee (2014) Structured population genetic algorithms: a literature survey. Artif Intell Rev 41(3):385–399

187. Eiben Agoston E, Hinterding Robert, Michalewicz Zbigniew (1999) Parameter control in evolutionary algorithms. IEEE Transactions Evolut Comput 3(2):124–141