Engineering Applications of Artificial Intelligence 90 (2020) 103479

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Differential Evolution: A review of more than two decades of research ☆

Bilal [a], Millie Pant [a,*], Hira Zaheer [b], Laura Garcia-Hernandez [c], Ajith Abraham [d]

[a] Department of Applied Science and Engineering, Indian Institute of Technology, Roorkee, 247667, India
[b] Department of Mathematics, Jamia Millia Islamia, New Delhi, 110025, India
[c] Area of Project Engineering, University of Cordoba, Spain
[d] Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, Auburn, WA 98071, USA

ARTICLE INFO

Keywords:
Meta-heuristics
Differential evolution
Mutation
Crossover
Selection

ABSTRACT

Since its inception in 1995, Differential Evolution (DE) has emerged as one of the most frequently used algorithms for solving complex optimization problems. Its flexibility and versatility have prompted several customized variants of DE for solving a variety of real life and test problems. The present study, surveys the near 25 years of existence of DE. In this extensive survey, 283 research articles have been covered and the journey of DE is shown through its basic aspects like population generation, mutation schemes, crossover schemes, variation in parameters and hybridized variants along with various successful applications of DE. This study also provides some key bibliometric indicators like highly cited papers having citations more than 500, publication trend since 1996, journal citations etc. The main aim of the present document is to serve as an extended summary of 25 years of existence of DE, intended for dissemination to interested parties. It is expected that the present survey would generate interest among the new users towards the philosophy of DE and would also guide the experience researchers.

1. Introduction

Optimization is a kind of the decision making, or more specifically, is one of the major quantitative tools in the machinery of decision making, where decisions have to be taken to optimize one or more objectives under some prescribed set of circumstances. It may be said that optimization problems are ubiquitous in nature as most of the real-life problems can be formulated in terms of optimization models, involving several criteria and objectives. It forms the core of various fields such as Mathematics, Computational Science, Operations Research, Engineering, Economics, Physics, and Biology, etc. (Boyd and Vandenberghe, 2004). Optimization is a wide area of research which prescribes a particular method for solving a particular class of problems like Linear Programming Problems (LPP), Integer Programming Problems (IPP), Quadratic Programming Problem (QPP), Non-convex optimization and many more. The difficulty, however, arises when it becomes difficult to identify the nature of the problem. Under such circumstances when the nature of the problem is not known a priori, it becomes very difficult to select an appropriate method for obtaining the solution. Researchers are therefore focusing on generic algorithms that can be exploited for solving a wide range of problems. Past few decades have witnessed a boom of such general-purpose algorithms that can be put under the umbrella term of Meta-heuristics, defined in the next subsection.

1.1. Meta-heuristic techniques

Literature does not provide a very clear definition of Meta-heuristics but they may be considered as high-level strategies for guiding the heuristics algorithms (Ibarra and Kim, 1977). These are the optimization techniques mainly based on function evaluation and make little or no use of the properties of objective functions and constraints. Meta-heuristics are thus problem independent techniques not taking advantage of any specificity of the problem. Classification of meta-heuristics techniques is explained below and is pictorially depicted with the help of a network graph given in Fig. 1.
Meta-heuristics may be segregated into two categories

1. *Neighborhood-based Algorithms:* These are local search algorithms that make use of the concept of neighborhood by keeping track of the neighborhood solutions. In each iteration, the solution moves towards its neighbor solution so as to improve the current objective function value. Two popular neighborhood algorithms are:
   (a) Simulated Annealing (1979)
   (b) Tabu Search (1989)

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.engappai.2020.103479.

* Corresponding author.
E-mail addresses: bilal25iitr@gmail.com (Bilal), millifpt@iitr.ac.in (M. Pant), hirazaheeriitr@gmail.com (H. Zaheer), ir1gahel@uco.es (L. Garcia-Hernandez), ajith.abraham@ieee.org (A. Abraham).

https://doi.org/10.1016/j.engappai.2020.103479
Received 12 October 2019; Received in revised form 6 December 2019; Accepted 6 January 2020
Available online 18 February 2020
0952-1976/

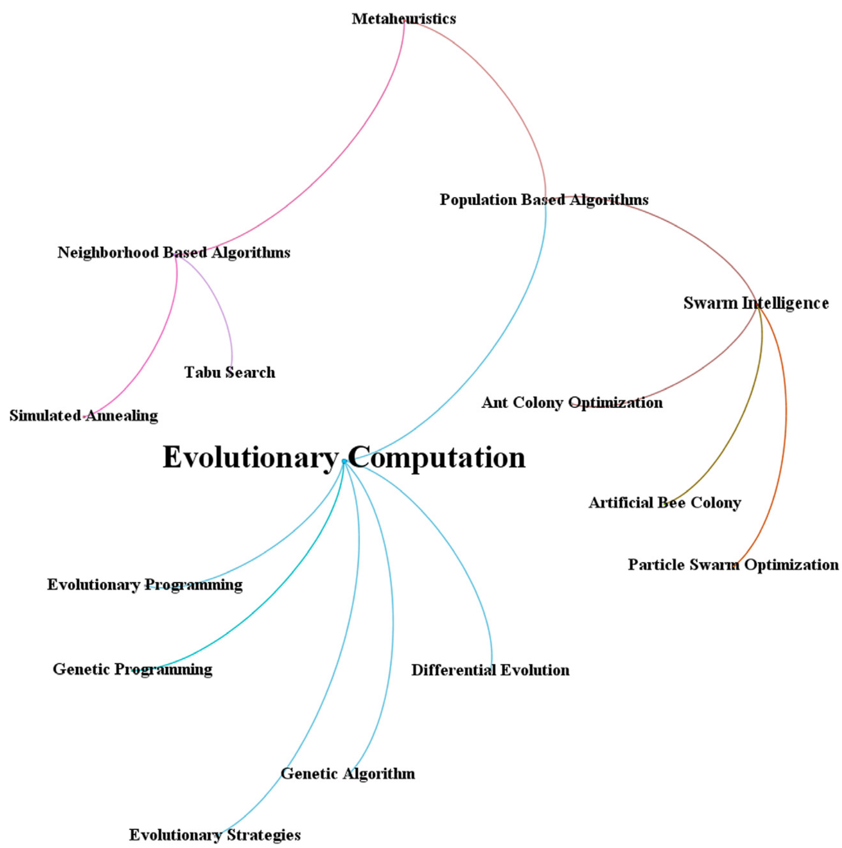


Fig. 1. Network of various meta-heuristics

2. *Population-based Algorithms:* These algorithms work on the set of solutions (or population), these are also referred to as nature-inspired algorithms as most of the algorithms are based on some natural analogy or philosophy. Population-based algorithms can be classified as:

(a) Swarm Intelligence (SI): These techniques are based on the socio-cooperative behavior displayed by various species in nature. These algorithms mostly follow the foraging actions of species. Popular algorithms based on swarm intelligence are as follows:

(i) Ant Colony Optimization (2006)
(ii) Particle Swarm Optimization (1995)
(iii) Artificial Bee Colony Optimization (2007)

(b) Evolutionary Algorithms (EA): These techniques follow the concepts of the theory of evolution of species. Some of the well-known evolutionary algorithms are:

(i) Evolutionary Programming (1960)
(ii) Genetic Algorithm (1957)
(iii) Genetic Programming (1992)
(iv) Evolutionary Strategies (1964)
(v) Differential Evolution (1995)

The algorithms mentioned above and many other such algorithms that are proposed from time to time have proved the mettle in solving complex and intricate problems that are otherwise difficult to solve through classical gradient-based methods. This focus of the present document is Differential Evolution (DE), an algorithm belonging to the class of evolutionary algorithms. The manuscript is divided into seven sections, opening with Section 1, which provides a brief introduction to the Meta-heuristic techniques available for solving optimization problems. Section 2, provides an introduction to DE, the core of the paper, also a performance assessment of benchmark problems. In Section 3, methodology used for preparing this survey is provided. Review of articles laying emphasis on selected aspects of DE is given Section 4. Section 5 provides the statistical analysis on the basis of different publishers (Elsevier, Springer, IEEE, Taylor & Francis Online etc.) Statistics based on journal rank by online platforms like incites (an online platform for ranking the Science citation index (SCI) articles). Section 6, provides an analysis and discussion on the status of DE since 1995 to 2018. Finally, future directions and conclusion are provided in Section 7 highlighting the key points.

2. Differential Evolution

Conceptualized by Storn (1996), the first documented article on DE appeared in the form of a technical report. A year later, the performance of DE was demonstrated at the First International Contest on Evolutionary Optimization in May 1996, which was held in conjunction with the 1996 IEEE International Conference on Evolutionary Computation (CEC) Storn and Price (1996). Since then DE has emerged to be a popular choice among researchers for solving optimization problems occurring in different domains. The total number of citations of DE since 1996 are recorded as 20366 till date as per Google Scholar citation and 7 of its variants has citations above 500 out of which 8 most cited variants are shown in Table 1. Fig. 2, provides the percentage citation of DE and its variants. Also, since 2005, many variants of DE have managed to obtain a position among the top three algorithms in the CEC competitions in successive years except for 2013, when DE obtained the 4th rank. The consistent performance of DE in different CEC competitions is shown in Table 2. Count of DE variant by Rank can be seen in Fig. 3.

Literature (Das and Suganthan, 2011; Das et al., 2016; Neri and Tirronen, 2010) reveals the emergence of DE as an effective and competitive member of the family of population based search algorithms such

Table 1
DE and its variants with citations above 500.

Variants	Year	Number of citation
Basic Differential Evolution (DE) (Storn and Price, 1997)	1996	20 366
Self-Adaptive Differential Evolution (SaDE) (Qin and Suganthan, 2005)	2005	2410
Adaptive Differential Evolution with Optional External Archive (JADE) (Zhang and Sanderson, 2009)	2009	1888
Opposition Based Differential Evolution (ODE) (Rahnamayan et al., 2008)	2008	1296
Neighborhood Based Differential Evolution (NDE) (Das et al., 2009)	2009	960
Composite Differential Evolution (CoDE) (Wang et al., 2011)	2011	898
Fuzzy Adaptive Differential Evolution (FADE) (Liu and Lampinen, 2005)	2005	857
Generalized Differential Evolution (GDE3) (Kukkonen and Lampinen, 2005)	2005	523

Table 2
Performance of DE variants in different CEC competition.

DE variant	CEC competition	Rank
SaDE (Qin and Suganthan, 2005)	CEC 2005	3rd
ε_DE (Takahama and Sakai, 2006)	CEC 2006	1st
GDE3 (Kukkonen and Lampinen, 2005)	CEC 2007	2nd
jDEdynNP-F (Brest et al., 2008)	CEC 2008	3rd
jDE (Brest et al., 2006)	CEC 2009	1st
εDEg (Takahama and Sakai, 2010)	CEC 2010	1st
DE-ACr (Reynoso-Meza et al., 2011)	CEC 2011	2nd
SHADE (Tanabe and Fukunaga, 2013)	CEC 2013	4th
L-SHADE (Tanabe and Fukunaga, 2014)	CEC 2014	1st
SPS-L-SHADE-EIG (Guo et al., 2015)	CEC 2015	1st
L-SHADE-Epsin (Awad et al., 2016a)	CEC 2016	1st
L-SHADE-cnEpsin (Awad et al., 2017a)	CEC 2017	3rd
L-SHADE-RSP (Akhmedova et al., 2018)	CEC 2018	2nd

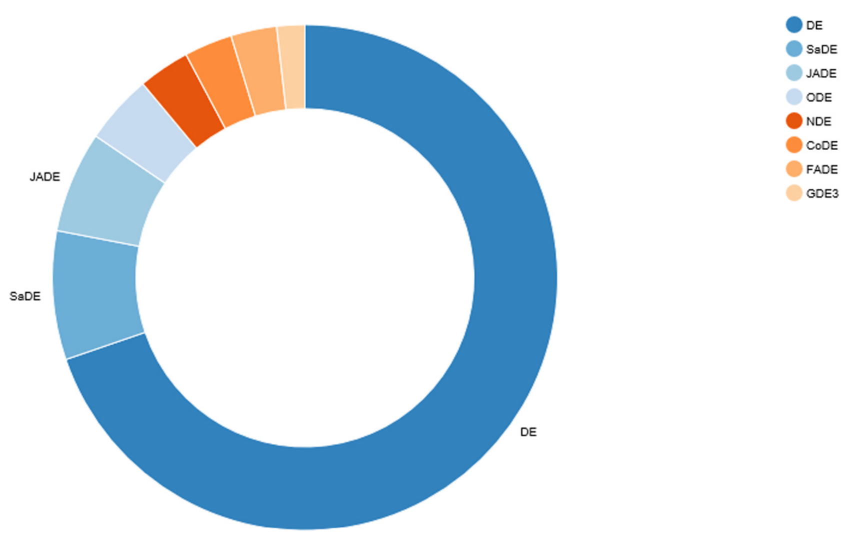


Fig. 2. Percentage citation in DE and its variants.

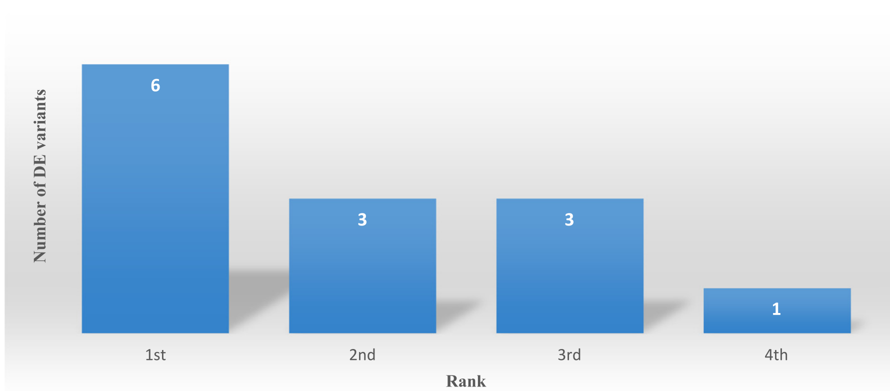


Fig. 3. Count of DE variant by rank.

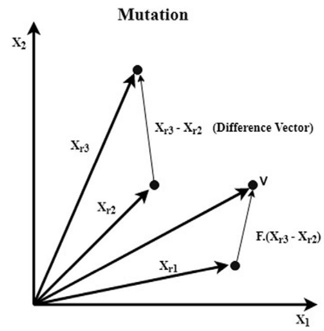


Fig. 4(a). Mutation scheme of DE.

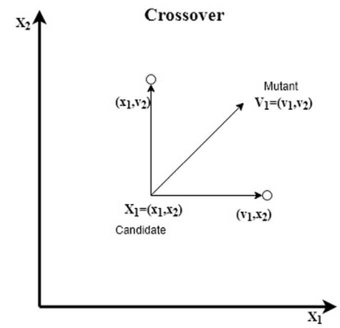


Fig. 4(b). Crossover scheme of DE.

as Genetic Algorithms (Goldberg, 1989), Evolution Strategies (Rochenberg, 1973), Evolutionary Programming (Fogel et al., 1965), Particle Swarm Optimization (PSO) (Eberhart and Kennedy, 1995), Firefly Algorithms(FA) (Yang, 2009).

2.1. Working of differential evolution

DE works in two phases: initialization and evolution. In first phase, population is generated randomly and in the second phase, which is evolution, the generated population goes through mutation, crossover and selection processes which are repeated until a termination criteria is met. Figs. 4(a)–4(c) shows the mutation, crossover and selection scheme of DE respectively.

(a) **Initialization:** During initialization, a set of uniformly distributed population is generated as follows: Let $S^G = \{X_j^G : j = 1, 2, \ldots, NP\}$ be the population at any generation G, NP denotes the size of population. Here, X_j^G denotes a D-dimensional vector as $X_j^G = \left\{x_{1,j}^G, x_{2,j}^G, \ldots, x_{D,j}^G\right\}$. X_j^G is generated using uniformly distributed random number $rand\,(0,1)$

$$X_j^G = X_{low} + \left(X_{upp} - X_{low}\right) * rand\,(0,1) \tag{1}$$

where X_{low}, X_{upp} are lower and upper bounds of search space S^G.
Once the initial population is generated, the next phase of evolution is activated.

(b) **Evolution:** This is the second phase where mutation, crossover and selection operations are performed.

Mutation: In mutation we generate a mutant vector V_j^G for each target vector X_j^G at generation G as

$$V_j^G = X_{r1}^G + F * \left(X_{r2}^G - X_{r3}^G\right) \tag{2}$$

where F is the scaling factor and value of is F is vary from 1 to 0 and $r1, r2, r3 \in \{1, 2, \ldots, NP\}$ are mutually different, randomly chosen vectors.

Crossover: After mutation, crossover is done to generate a new vector called trial vector denoted as $U_j^G = \left\{u_{1,j}^G, u_{2,j}^G, \ldots, u_{D,j}^G\right\}$. Crossover is performed between target vector $X_j^G = \left\{x_{1,j}^G, x_{2,j}^G, \ldots, x_{D,j}^G\right\}$ and mutant vector $V_j^G = \left\{v_{1,j}^G, v_{2,j}^G, \ldots, v_{D,j}^G\right\}$ using a crossover probability Cr whose value is between 01. U_j^G is generated as

$$u_{i,j}^G = \begin{cases} v_{i,j}^G \; if\; rand_j \le Cr \\ x_{i,j}^G \; otherwise \end{cases} \tag{3}$$

where $i \in \{1, 2, \ldots, D\}$ *and* $Cr \in [0,1]$.

Selection: In this operation, a comparison is done between the target vector and trial vector according to their fitness value. The one

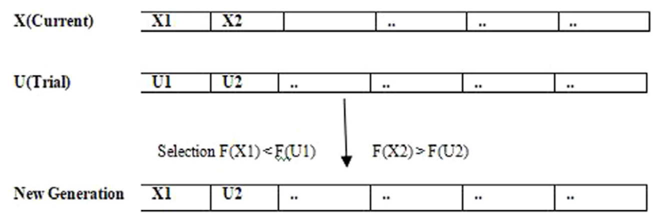


Fig. 4(c). Selection scheme of DE.

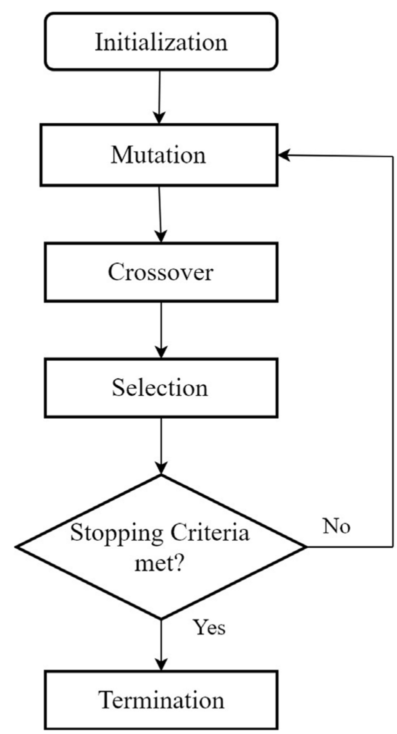


Fig. 5(a). Flowchart of DE.

having better fitness survives to the next generation. This operation is performed as:

$$X_j^{G+1} = \begin{cases} U_j^G \; iff \left(U_j^G\right) \le f\left(X_j^G\right) \\ X_j^G \; otherwise \end{cases} \tag{4}$$

Mutation, Crossover and selection of evolution phase are repeated till a predefined termination criterion is satisfied. Flowchart of DE and its Pseudo are provided in Figs. 5(a) and 5(b) respectively.

$Generate\ initial\ population\ of\ size\ NP$

$Do\ while$

$For\ each\ individual\ j\ in\ the\ population$

$Generate\ three\ random\ numbers\ r1 \neq r2 \neq r3 \neq j$

$For\ each\ parameter\ j, in\ the\ population$

$$X'_{i,j} = \begin{cases} X_{i,r3} + F * (X_{i,r2} - X_{i,r1}) & if\ rand_j \leq Cr \\ X_{i,j} & otherwise \end{cases}$$

$End\ for$

$Replace\ X_J\ with\ new\ particle\ X'_j, if X'_j is\ better$

$End\ For$

$Until\ the\ termination\ condition\ is\ satisfied$

Fig. 5(b). Pseudo code of DE.

2.2. Performance assessment

According to *"No free lunch theorem for optimization"*, all algorithms that search for an extremum of a cost function perform exactly the same when averaged over all possible cost functions. So, for any search/optimization algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class. Consequently, to prove the efficiency of a meta-heuristic algorithm over the other, many experiments are to be performed to reach a conclusion. Several performance metrics are available in the literature that can be used for evaluating the performance of a meta-heuristic algorithm like that of DE. In this section, we highlight some popular metrics:

Benchmark problems:

The first and foremost step to evaluate the performance of an optimization algorithm is to have an extensive set of test functions and benchmark problems based on properties like modality, scalability etc. Some standard benchmark problems include Ackley, Rosenbrock, Rastrigin, Sphere and many more. Selected benchmark functions along with their properties are provided in Appendix.

Stopping criteria:

Stopping criteria or quantifying "how and when" to stop the algorithm is another measure of evaluating the performance of an algorithm. Basically, stopping criteria provides an idea about the rate of convergence of an algorithm. Commonly used stopping criteria include (a) Number of runs, (b) Number of function evaluation (FE) (c) Terminating the algorithm before it reaches maximum function evaluation if the error in the function value is less than an absolute function error value.

Fitness value:

It is a numerical value used to rank solutions. Since the meta-heuristic algorithms are stochastic in nature, the values are observed in the form of mean, median, best, worst and standard deviation of the fitness values.

Algorithm complexity:

It is a rough approximation of the number of steps necessary to execute an algorithm (Suganthan et al., 2005). While evaluating complexity, one talks about the order of operation count, not of the exact count. Complexity can be measured as:

$$\frac{Algorithm\ complexity = T_2 - T_1}{T_1}$$

where,

$T_1 = \frac{\sum t_{1i}}{Number\ of\ function}$, t_{1i} is the computing time for 'A' evaluation of the problem.

$T_2 = \frac{\sum t_{2i}}{Number\ of\ function}$, t_{2i} is the complete computing time for the algorithm with 'B' evaluation of the problem i.

Other performance indicators

Some other indicators that are used for assessing the performance of an algorithm are:

a. Computation time.
b. Success Rate = (successful runs in maximum FE)/total runs.
c. Success Performance = mean (FEs for successful runs) ∗ (total runs)/(successful runs).
d. Graphical comparison of all optimization algorithms.

Statistical Analysis:

Statistical analysis is needed when even after comparing the algorithms based on above-mentioned performance metrics, no concrete conclusion can be made. Some well-known non-parametric models that can be used for analyzing and ranking the meta-heuristic algorithms are listed below (Bagdonavičius et al., 2013):

a. Friedman test.
b. Kruskal–Wallis test.
c. Spearman Rank Correlation.
d. Mann Whitney test.
e. Bonferroni test.
f. Jonckheere–Terpstra test.

3. Data collection and methodology

The paper provides an extensive review of DE algorithm by considering the research papers having following types of variants: population-based variants, mutation scheme based variants, crossover based variants, and selection based variants, hybrid variants of DE. Besides, discrete variants of DE are also discussed. Along with real life applications of DE in different areas of research and selected recent variants of DE. The search for research articles to be included in this paper is based on Journal Citation Report (JCR) (Thomson Reuters, 2018) until October, 2018.

Choice of keywords:

Differential Evolution + optimization, Differential Evolution + optimization + variants, Differential Evolution + mutation, Differential Evolution + crossover, Differential Evolution + selection, Differential Evolution + change in parameters, hybrid Differential Evolution, Differential Evolution + review (TITLE), Differential Evolution + survey (TITLE).

A total of 283 articles are reviewed, including 219 journal papers, 56 conference papers and 8 books/ edited books. The rank of articles are categories as per their quartile score (Q1, Q2, Q3, and Q4) using the online platform Incites (https://incites.clarivate.com/).

A pictorial representation of the methodology used in the paper is presented in Fig. 6.

4. Variants of differential evolution

DE has undergone a plethora of modifications since its conceptualization to further enhance its performance on complex optimization problems. The modifications are based on the parameters involve and the steps of DE i.e. population alignment, mutation, crossover, and selection.

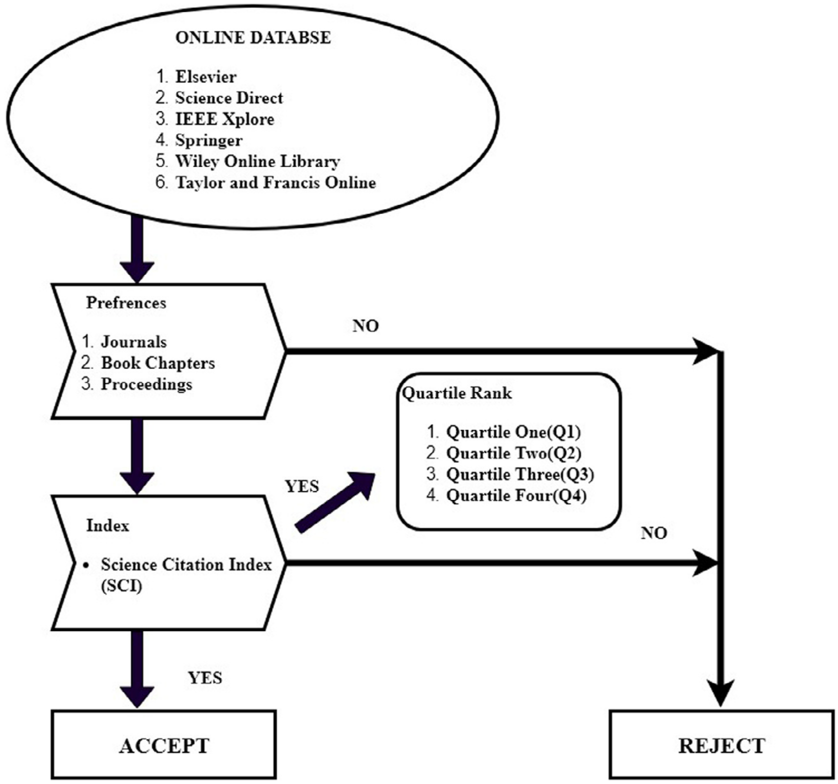


Fig. 6. Flowchart of database collection process.

4.1. Initialization techniques

In population-based search algorithms, population is initialized through computer generated random numbers which usually follow a uniform distribution. Though, this is a simple method for initializing the population, researchers observed that customizing the initialization process may help in improving the performance of the algorithm. Consequently, a variety of initialization methods have been proposed in literature. Mostly these modifications are based on either contracting the search space in the beginning itself to encourage faster convergence or are based on dividing the population into smaller subgroups of populations that can perform in parallel or tries to adaptively tune the population. Some interesting initialization methods are discussed in the following paragraph.

Rahnamayan et al. (2008) proposed an opposition based differential evolution, where an opposition based rule is used for initializing the population. Pant et al. (2009) used the quadratic interpolation method for initializing the population. Ali et al. (2009a) proposed a technique in which the population is initialized using the simplex method. Ali et al. (Tanabe and Fukunaga, 2013) proposed decentralization of initial population. Ali et al. (2013) also presented a compilation of different methods for population initialization. De Melo and Botazzo Delbem (2012) applied smart sampling for the initialization of the population. They made use of machine learning to identify the promising regions of the continuous search space. Ali et al. (De Melo and Botazzo Delbem, 2012) introduced a multi population-based DE for solving global optimization problems. They proposed to divide the population into different subgroups and used different mutation and update strategies to guide the search process and applied the proposed algorithm on large scale global optimization problems.

Researchers also worked on implementing an adaptive population size, where the size of the population varies as the algorithm progresses. The concept of adaptive population was first suggested by Teo in Teo (2005, 2006). Deng et al. (2013) developed an improved self-adaptive differential evolution algorithm with multiple strategies (ISDEMS) using a different search strategy and a parallel evolution mechanism. In the ISDEMS algorithm, the population is dynamically divided into multiple populations according to the fitness value of the individuals. Multiple strategies are used to improve the diversity of the individuals, to avoid premature convergence and to ensure efficiency in exchanging information among subpopulations. In Sun et al. (2012), authors have proposed a DE variant called adaptive population topology differential evolution algorithm (APTDE) for solving unconstrained problems. The authors showed that their method helps in avoiding premature convergence by updating the population topology adaptively.

Zhu et al. (2013) proposed an adaptive population tuning scheme (APTS) for DE for dynamically adjusting the population size based on ranking technique and perturbed the population to generate "fine" individuals. Sun (2017) proposed a novel symbiosis co-evolutionary model based on the population topology of DE, namely SCoPTDE, in which the population is divided into small species using specific topologies. Wang et al. (2016) introduced Cooperative Differential Evolution (CDE) for multi-objective function using multi-population strategy. Di Carlo et al. (2015) also used the multi-population technique. Aalto and Lampinen (2015) proposed an adaptive mechanism population-based DE called cumu-DE in which a probability mass function mechanism is used for automatically adapting the population size. Awad et al. (2017b) used niching based population reduction in a DE variant named sinusoidal differential evolution. Du Plessis and Engelbrecht (2012) proposed differential evolution using competitive population evolution dynamic environment. In Zhao et al. (2016), differential evolution with self-adaptive strategy and control parameters based on symmetric Latin hypercube design for unconstrained optimization problems in which the initial population is initialized by symmetric Latin hypercube design (SLHD) to increase the diversity.

4.2. Modifications in mutation scheme

Mutation is the most important step of DE as it produces a new individual in the population. A lot many researchers have worked towards this direction and have proposed modifications in mutation strategies. Yu et al. (2015) proposed a DE variant in which mutation becomes adaptive when the population clusters around the local optima. They also considered adaptive control parameters and validated the performance of the algorithm on eighteen benchmark problems. Islam et al. (2012) proposed a new mutation strategy with binomial crossover and two adaptive control parameters. In their work, a biased selection method is introduced where the mutant vector undergoes crossover with 'p' top-ranked population vectors, rather than the target vector. The authors validated the results on 25 CEC 2005 benchmark problems. Bujok and Tvrdík (2015) used current-to-pbest mutation strategies of JADE and proposed a modified strategy. They implemented the proposed algorithm on rotated benchmark problems. Mohamed (2015) proposed a triangular mutation approach based on the convex combination vector. He used three randomly selected vectors and calculated the difference between best and worst vector and combined it with basic mutation rule. Mezura-Montes and Cruz-ramírez (Ameca-Alducin et al., 2015) developed a method in which feasible solutions as reference are not required and is inspired by differential mutation. Authors combined two well-known DE variants: DE/rand/1/bin (base variant) and DE/best/1/bin (alternative variant) to propose a new mutation strategy.

In Zhang and Sanderson (2009), the authors proposed a new mutation strategy "DE/current-to-pbest" with optional external archive and adaptively updating control parameters. The DE/current-to- $p$best is a generalization of the classic "DE/current-to-best", while the optional archive operation utilizes historical data to provide information on progress direction. Both operations diversify the population and improve the convergence performance. In Mallipeddi and Suganthan (2010), an ensemble of mutation strategies and control parameters with the DE (EPSDE) is proposed. In EPSDE, a pool of distinct mutation strategies along with a pool of values for each control parameter coexists throughout the evolution process and competes to produce offspring solution. Sacco and Henderson (2014) proposed a new mutation operator for DE. The proposed operator was inspired by the topographical heuristic algorithm (TA) introduced in the early nineties (Törn and Viitanen, 1994), for solving global optimization problems. Sacco and Henderson named the new variant TopoMut-DE and used it for solving a highly multimodal nuclear reactor problem. Wei et al. (2015) proposed non-dominated sorting mutation operator based on fitness and diversity named. The corresponding algorithm was named MS-CDE and was used for solving constrained optimization problems. In this algorithm, the fitness of each individual is calculated and sorted and parents for mutation are selected on the basis of fitness as well as diversity.

Cui et al. (2016) developed a DE algorithm named as MPADE in which the population is divided into three parts based on the fitness values and three mutation strategies are applied on that for exploitation or exploration and further an adaptive technique is designed for parameter adjustment. MPADE was tested on 55 benchmark problems and 15 real-life problems. Some other modifications in mutation include: concept of donor mutation proposed by Fan et al. (2003) and Kumar et al. (2011a). Kaelo and Ali (2007) used the attraction–repulsion concept of electromagnetism to enhance mutation operation. Das et al. (2009)applied the concept of a neighborhood for each population member and used the DE/target-to- best/1/bin scheme of mutation in DE. Ali and Pant (2011) proposed a mutation operator following Cauchy distribution. In Lai et al. (2009), the authors proposed a wavelet-based mutation operation to control the scaling factor.

Some other modified mutations are multiple-deme based mutation (DEMDE), Epitropakis et al. (2011) proposed proximity-based mutation operators in which an attempt to efficiently guide the evolution of the population toward the global optimum, without sacrificing the search capabilities of the algorithm.

In Sun and Cai (2017), a novel DE algorithm with neighborhood-dependent mutation operator is proposed called neighborhood-dependent DE (NDE) in which a pool of population topologies is used to define multiple neighborhood relationships for each individual and then the neighborhood relationships are adaptively selected for the functions being solved during the evolutionary process and applied on the benchmark functions from CEC2013. In Opara and Arabas (2013), a censoring mutation is proposed and is used to guide and customize the search process. Liao et al. (2014) improved the performance of DE by using a ring topology based mutation strategy for faster convergence to attend the global solution.

Biswas et al. (2014) proposed a mutation operator named parent-centric mutation with normalized neighborhoods for inducing niching behavior in DE. Al-dabbagh et al. (2014) proposed a new variant of DE called DE-BEA with new mutation scheme called bacterial mutation scheme. In DE-BEA, the standard mutation scheme DE/current-to-rand/1/bin is replaced by DE/current-to-rand/bin scheme and results were validated on benchmark functions including several CEC 2005 test problems. Sharma and Pant (2011a) proposed SaMDE, a dynamically adaptive mutation step size scheme for DE and called SaMSDE. Sindhya et al. (2011) suggested a hybrid version of mutation operator for multi-objective optimization problems in which the standard mutation scheme is hybridized with polynomial based operator to improve the performance of differential evolution. Piotrowski (2013) proposed a memetic mutation operator based on global and local neighborhood. Choudhary et al. (2017), proposed a mutation operator with stochastic mutation factor inspired by levy flight random walk. Yu et al. (2018) proposed a mutation operator for constrained multi-objective problems. They showed that the proposed mechanism can produce well-distributed Pareto optimal front while satisfying the concerning constraints. Ortiz et al. (2013) proposed a new mutation scheme along with auto-adaptive control parameters and showed that the proposed algorithm can help in avoiding the stagnation of population points. Zhang and Yuen (2015) proposed directional mutation operator. Some other mutation strategies proposed in literature include: interpolation-based mutation (Kumar et al., 2012), intersect mutation operator (Zhou et al., 2013) and ranking based mutation (Gong and Cai, 2013).

4.3. Modified crossover schemes

As we already know, the crossover operator constructs a new trial vector with the help of a mutant vector. Initially, exponential crossover was proposed in the original work of Storn (1996) but later on it was mostly binomial variant that gained popularity among researchers (Storn and Price, 1997). An efficient comparative study of binomial and exponential crossover is given by Zaharie (2007). Zhao and Suganthan (2013) showed the success of exponential crossover in the high dimensional optimization problem; Gong et al. (2014) proposed a crossover rate *repair* technique for adaptive DE algorithms. In their scheme, the crossover rate in DE is *repaired* by its corresponding binary string, i.e. by using the average number of components taken from the mutant vector.

Pant et al. (2008a) proposed a parent centric crossover approach in which multiple parents recombines to produce the child and showed that the proposed algorithm works better in terms of convergence rate and robustness. Fister et al. (2016) gave Epistatic arithmetic crossover based on the Cartesian graph product. Ali (2007) proposed a crossover rule called the preferential crossover rule, to reduce the drawbacks of scaling parameters and applied their variant on 50 application problems. Fan and Zhang (2016) proposed self-adaptive differential evolution with adaptive crossover strategies. Wang et al. (2012) introduced an orthogonal crossover scheme and applied it on the variants of DE. Mukherjee et al. (2013) gave a locality-based crossover scheme for dynamic optimization problems. Guo and Yang (2015) proposed an eigenvector based crossover operator and showed that this concept can be applied to any crossover strategy. Zou and Gao (2012) modified crossover rate by using a linear increasing strategy. Qin and Suganthan (2005) proposed a self-adaptive Differential Evolution (SaDE), in which the generation of trial vectors and control parameter values are self-adaptive based on previous experiences.

4.4. Modifications done in selection scheme

DE has a unique selection mechanism that separates it out from the contemporary algorithms. Though modifications suggested in the selection scheme are limited to only a few papers, researchers have shown that suitable changes in it can further help in improving the performance of the algorithm.

Yi et al. (2016) proposed a fitness function value based p best selection mechanism. If the offspring is having better fitness value it means the p-best of that particular offspring is suitable for exploitation. By doing so, the population is not gathered near p-best, which results in the diversification of the population. Pan et al. (Gämperle et al., 2002) proposed each target vector to be associated with a different strategy list (SL), a scaling factor F list (FL) and a crossover rate R list (CRL). When a trial vector is generated F and CR are selected from strategy lists and if an obtained trial vector is better than target vector then F and CR will enter into the winning strategy list (wSL), a winning F (wFL) and winning crossover (wCRL) respectively. After certain number of iterations, F and CR values are updated by selecting elements from wFL, wCRL and wSL.

4.5. Changes in parameters

The four main parameters of DE are population size *NP*, the crossover rate C_r and the scaling factor *F*. *N*umerous studies have been done for finding out the proper setting of these parameters to get an optimal solution. Initially, Storn (1996) suggested that the population size of 5D–10D and the value of scaling factor as 0.5 are sufficient for obtaining an optimal solution, where D denotes the dimension of the problem. Later, Gämperle et al. (2002) recommended that choosing the population size between 3D to 8D, scaling factor as 0.6 and crossover rate in the interval [0.3–0.9] is a good choice. According to Storn and Price (1997) and Liu and Lampinen (2002), the value of F and C_r should be in between [0.5, 1] and [0.8, 1] respectively.

Ronkkonen et al. (2005) suggested that to obtain a good solution the value of F should lie in the interval [0.4, 0.95] and taking F as 0.9 initially, is a good choice. For the crossover probability, they suggested that if the objective function is separable then C_r should lie between [0, 0.2] whereas it should be in between [0.9, 1] if the function is not separable i.e if its parameters are dependent. Zielinski and Laur (2006) suggested that in many cases by setting F and Cr $\geq$ 0.6 good results can be obtained.

Zhao et al. (2016) presented a DE approach for unconstrained optimization problems with self-adaptive strategy and control parameters. To increase the diversity, the population is initialized through Latin hypercube design, trial vector is selected adaptively and F and CR values are updated through Cauchy and Normal Distribution.

In the approach of Lin et al. (2015), two parents are picked from the population to give a better evolutionary direction. The success rate of offspring and the scaling factor is made adaptive based on evolutionary progress. In Qin et al. (2009) Self-adaptive DE (SaDE) algorithm is proposed, in which both, the method of generation of trial vector and their associated control parameter values are moderately self-modified by learning from their previous experiences in improving solutions.

Ghosh et al. (2011) described a very simple and very flexible technique for tuning both F and Cr, on the run, without any user involvement. The adaptation strategy is based on the value of the objective function for different individuals in the DE population. Wang et al. (2013) introduced Gaussian bare-bones DE (GBDE) and its modified variants which were parameter-free. Meng et al. (2018) proposed Parameters with Adaptive Learning Mechanism called (PALM) in DE to tackle the inconvenience involved in selecting the control parameters.

Ortiz et al. (2013) proposed DE with auto-adaptive control parameters to overcome the stagnation of population near local minima. Brest et al. (2006) also used the concept of self-adaptive control parameters in DE and tested the proposed algorithm on numerical benchmark problems. Zamuda and Brest (2015) further used the self-adaptive mechanism of control parameters in DE to randomize the frequency and propagation. In Guo et al. (2014), proposed an improved Differential Evolution having self-adaptive control parameters based on simulated annealing. In their variant, firstly, two groups of the population are initialized. Secondly, it generates a set of control parameters for one of the two populations and then further creates another new series of control parameter for the other population through mutating the initial control parameters. Once the control parameters are generated, the two populations are mutated and crossed to produce two groups of trial vectors. Finally, the target vectors are selected from the two groups of trial vectors through selection operation. To enhance its global search capabilities, simulated annealing (SA) is involved in the selecting operation and the control parameters with better performance are chosen as the initial control parameters of the next generation.

4.6. Hybrid variants of different evolution

Hybridization is another important modification in DE which is implemented to enhance its performance and convergence speed. Plenty of work can be found in the literature on the hybridization of DE. The authors in El Dor et al. (2012) hybridized PSO with DE for enhancing the algorithm. Pant et al. (2008b) also proposed a hybrid version of DE with PSO and results show that the proposed DE-PSO is quite competent for solving the considered test functions as well as real-life problems. Liu et al. (2010) proposed a hybrid technique using DE with PSO for constrained optimization problems. In Long et al. (2013), direct search technique augmented Lagrangian method is hybridized with DE to solve constrained problems.

Zhang et al. (2009a) proposed hybrid variants where DE is hybridized with PSO for enhancing the exploration ability and also modified DE operators are incorporated for enhancing each particle's local tuning ability. Yan and Shi (2011) hybridized DE with PSO and Harmony Search Optimization and named the resulting algorithm HDPH. They validated the proposed hybrid approach on benchmark problems and reported that it gave promising results. Caponio et al. (2009) proposed DE variant named super-fit memetic differential evolution (SFMDE) in which DE is hybridized with PSO and two local search algorithms namely Nelder Mead and Rosenbrock. The efficiency of the algorithm was tested on for image processing problem and optimal control drive design for direct current (DC) motor. Zhang et al. (2017) hybridized cuckoo search and krill herd optimization with DE. Omran and Engelbrecht (2009) developed a hybridized DE named Free Search DE (FSDE), in which DE is hybridized with the Free Search Algorithm and opposition based learning algorithm.

Menchaca-Mendez and Coello (2009) hybridized DE with Nelder Mead algorithm and applied it for solving constrained optimization problems. Ali et al. (2009b) hybridized the Ant Colony Algorithm with DE and applied it on problems related to water resources. Liao (2010) proposed two hybrid DE variants, one hybridized with local search operator and other with harmony search algorithm for solving engineering design applications. In Dong et al. (2016), the authors hybridized DE with an estimation of distribution algorithm for locating the promising global solutions.

Cai et al. (2011) presented a hybrid DE algorithm called clustering-based DE (CDE), which is based on k-means clustering used for solving unconstrained global optimization problems. Sharma and Pant (2011b) hybridized Differential evolution with Artificial Bee Colony algorithm. The hybridization of DE and ABC was also reported in Stanarevic (2012). In Ustun and Akdagli (2017) also, the authors hybridized ABC with DE, and named it hybrid optimization method (HOM). Worasucheep (2015) used opposition based hybrid ABC and DE called OABCDE for solving the continuous optimization problems. The proposed OABCDE was tested on more the 15 benchmark problem with varying dimensions of size 30, 60 and 100. Zhang et al. (2013) incorporated the concepts of Lamarckian and Baldwinian learning into

DE, to improve its performance and used it for solving well placement problem. Sethanan and Pitakaso (2016a) applied three local search technique shifting, exchange (SWAP) and k-variable move algorithms into DE and used it for solving assignment problems. Li et al. (2011) hybridized DE with a local search operator simplified quadratic approximation(SQA) for the constrained optimization problem.

In Zhang et al. (2009b), the authors proposed a hybrid version of DE with simplified quadratic interpolation to optimize the performance of the antenna. In Yuan et al. (2008) chaos theory was applied to obtain self-adaptive parameter settings in differential evolution (DE) to solve short term hydrothermal system generation scheduling problems. Awad et al. (2017c) proposed CADE a hybrid version of the Cultural Algorithm (CA) and Differential Evolution (DE). In this algorithm both the meta-heuristics work in parallel.

4.7. Discrete differential evolution

Wang et al. (2010a) proposed a hybrid discrete DE algorithm to solve blocking job scheduling problems. The individuals are represented as discrete permutations and mutation and crossover are performed directly in the discrete domain and a neighborhood-based local search algorithm is embedded to maintain exploration and exploitation. Liu et al. (2009) proposed a discrete DE algorithm for solving job-shop scheduling problems, the floating values are encoded as discrete with the help of random key encoding scheme and then mutation, uniform crossover and selection are activated in the algorithm.

Sauer and Coelho (2008) worked for solving traveling salesman with DE. They used four methods for solving the problem, first the problem was solved using DE, in the second DE was hybridized using variable neighborhood search (VNS), in third DE was hybridized using Lin–Kernighan–Heulsgaun (LKH) method and in fourth, DE was hybridized with both VNS and LKH methods.

Zhang et al. (2008) proposed a discrete DE by applying the idea of JADE and suggested discrete encoding with the help of rank-based representation schema. Yuan et al. (2009) proposed a discrete binary differential evolution (DBDE) approach to solve the unit commitment problem (UCP) in which they used the method given by Gong and Tuson (2007) for binary encoding in mutation and crossover and selection are same as in basic DE. In Li-bao et al. (2016), the authors proposed a hybrid mutation scheme based discrete differential evolution for optimizing multidimensional knapsack problem.

In Pan et al. (2011), the authors proposed a hybrid discrete differential evolution (HDDE) algorithm to minimize the total make-span in a flow shop scheduling problem, applied job permutation-based mutation and crossover and hybridized the algorithm with a local search algorithm based on insert and swap neighborhood structures. He et al. (2009) proposed a discrete binary differential evolution(BDE) for feature selection, to select the best feature subset.

Cuevas et al. (2011) proposed a discrete differential evolution algorithm in which solutions are initialized as discrete points which are converted into real values using a forward transformation, mutation and crossover operations are performed after that, and the real values are converted back to integers. After this step, selection is performed and the algorithm is used in detecting circles with sub-pixels' accuracy in noisy images.

Datta and Figueira (2013), worked directly for discrete integer values. They kept the basic crossover and selection as it is but modified the mutation operation. During mutation, the random vectors can take three values 0 or 1 out of eight combinations; one is selected in each case. In Mahdavi (2017) authors proposed an enhancing discrete differential evolution for conducting election. Zeng et al. (2012) proposed a Pareto Utility Discrete Differential Evolutionary (PUDDE) based optimization model for solving operator allocation problems (OAP). The proposed algorithm is a combination of discrete event simulation (DES) and Pareto utility selection strategy in which the discrete value is handled with the help of two modified operators namely addition and subtraction. Fatih Tasgetiren et al. (2010) presented a discrete DE with a parallel population to solve the traveling salesman problem. In Tasgetiren et al. (2009) authors proposed Discrete DE to solve a single machine total weighted tardiness problem with sequence-dependent setup times. Deng and Gu (2012) formulate a hybrid DE for solving no-idle flow shop scheduling in which DE was hybridized with a neighborhood-based local search algorithm.

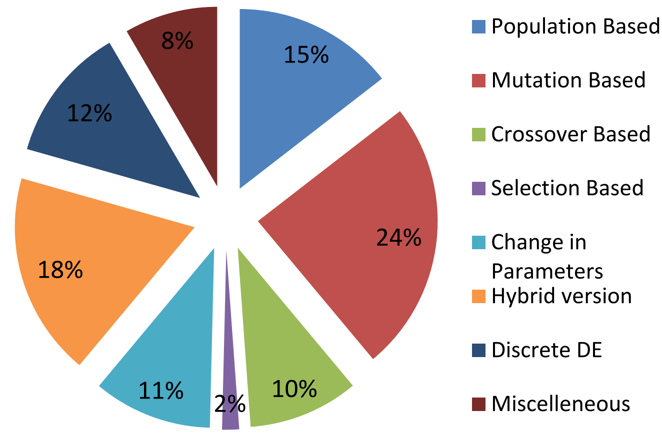


Fig. 7(a). Proportion of different DE variants.

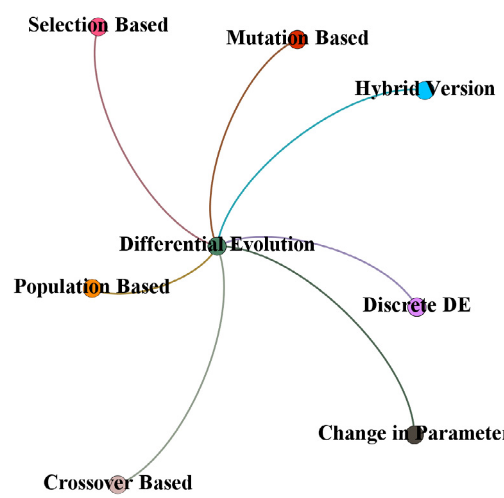


Fig. 7(b). Network graph of different DE variants.

4.8. Miscellaneous

Reed et al. (2013) gave the idea to adjust both mutation and crossover rates, during the optimization, in a manner that increases the convergence rate to the desired solution. Performance is demonstrated on a challenging problem of identifying imperfections in submerged shell structures.

In Zou et al. (2013) the authors proposed a modified DE (MDE) which employs Gaussian and uniform distribution to adjust scale factor and crossover rate. They assured that this modification helped in increasing the diversity of the population. Segura et al. (2014) tell about the relationship between success and failure of adaptive schemes and the balance between exploration and exploitation introduced by the formulation trial vector. Elsayed et al. (2013a) proposed a DE variant in which different mutation operators are used and also a covariance adaption matrix evolution strategy local search algorithm is incorporated in that. Parouha and Das (2016) incorporated the "pbest",

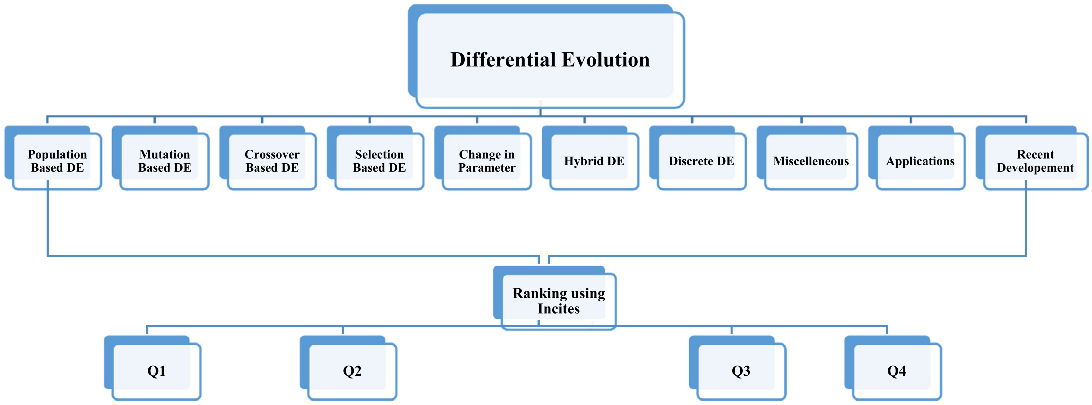


Fig. 8. Ranking of different DE variants.

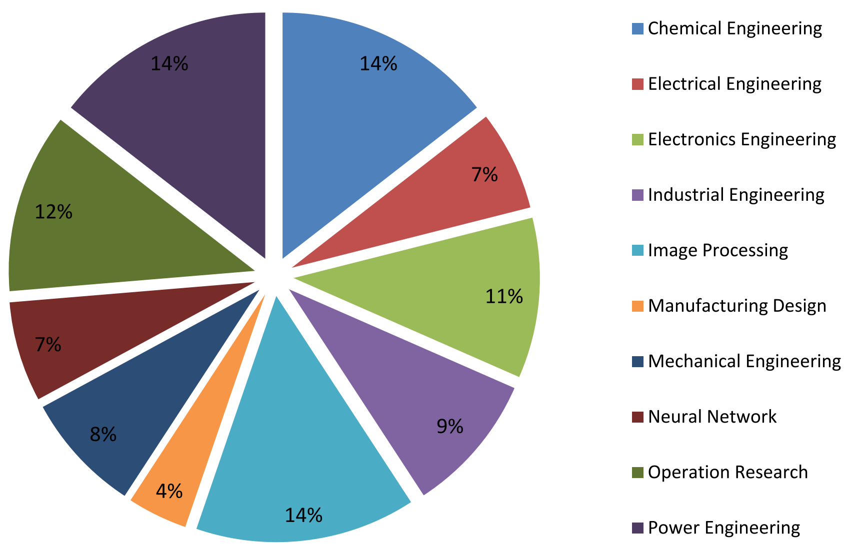


Fig. 9. Percentage usage of DE in different reviewed applications areas.

"gbest" (local best, global best) concept of PSO into DE and proposed a variant named Memory based DE (MBDE).

Wu et al. (2016) proposed multi population-based ensemble DE named as MPEDE in which the whole population is divided into three equally sized smaller subpopulation and one much larger population. Three different mutation strategies names as "current-to-pbest/1", "current-to-rand/1" and "rand/1" are used in the three populations. After some generations, the current best strategy is determined and the larger population is assigned to it. Also, the control parameters of each strategy are made adaptive.

Elsayed et al. (2013b) proposed two DE variants for solving constrained optimization problems. In this mutation and crossover operators are mixed and made adaptive by an adaptive learning process. In Salman et al. (2007), a self-adaptive DE is proposed in which the manual setting of the control parameter is eliminated. Ameca-Alducin et al. (2014) proposed a method to solve dynamic constrained optimization problems in which feasibility rule is applied for handling constraints. They combined two popular DE variants namely DE/rand/1/bin and DE/best/1/bin later applied hill-climbing local search algorithm for faster convergence. Guo et al. (2016) proposed a constrained min–max optimization algorithm by generating the method of escape vectors. Jiang and Qiang (2013) introduced DE with a base vector group. All individuals are ordered according to their fitness from high to low. Then a certain number of individuals in front are chosen to form a base vector group. For each target vector, its mutation base vector is chosen from the base vector group when the mutation is operated.

Fig. 7(a) Shows a pie graph of different DE variants and a network graph of DE with its variants can be seen in Fig. 7(b). Fig. 8 shows ranking of DE variants using incites.

4.9. Applications of differential evolution

DE has been applied to a wide range of problems arising in various fields of science, engineering, and management. It is very difficult to summarize all the applications at one place; therefore, a brief review of some of the major applications is shown. A literature survey of applications of DE is shown in the following Table 3. A Pie chart of number of applications of DE is provided in Fig. 9. Fig. 10 shows a network graph of DE applications.

Some of the screens capture of DE applications in different engineering applications are illustrated in Fig. 11.

4.10. Recent developments in differential evolution (2016–2018)

Awad et al. (2016b) introduced a differential stochastic fractal evolutionary algorithm (DSF-EA) with balancing the exploration or exploitation feature. They presented a three-stage optimization algorithm with differential evolution diffusion, success-based update process and dynamic reduction of population size. Maciel et al. (2016) applied DE to estimate the parameters of the Nelson–Siegel and Svensson functions using real data of US Treasury government bonds. Wang et al. (2017a) proposed a two-phase DE for Uniform Designs in Constrained

Table 3
Different application areas of DE.

Areas	Publications
Chemical engineering	Lampinen and Zelinka (1999), Kapadi and Gudi (2004), Rane et al. (2005), Babu and Angira (2006), Babu and Munawar (2007), Kumar et al. (2011a), Dos Santos et al. (2012), Silva et al. (2012), Li et al. (2013a), Sacco et al. (2013) and Mandal and Chakraborty (2013)
Electrical engineering	Georgilakis (2009), Uyar et al. (2011), Arya et al. (2012), Arya and Choube (2013) and Coelho et al. (2013)
Electronics engineering	Thangaraj et al. (2010), Baatar et al. (2013), Goudos et al. (2011), Islam et al. (2012), Li and Yin (2012), Baatar et al. (2013), Goudos et al. (2013) and Sotiroudis et al. (2013)
Industrial engineering	Babu and Angira (2008), Raj et al. (2008), Akrout et al. (2012), Raj et al. (2012), Yildiz (2013), Nwankwor et al. (2013), Kumar and Pant (2013) and Kadhar et al. (2015)
Image processing	Omran and Engelbrecht (2006), De Falco et al. (2008), Rahnamayan and Tizhoosh (2008), Das and Konar (2009), Cuevas et al. (2010), Maulik and Saha (2010), Kumar et al. (2011b), Sabat et al. (2011), Nakib et al. (2012), Lee and Cho (2012) and Mesejo et al. (2013)
Manufacturing design	Noktehdan et al. (2010), Zamuda et al. (2011) and Kao and Chen (2013)
Mechanical engineering	Joshi and Sanderson (1999), Coello Coello (2000), Liao (2010), Wang et al. (2010b), Zhang and Rangaiah (2012) and De Melo and Carosio (2013)
Neural network	Ilonen et al. (2003), Magoulas et al. (2004), Du et al. (2007), Subudhi and Jena (2008) and Dragoi et al. (2013)
Operation research	Onwubolu and Davendra (2006), Nearchou (2008), Qian et al. (2009), Adeyemo and Otieno (2010), Wang et al. (2010c), Hachicha et al. (2011), Wang and Xu (2011), Le Hai et al. (2012) and Reddy and Ravi (2013)
Power engineering	Zielinski et al. (2006), Chang et al. (2007), Noman and Iba (2008), Bhattacharya and Chattopadhyay (2010), Wang et al. (2010b), Zhang et al. (2010), Boussaïd et al. (2011), El Ela et al. (2011), Sivasubramani and Swarup (2012), Li et al. (2013b) and Srinivasa Reddy and Vaisakh (2013)

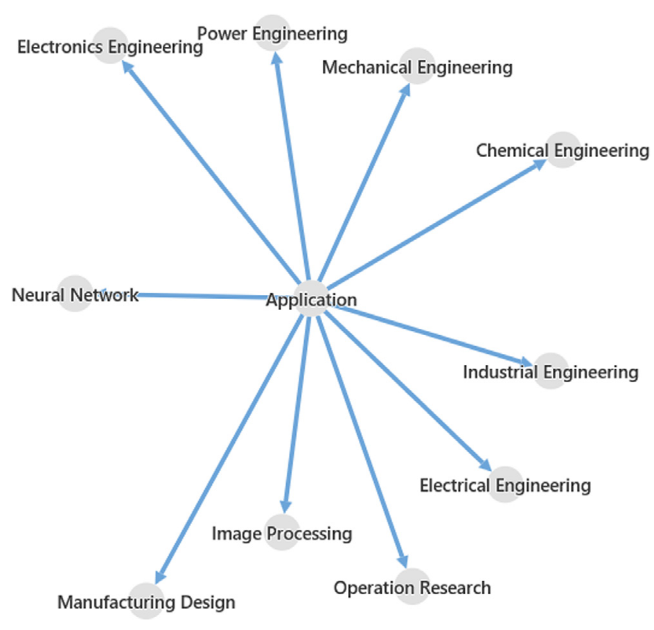


Fig. 10. Network graph of DE applications in different areas.

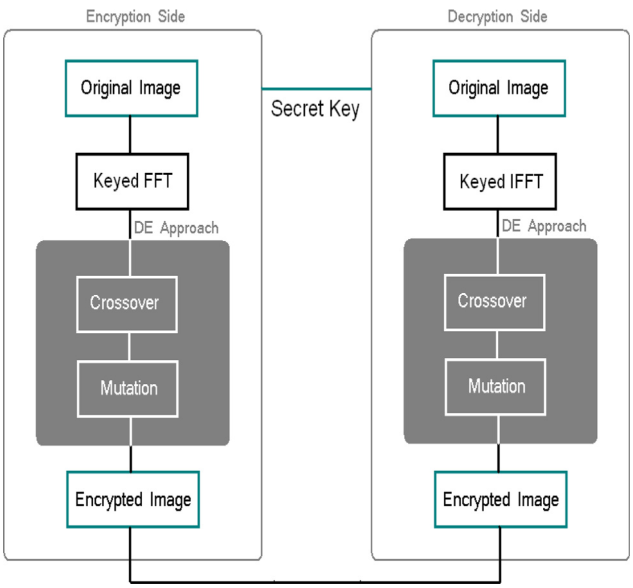
Experimental Domains Asafuddoula et al. (2015) introduced a new constraint handling technique in DE in which every solution is assigned a random sequence of constraints and the solutions are sorted on the basis of satisfied constraint and the violation measure. Sallam et al. (2017) proposed a new two-phase (exploration and exploitation) multi-operator DE algorithm. It starts with the exploration phase, dynamically placing emphasis on the best-performing DE based on two landscape indicators and its performance history. This process is repeated during the exploitation phase. They applied their algorithm for solving many real-world problems.

Cheng et al. (2016) proposed a DE variant for on multi-objective optimization problems. In the proposed algorithm, the objective space is divided into grids based on non-dominated solutions in the population. Based on these grids, three indexes are defined which include grid fitness, grid density, and grid objective-wise standard deviation to measure rank, density, and population search status respectively of individuals.

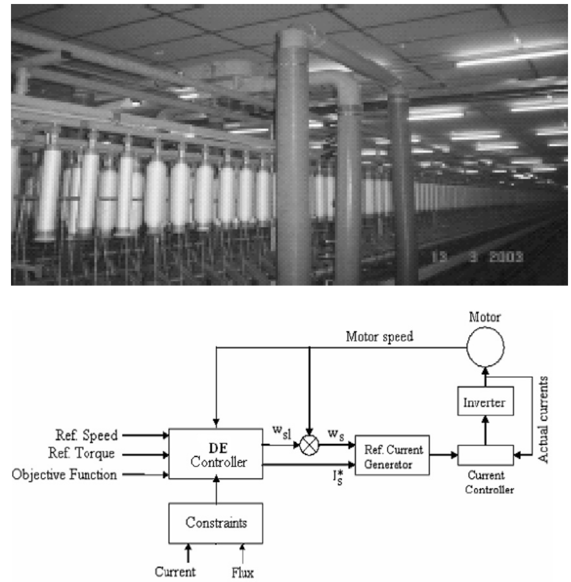
ZorarpacI and Özel (2016) hybridized DE with ABC for feature selection of classification and evaluated the performance of the proposed algorithm on a dataset from the UCI Repository. Cai et al. (2018) proposed social learning DE, inspired by the social learning behavior of the animal kingdom. The strategy was named adaptive social learning (ASL). In this strategy, each individual is only allowed to interact with its neighbors and the parents for mutation will be selected from its neighboring solutions.

In Tian and Gao (2018), the authors tried to improve the performance of DE by designing a stochastic mixed mutation strategy and an information intercrossing and sharing mechanism. Mlakar et al. (2016) proposed a hybrid differential evolution by using a switching probability strategy of Cuckoo search and applied it for multi-level image thresholding. Ali et al. (2016) also combined two heuristic search techniques in which DE is used as population space for the cultural algorithm. Do et al. (2016) proposed a novel DE approach in which scaling factor F and crossover rate Cr are made adaptive, also the mutation and selection phases of the original DE are substituted by the best individual-based mutation and elitist selection techniques. Fan et al. (2016) proposed a multi-population multi-strategies differential evolution algorithm that was applied for searching the globally stable structure of Fe and Cr nano-clusters.

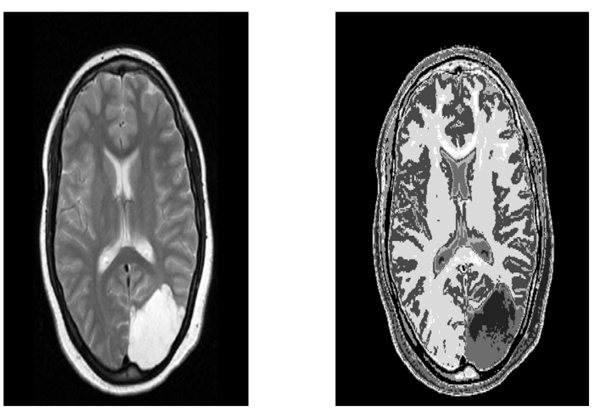
Li et al. (2016) combined two DE strategies and developed a new hybridized DE algorithm. In their work two popular techniques of DE (JADE and CoDE), are combined to form a new DE algorithm named as HMJCDE. Yeh et al. (2017) introduced a new version of DE named Gaussian barebones differential evolution (GBDE). In this algorithm they proposed two strategies (1) Gaussian based mutation



DE in cryptosystem (Abuhaiba and Hassan, 2011)

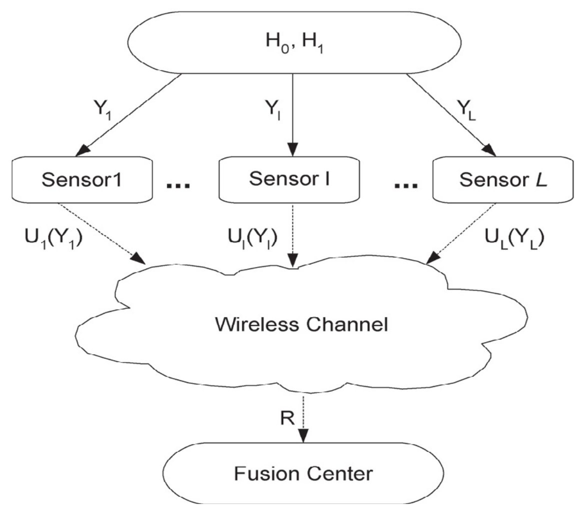


DE in Industrial Engineering (Textile spinning ring frame) (Raj et al., 2008)



c

DE in Image Processing (MRI images) (Omran and Engelbrecht, 2006)



DE in Power Engineering (Binary hypothesis detection problem) (Boussaïd et al., 2011)

Fig. 11. Screen capture of selected DE applications (Abuhaiba and Hassan, 2011; Raj et al., 2008; Omran and Engelbrecht, 2006; Boussaïd et al., 2011).

strategy (2) hybrid crossover strategy, the hybridization of the binomial and arithmetic crossover strategies, for DE to balance the global search ability and convergence rate. Sahoo et al. (2016) proposed a DE variant by changing its important parameters step size and crossover probability for achieving better performance. The strategy used was DE/rand-to-rand, with population size as 70; F and CR were decreased exponentially from 1.0 to 0.001 and 1.0 to 0.1 respectively. They applied it for solving load frequency control (LFC) of interconnected power systems. Cárdenas-Montes (2018) suggested Weibull-based scaled-differences schema for Differential Evolution. In this work, the statistical distribution of these differences of high-performance variants of Differential Evolution is modeled through a Weibull probability distribution.

Banitalebi et al. (2016) proposed a self-adaptive binary DE in which new trial vectors are generated adaptively. Wang et al. (2018a) proposed a novel global and local surrogate-assisted DE for solving an expensive constrained optimization problem with inequality constraints. The proposed method consisted of two main phases: (1) global surrogate-assisted phase and (2) local surrogate-assisted phase. In the global surrogate-assisted phase, DE served as the search engine to produce multiple trial vectors. In the local surrogate-assisted phase, the interior point method coupled with radial basis function is utilized to refine each individual in the population.

Peñuñuri et al. (2016) employed self-adaptiveness in DE and used DE/rand/1/bin strategy for finding a good value of Cr parameter. Zhou et al. (2017) suggested Adaptive DE with sorting crossover rate for continuous optimization problems. Qiu et al. (2016) maintained an equilibrium for exploration and exploitation by giving the same probability to population-based and neighborhood-based mutation operators respectively and applied this for solving multi-objective problems. Sakr et al. (2017) solved the optimal reactive power management problem using a modified multi-objective DE. Wang and Tang (2016) applied the concept of clusters for solving multi-objective optimization problems in DE. Zhou et al. (2016) worked for solving NP-hard problem of job scheduling by modifying a DE variant in which individuals in the population were coded into job permutations and based on this

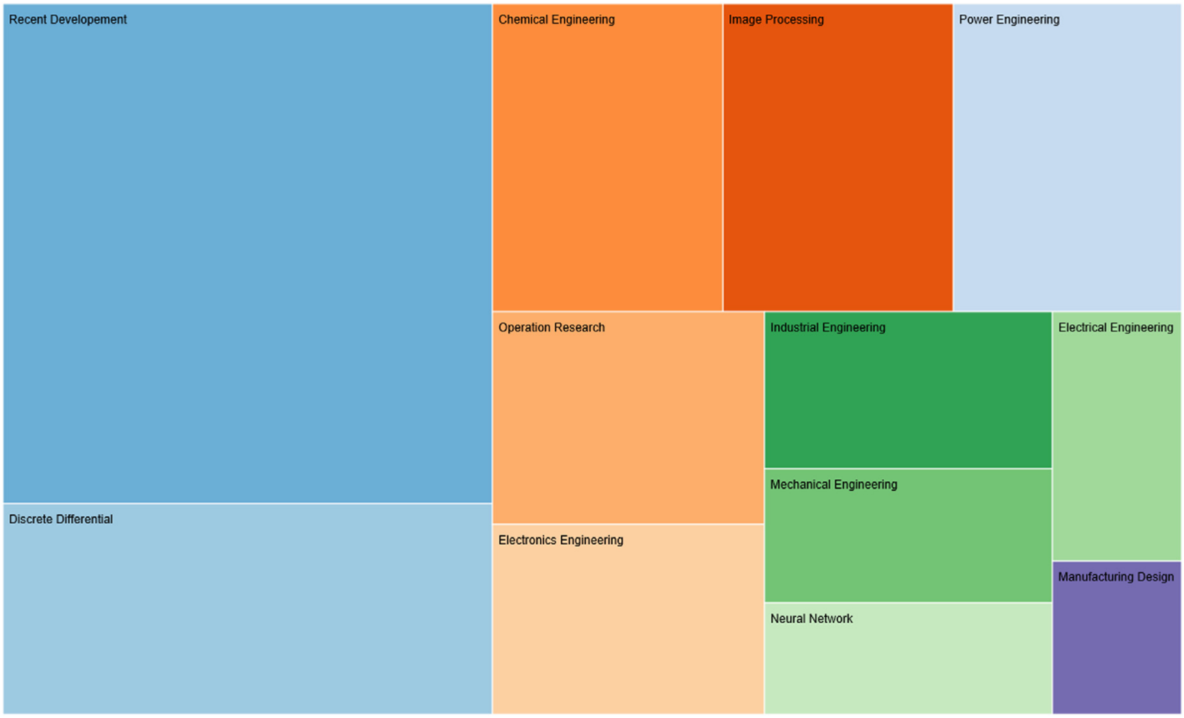


Fig. 12. Quantitative analysis of research articles on the basis of applications of DE. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

encoding system, discrete mutation and crossover operations were proposed. Wang et al. (2018b) proposed a composite DE for solving constrained optimization problems. Here, the authors used three different trial vector strategies to maintain a balance between diversity and convergence. Zhou and Zhang (2018) introduced a DE variant with an underestimation based multi-mutation strategy (UMS). DE highly depends on the mutation vector and it is very difficult to choose the best mutation strategy for some optimization problem. So, they proposed an Underestimation-Based multi-mutation strategy in DE. Hultmann Ayala et al. (2015) used Free Search Differential Evolution (FSDE) algorithm to solve power engineering problems. Sethanan and Pitakaso (2016b) proposed five modified DE algorithms by adding two additional steps, reincarnation and survival process and applied it to solve a typical type of vehicle routing problem. Du et al. (2017) borrowed idea from control theory and applied it in DE to improve its performance. Shih et al. (2017) proposed an enhanced differential evolution algorithm for solving directional overcurrent relay (DOCR) coordination problem. Segura et al. (2015) proposed a DE variant for dealing with the problems having large dimensions. Two main modifications in their work were the increase in the diversity of trial vectors and adaptive behavior of mutant factor. Balaji et al. (2016) developed a DE variant named mathematical approach assisted differential evolution (MADE) for solving maintenance scheduling problem in power system. Salehinejad et al. (2017) worked on the idea of low population size DE for faster convergence. They proposed a new mutation scheme in which mutation can be performed even when the population size is less than four population size. Wang et al. (2017b) worked on a multi-objective evolution algorithm based on neighborhood and gradient mechanism for achieving the exploration–exploitation balance of the algorithm. The algorithm was tested on two, three and four objectives optimization problem of wind turbine blade design. Chen et al. (2016) proposed a new ranking-based mutation operator in DE and applied the algorithm for solving dynamic optimization problems arising in chemical engineering. Pham (2016) developed a DE variant with three modifications in it including adaptive p-best strategy, the directional rule for mutation and the nearest neighborhood comparison method for solution. This variant was applied for a truss optimization problem with dynamic constraints. Koutny (2016) proposed meta DE in the medical field. They validated their results in measuring the continuous blood glucose level in diabetic patients from Jaeb Center for Health Research. Castillo et al. (2019) further proposed a dynamic parameter adaptation in meta-heuristics using a high-speed interval type 2 fuzzy system approach. Harold et al. (Chamorro et al., 2019) a novel fuzzy adaptive DE algorithm for the tuning of a fuzzy controller for the improvement of the synthetic inertia control in power systems. Prauzek et al. (2016) applied DE on wireless sensors with the help of fuzzy controllers. Fig. 12, tree-map representation justifies the research articles by the proportion of colored area.

5. Bibliometric analysis

This section visualizes the bibliometric analysis for this study based on 'InCites' (https://incites.clarivate.com/).

5.1. Statistics according to quartile

Based on Impact Factor (IF) data, the journal citation report published by Thomson Reuters (https://incites.clarivate.com/) provides yearly rankings of science and social science journals, in the subject categories relevant for the journal.

Quartile rankings (Qr) are therefore derived for each journal in each of its subject categories according to which quartile of the IF distribution the journal occupies for that subject category. Q1 denotes the top 25% of the IF distribution, Q2 for middle-high position (between top 50% and top 25%), Q3 middle-low position (top 75% to top 50%), and Q4 the lowest position (bottom 25% of the IF distribution). Tables 4(a) and 4(b) provides the journal names and their quartiles. Impact factors of different quartile journals are provided in Fig. 13. Fig. 14 provides the quantitative analysis of Quartile rankings. Average IF of quartile ranking can be seen in Fig. 15.

Table 4(a)
Journal name and ranking.

S. No.	Q1	Q2
1.	IEEE Transactions on Evolutionary Computation	International Journal of Bio-Inspired Computation
2.	Swarm and Evolutionary Computation	Soft Computing
3.	Artificial Intelligence Review	Chemometrics and Intelligent Laboratory Systems
4.	Information Sciences	Progress in Nuclear Energy
5.	Applied Soft Computing Journal	Chemometrics and Intelligent Laboratory Systems
6.	IEEE Transactions on Cybernetics	CAD Computer-Aided Design
7.	European Journal of Operational Research	Optimization and Engineering
8.	IEEE Transactions on Systems, Man, and Cybernetics: Systems	Progress in Nuclear Energy
9.	Computers & Industrial Engineering	Journal of Hydro-Environment Research
10.	Computers and Operations Research	IET Generation, Transmission and Distribution
11.	Mechanism and Machine Theory	Pattern Recognition Letters
12.	Applied Mathematics and Computation	International Journal of Advanced Manufacturing Technology
13.	Expert Systems with Applications	Computers in Industry
14.	Computers and Operations Research	Neural Processing Letters
15.	Knowledge-Based Systems	International Journal of Production Research
16.	Mechanical Systems and Signal Processing	Electric Power Systems Research
17.	Neurocomputing	Computer Methods and Programs in Biomedicine
18.	IEEE Transactions on Industrial Informatics	
19.	Energy Conversion and Management	
20.	Computers and Chemical Engineering	
21.	Chemical Engineering Science	
22.	International Journal of Electrical Power and Energy Systems	
23.	IEEE Transactions on Antennas and Propagation	
24.	IEEE Antennas and Wireless Propagation Letters	
25.	IEEE Transactions on Geoscience and Remote Sensing	
26.	Agricultural Water Management	
27.	IEEE Transactions on Power Systems	
28.	IEEE Transactions on Vehicular Technology	
29.	Composite Structures	
30.	Computer Physics Communications	
31.	International Journal of Production Economics	
32.	Computers and Electronics in Agriculture	
33.	Advances in Engineering Software	
34.	Engineering Applications of Artificial Intelligence	

Table 4(b)
Journal name and ranking.

	Q3	Q4
1.	Acta Polytechnica Hungarica	Lecture Notes in Computer Science
2.	Computational Optimization and Applications	Optimization Letters
3.	Journal of Global Optimization	Operations Research Letters
4.	Pattern Analysis and Applications	Journal of Phase Equilibria and Diffusion
5.	Process Biochemistry	Intelligent Automation and Soft Computing
6.	IEEE Transactions on Magnetics	Applied Artificial Intelligence
7.	Energy	Computers and Geosciences
8.	Journal of Intelligent and Fuzzy Systems	Advances in Soft Computing
9.	Studies in Fuzziness and Soft Computing	

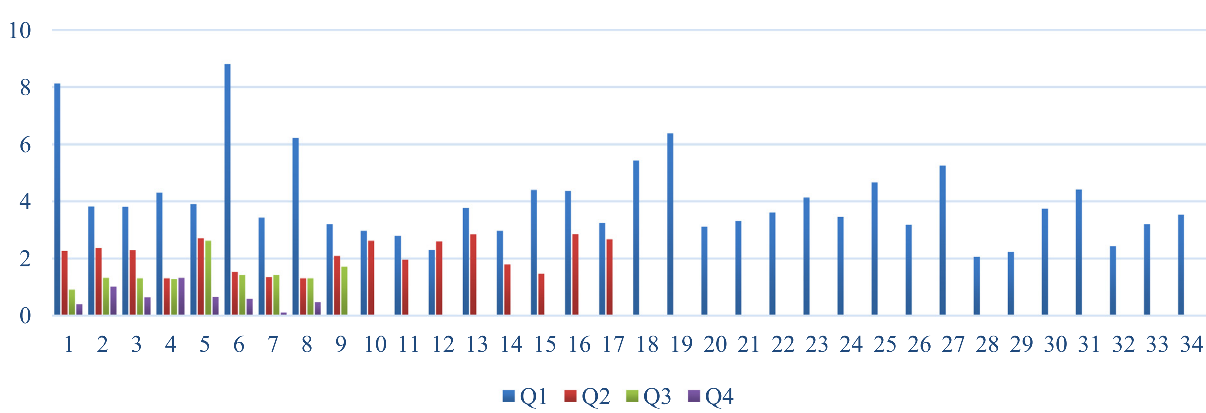


Fig. 13. Impact factor of different journals based on quartiles.

5.2. Statistics based on different publishers

The review done in the above section incorporates 283 papers; including 137 journal articles from Elsevier, 30 journal articles from Springer, 35 journal articles from IEEE, 6 journal articles from Taylor & Francis, 11 journal articles, 56 conferences and 9 books taken from the other sources. The articles have been explored through the different keywords like; Differential Evolution + optimization, Differential

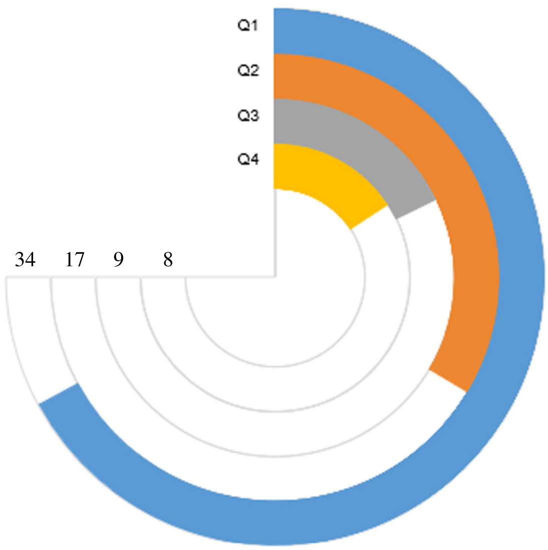


Fig. 14. Quantitative analysis of JCR quartile ranking.

Table 5
Number of papers in different publishers.

Publisher	Number of journals
Elsevier	137
Springer	30
IEEE	35
Taylor & Francis	6
Others	11

Evolution + optimization + variants, Differential Evolution + mutation, Differential Evolution + crossover, Differential Evolution + selection, Differential Evolution + change in parameters, Hybrid + Differential Evolution, Differential Evolution + review (TITLE), Differential Evolution + survey (TITLE). Table 5 shows the number of reviewed papers published in different publishers and can be visualized in Fig. 16(a), which shows the reviewed research article percentage in different publishers. Fig. 16(b) shows the number of reviewed research papers published in the last two decades. This study shows that different authors from different countries (Fig. 17) working on DE and applied in different application areas.

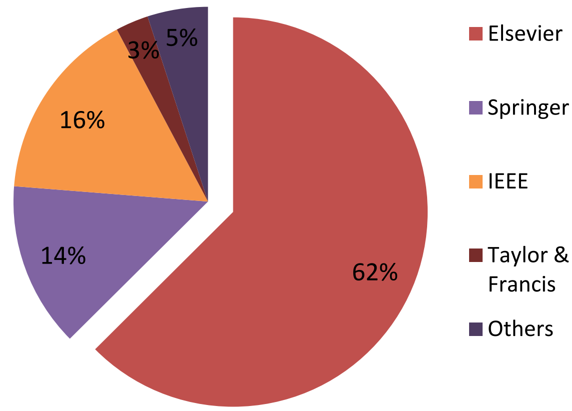


Fig. 16(a). Statistics based on different publishers.

For further analysis, we have generated network visualization map and density visualization map using "VOSviewer" (https://www.vosviewer.com/). Network Visualization consists for items (circles and labels) and links (relation between the items). Items are grouped into the clusters specified by the different colors assigned to the items. Fig. 18 shows the network visualization of keyword of the reviewed articles and link between them shows their co-occurrences. Figure shows the keyword 'differential evolution' has the highest occurrences and it is highly co-occurred with 'optimization,' 'global optimization,' and 'evolutionary algorithms' etc. Fig. 19 shows the network visualization of the terms taken from the abstract and title of the reviewed articles, links indicates the number of articles in which two terms occur together. In this the term 'model' has the highest occurrences and it strongly co-occurred with 'accuracy,' 'data' and 'prediction' etc.

6. Analysis and discussions

The main aim of this paper is to generate an interest among the readers for DE and to explore its potentials. In the present study, an attempt is made to provide an overview of the expedition of DE for a little more than 20 years (1995–2018) along with few papers from 2019. The journey is shown through the modifications that are proposed in the basic operators and parameters of DE like modification in the initialization methods, modification in mutation, crossover and selection operators; discrete variants of DE; hybridization and selected recent developments in DE.

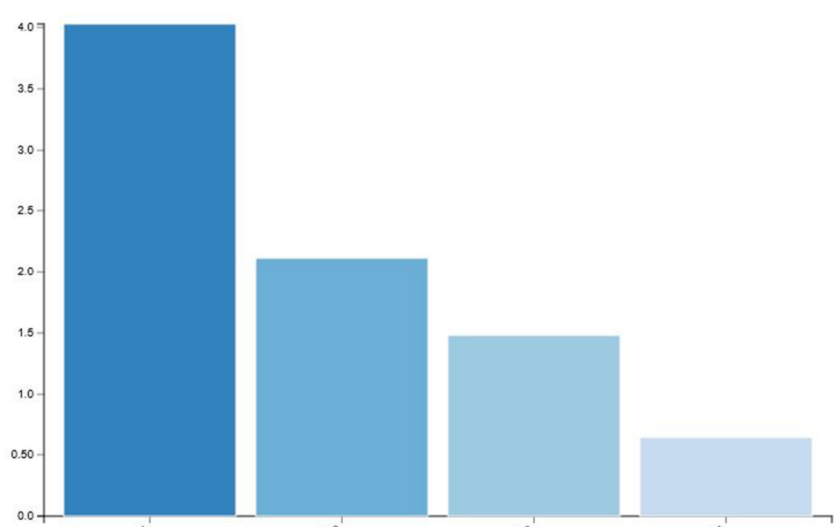


Fig. 15. Average impact factor different quartiles.

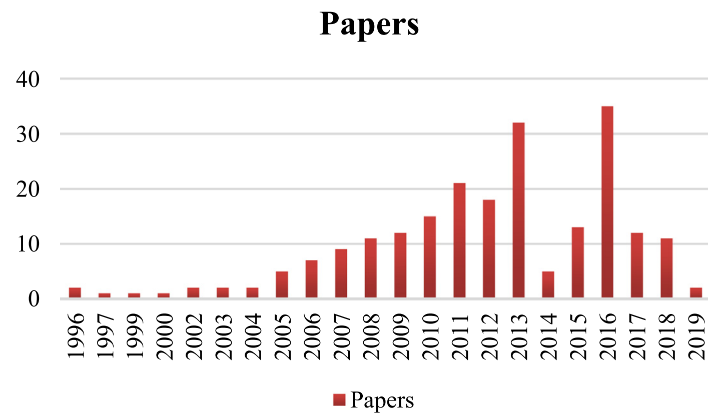


Fig. 16(b). Statistics based on number of papers in a particular year.

Fig. 17. Different countries working on DE.

From the collected data and statistics, the following inferences can be drawn:

1. DE has remained to be a consistent player in the research world of Evolutionary and Nature-Inspired Algorithms with DE related articles appearing in almost all the leading journals and publishers in allied areas.
2. From the application point of view, it can be seen that DE has been applied to various fields however, the areas where DE has been applied most frequently include Chemical Engineering, Industrial Engineering, and Image Processing, with each of these fields contributing to 14% of the total number of papers considered in the paper.
3. Modified variants of DE reveal that most of the work has been in proposing modified mutation schemes, taking 24% of the total publications considered in this study. This is an understandable development because the main operator of DE is a mutation. Hybridization is another area which attracted the researchers and makes up to 18% of the total contributions considered in this study. Fine-tuning of DE parameters and suggestions for the most appropriate values for DE parameters and proposals of discrete variants of DE have also gained the attention of researchers making 15% and 11% of the total contributions considered in this paper.
4. It is also observed that modifications in DE have also been a popular choice among the contestants of IEEE CEC competitions, with variants of DE obtaining a position among the top three entries in the competition since 2005. DE secured 1st position 2006; later for two consecutive years 2009 and 2010 and also for three consecutive years 2014–2016; in 2007, 2011 and 2018 DE obtained 2nd rank and in 2005, 2008 and 2017 DE secured 3rd rank among its competitors. Further it is also observed that most of the winning DE algorithms follow adaptive or self-adaptive strategy.
5. Bibliometric analysis reveals that 62% of the articles considered in this study are of Elsevier, followed by IEEE xplore (16%) and Springer (14%). JCR quartile ranking shows that out of the journal articles considered in this paper, 34 lies in the first quartile (top 25%) or Q1 category having an average impact of around 4.0, 17 lies in the second quartile (next 25%) i.e. Q2 category

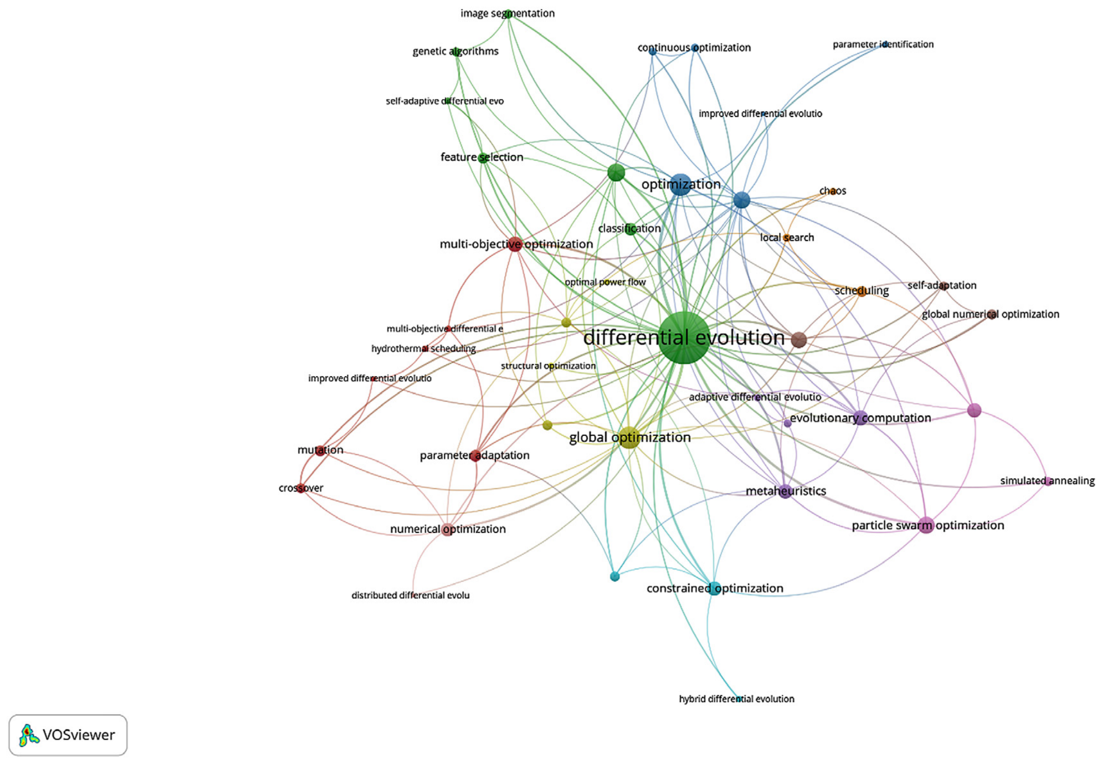


Fig. 18. Network visualization of the 'keywords'.

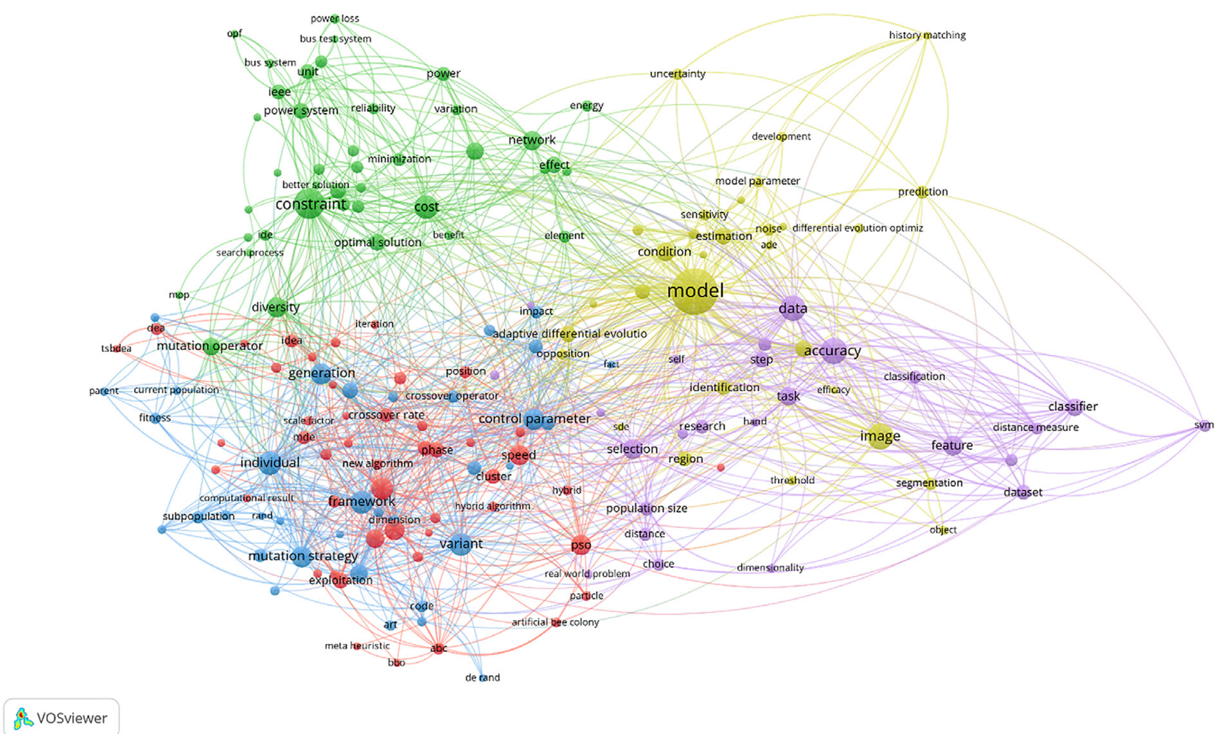


Fig. 19. Network visualization of the terms.

having an average impact of around 2.0, and the remaining 17 lies in the third (Q3) and the fourth (Q4) quartile with an average impact of around 1.0. Demographic spread of DE shows that most of the papers considered in this study are concentrated in Europe, South America and Africa.

7. Conclusions and future directions

The present study is done in two phases. The first phase provides the survey of literature on the basic aspects of DE, and in the second phase bibliometric analysis is provided. The first phase is expected to help the readers in identifying the potential areas of research in DE

Table A.1

Function name	Test function	Initial range	Optimum
Ackley function	$f(x) = -20e^{-0.02\sqrt{D^{-1}\sum_{i=1}^{D}x_i^2}} - e^{D^{-1}\sum_{i=1}^{D}cos(2\pi x_i)} + 20 + e$	$[-35, 35]^D$	$f(x^*) = 0$
Bohachevsky function	$f(x) = x_1^2 + x_2^2 - 0.3cos(3\pi x_1) - 0.4cos(4\pi x_2) + 0.7$	$[-100, 100]^D$	$f(x^*) = 0$
Dixon & Price function	$f(x) = (x_i - 1)^2 + \sum_{i=2}^{D} i(2x_i^2 - x_{i-1})^2$	$[-10, 10]^D$	$f(x^*) = 0$
Griewank function	$f(x) = \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-100, 100]^D$	$f(x^*) = 0$
Matyas function	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	$[-10, 10]^D$	$f(x^*) = 0$
Powell function	$f(x) = \sum_{i=1}^{D/4}\left[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4\right]$	$[-4, 5]^D$	$f(x^*) = 0$
Powell sum function	$f(x) = \sum_{i=1}^{D}\lvert x_i\rvert^{i+1}$	$[-1, 1]^D$	$f(x^*) = 0$
Rosenbrock function	$f(x) = \sum_{i=1}^{D-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	$[-30, 30]^D$	$f(x^*) = 0$
Rastringin function	$f(x) = \sum_{i=1}^{D}(x_i^2 - 10cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^D$	$f(x^*) = 0$
Scahffer function	$f(x) = 0.5 + \frac{sin^2(x_1^2 + x_2^2)^2}{1 + 0.001(x_1^2 + x_2^2)^2}$	$[-100, 100]^D$	$f(x^*) = 0$
Sphere function	$f(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$	$f(x^*) = 0$
Schwefel function	$f(x) = \sum_{i=1}^{D} -x_i sin\sqrt{\lvert x_i\rvert} + 418.982887D$	$[-500, 500]^D$	$f(x^*) = 0$
Step function	$f(x) = \sum_{i=1}^{D}\lfloor x_i + 0.5\rfloor^2$	$[-100, 100]^D$	$f(x^*) = 0$
Sum squares function	$f(x) = \sum_{i=1}^{D} ix_i^2$	$[-10, 10]^D$	$f(x^*) = 0$
Trid function	$f(x) = \sum_{i=1}^{D}(x_i - 1)^2 - \sum_{i=1}^{D} x_i x_{i-1}$	$[-100, 100]^D$	$f(x^*) = -200$
Watson function	$f(x) = \sum_{i=0}^{29}$	$\lvert x_i\rvert \le 10;\ a_i = i/29$	$f(x) = 0.002288$
Weierstrass function	$f(x) = \sum_{i=1}^{D}\left[\sum_{k=0}^{kmax} a^k cos(2\pi b^k(x_i + 0.5)) - D\sum_{k=0}^{kmax} a^k cos(\pi b^k)\right]$	$[-0.5, 0.5]^D$	$f(x^*) = 0$
Wolfe function	$f(x) = \frac{4}{3}(x_1^2 + x_2^2 - x_1x_2)^0.75 + x_3$	$[0, 2]^D$	$f(x^*) = 0$
Zakharov function	$f(x) = \sum_{i=1}^{D} x_i^2 + \left(\frac{1}{2}\sum_{i=1}^{D} ix_i\right)^2 + \left(\frac{1}{2}\sum_{i=1}^{D} ix_i\right)^4$	$[-5, 10]^D$	$f(x^*) = 0$
$z = x - o;\ o = Shifted\ Global\ Optimum$			
Shifted Sphere function	$f(x) = \sum_{i=1}^{D} z_i^2 + f_{bia}s_1$ $f = -450$	$[-100, 100]^D$	$f(x^*) = -450$
Shifted Schwefel's function	$f(x) = \max_i\{\lvert z_i\rvert, 1 \le i \le D\} + f_{bia}s_2;$ $f_{bia}s_2 = -450$	$[-100, 100]^D$	$f(x^*) = -450$
Shifted Rosenbrock's function	$f(x) = \sum_{i=1}^{D}\left(100(z_i^2 - z_{i+1}) + (z_i - 1)^2\right) + f_{bia}s_3;$ $f_{bia}s_3 = 390$	$[-100, 100]^D$	$f(x^*) = 390$

(continued on next page)

Table A.1 (*continued*).

Function name	Test function	Initial range	Optimum
Shifted Rastrigin's function	$f(x) = \sum_{i=1}^{D}\left(z_i^2 - 10cos\left(2\pi x_i\right) + 10\right) + f_{bia}s_4$; $f_{bia}s_4 = -330$	$[-5, 5]^D$	$f(x^*) = -330$
Shifted Griewank's function	$f(x) = \sum_{i=1}^{n}\frac{z_i^2}{4000} - \prod cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f_{bia}s_5$; $f_{bia}s_5 = -180$	$[-600, 600]^D$	$f(x^*) = -180$
Shifted Ackley's function	$f(x) = -20e^{-0.02\sqrt{D^{-1}\sum_{i=1}^{D}z_i^2}} - e^{D^{-1}\sum_{i=1}^{D}cos(2\pi z_i)} + 20 + e + f_{bia}s_6$; $f_{bias_6} = -140$	$[-32, 32]^D$	$f(x^*) = -140$
$z = M(x - o)$; $o = Shifted\ Global\ Optimum$; $M = linear\ Trnsformation$			
Rotated high conditioned elliptic function	$f(x) = \sum_{i=1}^{D}\left(10^6\right)^{\frac{i-1}{D-1}} z_i^2 + f_{bia}s_1$; $f_{bia}s_1 = -1300$	$[-100, 100]^D$	$f(x^*) = -1300$
Rotated bent cigar function	$f(x) = z_1^2 + 10^6\sum_{i=2}^{D} z_i^2 + f_{bia}s_2$; $f_{bia}s_2 = -1200$	$[-100, 100]^D$	$f(x^*) = -1200$
Rotated discus function	$f(x) = 10^6 z_1^2 + \sum_{i=2}^{D} z_i^2 + f_{bia}s_3$; $f_{bia}s_3 = -1100$	$[-100, 100]^D$	$f(x^*) = -1100$
Different power function	$f(x) = \frac{10}{D^2}\prod_{i=1}^{D}\left(1 + i\sum_{j=1}^{32}\frac{\lvert 2^j z_i - round\left(2^j z_i\right)\rvert}{2^j}\right)^{\frac{10}{D^{1.2}}} - \frac{10}{D^2} + f_{bia}s_5$; $f_{bia}s_5 = -1000$; $z = M\frac{5(x - o)}{100}$	$[-100, 100]^D$	$f(x^*) = -1000$

Table A.2

Benchmark function	Multimodal	Unimodal	Shifted	Rotated	Separable	Non-separable	Scalable	Non-scalable
Ackley function (Bäck and Schwefel, 1993; Suganthan et al., 2005)	✔		✔	✔		✔	✔	
Bohachevsky function (Bohachevsky et al., 1986)	✔				✔			✔
Bent cigar function (Liang et al., 2013)		✔		✔		✔		
Discus function (Liang et al., 2013)		✔		✔		✔		
Different powers function (Liang et al., 2013)	✔				✔			
Dixon & Price function (Dixon and Price, 1989)		✔				✔	✔	
Elliptic function (Liang et al., 2013)		✔	✔	✔		✔		
Griewank function (Griewank, 1981; Suganthan et al., 2005)	✔	✔	✔			✔	✔	
Matyas function (Hedar, 2019)		✔				✔		✔
Powell function (Powell, 1962)		✔				✔	✔	
Power Sum function (Rahnamayan et al., 2007)		✔			✔		✔	
Rastrigin function (Liang et al., 2013; Suganthan et al., 2005)	✔		✔	✔				
Rosenbrock function (Rosenbrock, 1960; Suganthan et al., 2005)		✔	✔			✔	✔	
Schwefel function (Hans-Paul, 1981), (Liang et al., 2013)	✔	✔	✔		✔		✔	
Schaffers function (Liang et al., 2013)	✔			✔		✔		
Sphere function (Schumer and Steiglitz, 1968; Suganthan et al., 2005)	✔		✔		✔		✔	
Sum squares function (Hedar, 2019)		✔			✔		✔	
Trid function (Hedar, 2019)	✔					✔		✔
Watson function (Hans-Paul, 1981)		✔				✔	✔	
Weierstrass function (Suganthan et al., 2005; Liang et al., 2013)	✔		✔	✔	✔		✔	
Wolfe function (Hans-Paul, 1981)		✔			✔		✔	
Zakharov function (Rahnamayan et al., 2007)	✔					✔	✔	
Hybrid and composition function								
Composition of Sphere's function (Li et al., 2008)	✔			✔			✔	
Composition of Rastrigin's function (Li et al., 2008)	✔			✔			✔	
Composition of Griewank's function (Li et al., 2008)	✔			✔			✔	
Composition of Ackley's function (Li et al., 2008)	✔			✔			✔	
Hybrid composition function (Li et al., 2008)	✔			✔			✔	
Expanded Griewank's plus Rosenbrock's function (Liang et al., 2013)	✔			✔		✔		
Expanded Scaffer's F6 function (Liang et al., 2013)	✔			✔		✔		

and Bibliometric analysis will be beneficial to the readers in identifying the leading journals and publication houses related to this area. The demographic spread may also help in collaborative research work. Some concluding remarks based on the present study are:

In most of the hybrid variants of DE, hybridization is done with some other meta-heuristics (mostly PSO as per the papers considered in this study). Though the integration of two meta-heuristic results in improving the quality of the solution, it rarely provides a deep insight on the "*whats*" and "*hows*" of the combination. A thorough investigation on hybridization methods may help in developing important and concrete guidelines which in turn may help the beginners working in this area. Also, there are very limited examples of combining DE (or any other meta-heuristic algorithms) with classical methods. Interaction of meta-heuristics with methods like mathematical programming methods can be an interesting research direction.

DE is naturally suited for continuous optimization problems and certain level of efforts are required for making it compatible for discrete and combinatorial optimization problems. Many researchers are working in this area to develop a suitable discrete variant of DE. Also, it is observed that the application of DE for solving multi and many-objective optimization problems is also limited. These two research areas can be explored further to analyze the potential of DE.

Finally, the authors would like to add that though we have tried to include most of the relevant DE publications, there is a possibility that some important articles are missed, for which the authors apologize.

Appendix

A.1. List of some benchmark test problems and their properties

See Table A.1.

A.2. List of benchmark test problems properties

See Table A.2.

References

Aalto, J., Lampinen, J., 2015. A population adaptation mechanism for differential evolution algorithm. In: Proceedings - 2015 IEEE Symposium Series on Computational Intelligence, SSCI 2015, pp. 1514–1521.

Abuhaiba, I.S.I., Hassan, M.A.S., 2011. Image encryption using differential evolution approach in frequency domain. Signal Image Process. Int. J. 2 (1), 51–69.

Adeyemo, J., Otieno, F., 2010. Differential evolution algorithm for solving multi-objective crop planning model. Agric. Water Manage. 97 (6), 848–856.

Akhmedova, S., Stanovov, V., Semenkin, E., 2018. LSHADE algorithm with a rank-based selective pressure strategy for the circular antenna array design problem. pp. 159–165.

Akrout, H., Jarboui, B., Siarry, P., Rebaï, A., 2012. A GRASP based on DE to solve single machine scheduling problem with SDST. Comput. Optim. Appl. 51 (1), 411–435.

Al-dabbagh, R., Botzheim, J., Al-dabbagh, M., 2014. Comparative analysis of a modified differential evolution algorithm based on bacterial mutation scheme. In: 2014 IEEE Symposium on Differential Evolution, SDE, pp. 1–8.

Ali, M.M., 2007. Differential evolution with preferential crossover. European J. Oper. Res. 181 (3), 1137–1147.

Ali, M.Z., Awad, N.H., Suganthan, P.N., Reynolds, R.G., 2016. A modified cultural algorithm with a balanced performance for the differential evolution frameworks. Knowl.-Based Syst. 111, 73–86.

Ali, M., Pant, M., 2011. Improving the performance of differential evolution algorithm using Cauchy mutation. Soft Comput. 15 (5), 991–1007.

Ali, M., Pant, M., Abraham, A., 2009a. Simplex differential evolution. Acta Polytech. Hung. 6 (5), 95–115.

Ali, M., Pant, M., Abraham, A., 2009b. A Hybrid Ant Colony Differential Evolution and its application to water resources problems. In: 2009 World Congr. Nat. Biol. Inspired Comput., No. 1, pp. 1133–1138.

Ali, M., Pant, M., Abraham, A., 2013. Unconventional initialization methods for differential evolution. Appl. Math. Comput. 219 (9), 4474–4494.

Ameca-Alducin, M.Y., Mezura-Montes, E., Cruz-Ramírez, N., 2014. Differential evolution with combined variants for dynamic constrained optimization. In: Proceedings of the 2014 IEEE Congress on Evolutionary Computation, No. 5, CEC 2014, pp. 975–982.

Ameca-Alducin, M.-Y., Mezura-Montes, E., Cruz-Ramírez, N., 2015. A repair method for differential evolution with combined variants to solve dynamic constrained optimization problems. In: Proc. 2015 Genet. Evol. Comput. Conf., - GECCO '15, pp. 241–248.

Arya, R., Choube, S.C., 2013. Differential evolution based technique for reliability design of meshed electrical distribution systems. Int. J. Electr. Power Energy Syst. 48 (1), 10–20.

Arya, L.D., Koshti, A., Choube, S.C., 2012. Distributed generation planning using differential evolution accounting voltage stability consideration. Int. J. Electr. Power Energy Syst. 42 (1), 196–207.

Asafuddoula, M., Ray, T., Sarker, R., 2015. A differential evolution algorithm with constraint sequencing: An efficient approach for problems with inequality constraints. Appl. Soft Comput. J. 36, 101–113.

Awad, N.H., Ali, M.Z., Suganthan, P.N., 2017a. Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In: 2017 IEEE Congr. Evol. Comput. CEC 2017 - Proc., pp. 372–379.

Awad, N.H., Ali, M.Z., Suganthan, P.N., 2017b. Ensemble of parameters in a sinusoidal differential evolution with niching-based population reduction. Swarm Evol. Comput. 39 (2017), 141–156.

Awad, N.H., Ali, M.Z., Suganthan, P.N., Jaser, E., 2016b. A decremental stochastic fractal differential evolution for global numerical optimization. Inf. Sci. (Ny). 372, 470–491.

Awad, N.H., Ali, M.Z., Suganthan, P.N., Reynolds, R.G., 2016a. An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems. In: 2016 IEEE Congr. Evol. Comput. CEC2016, pp. 2958–2965.

Awad, N.H., Ali, M.Z., Suganthan, P.N., Reynolds, R.G., 2017c. CADE: A hybridization of Cultural Algorithm and Differential Evolution for numerical optimization. Inf. Sci. (Ny). 378, 215–241.

Baatar, N., Pham, M., Koh, C., 2013. Multiguiders and nondominate ranking differential evolution algorithm formultiobjective global optimization of electromagnetic problems. IEEE Trans. Magn. 49 (5), 2105–2108. http://dx.doi.org/10.1109/TMAG.2013.2240285.

Baatar, N., Zhang, D., Koh, C., 2013. An improved differential evolution algorithm adopting λ-best mutation strategy for global optimization of electromagnetic devices. IEEE Trans. Magn. 49 (5), 2097–2100. http://dx.doi.org/10.1109/TMAG.2013.2240284.

Babu, B.V., Angira, R., 2006. Modified differential evolution (MDE) for optimization of non-linear chemical processes. Comput. Chem. Eng. 30 (6–7), 989–1002.

Babu, B., Angira, R., 2008. Optimization of industrial processes using improved and modified differential evolution. Soft Comput. Appl. Ind. SE - Stud. Fuzziness Soft Comput. 226, 1–22.

Babu, B.V., Munawar, S.A., 2007. Differential evolution strategies for optimal design of shell-and-tube heat exchangers. Chem. Eng. Sci. 62 (14), 3720–3739.

Bäck, T., Schwefel, H.-P., 1993. An overview of evolutionary algorithms for parameter optimization. Evol. Comput. 1 (1), 1–23.

Bagdonavičius, V., Julius, K., Nikulin, M.S., 2013. Non-Parametric Tests for Complete Data. John Wiley and Sons.

Balaji, G., Balamurugan, R., Lakshminarasimman, L., 2016. Mathematical approach assisted differential evolution for generator maintenance scheduling. Int. J. Electr. Power Energy Syst. 82, 508–518.

Banitalebi, A., Aziz, M.I.A., Aziz, Z.A., 2016. A self-adaptive binary differential evolution algorithm for large scale binary optimization problems. Inf. Sci. (Ny). 367–368, 487–511.

Bhattacharya, A., Chattopadhyay, P.K., 2010. Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. IEEE Trans. Power Syst. 25 (4), 1955–1964.

Biswas, S., Kundu, S., Das, S., 2014. An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution. IEEE Trans. Cybern. 44 (10), 1726–1737.

Bohachevsky, I.O., Johnson, M.E., Stein, M.L., 1986. Generalized simulated annealing for function optimization. Technometrics 28 (3), 209–217.

Boussaïd, I., Chatterjee, A., Siarry, P., Ahmed-Nacer, M., 2011. Hybridizing biogeography-based optimization with differential evolution for optimal power allocation in wireless sensor networks. IEEE Trans. Veh. Technol. 60 (5), 2347–2353.

Boyd, Stephen, Vandenberghe, Lieven, 2004. Convex Optimization. Cambridge University Press.

Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V., 2006. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Trans. Evol. Comput. 10 (6), 646–657. http://dx.doi.org/10.1109/TEVC.2006.872133.

Brest, J., Zamuda, A., Bošković, B., Maučec, M.S., Žumer, V., 2008. High-dimensional real-parameter optimization using self-adaptive differential evolution algorithm with population size reduction. In: 2008 IEEE Congr. Evol. Comput., CEC2008, pp. 2032–2039.

Bujok, P., Tvrdík, J., 2015. New variants of adaptive differential evolution algorithm with competing strategies. Acta Electrotech. Inform. 15 (2), 49–56.

Cai, Z., Gong, W., Ling, C.X., Zhang, H., 2011. A clustering-based differential evolution for global optimization. Appl. Soft Comput. J. 11 (1), 1363–1379.

Cai, Y., Liao, J., Wang, T., Chen, Y., Tian, H., 2018. Social learning differential evolution. Inf. Sci. (Ny). (1339), 433–434–1351.

Caponio, A., Neri, F., Tirronen, V., 2009. Super-fit control adaptation in memetic differential evolution frameworks. Soft Comput. 13 (8–9), 811–831.

Cárdenas-Montes, M., 2018. Weibull-based scaled-differences schema for Differential Evolution. Swarm Evol. Comput. 38 (2016), 79–93.
Castillo, O., et al., 2019. A high-speed interval type 2 fuzzy system approach for dynamic parameter adaptation in metaheuristics. Eng. Appl. Artif. Intell. 85 (2018), 666–680.
Chamorro, H.R., Riaño, I., Gerndt, R., Zelinka, I., Gonzalez-Longatt, F., Sood, V.K., 2019. Synthetic inertia control based on fuzzy adaptive differential evolution. Int. J. Electr. Power Energy Syst. 105 (2018), 803–813.
Chang, C.F., Wong, J.J., Chiou, J.P., Su, C.T., 2007. Robust searching hybrid differential evolution method for optimal reactive power planning in large-scale distribution systems. Electr. Power Syst. Res. 77 (5–6), 430–437.
Chen, X., Du, W., Qian, F., 2016. Solving chemical dynamic optimization problems with ranking-based differential evolution algorithms. Chin. J. Chem. Eng. 24 (11), 1600–1608.
Cheng, J., Yen, G.G., Zhang, G., 2016. A grid-based adaptive multi-objective differential evolution algorithm. Inf. Sci. (Ny). 367–368, 890–908.
Choudhary, N., Sharma, H., Sharma, N., 2017. Differential evolution algorithm using stochastic mutation. In: Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2016, pp. 315–320.
Coelho, L.D.S., Mariani, V.C., Ferreira Da Luz, M.V., Leite, J.V., 2013. Novel gamma differential evolution approach for multiobjective transformer design optimization. IEEE Trans. Magn. 49 (5), 2121–2124.
Coello Coello, C.A., 2000. Use of a self-adaptive penalty approach for engineering optimization problems. Comput. Ind. 41 (2), 113–127.
Cuevas, E., Zaldivar, D., Pérez-Cisneros, M., 2010. A novel multi-threshold segmentation approach based on differential evolution optimization. Expert Syst. Appl. 37 (7), 5265–5271.
Cuevas, E., Zaldivar, D., Pérez-Cisneros, M., Ramírez-Ortegón, M., 2011. Circle detection using discrete differential evolution optimization. Pattern Anal. Appl. 14 (1), 93–107.
Cui, L., Li, G., Lin, Q., Chen, J., Lu, N., 2016. Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations. Comput. Oper. Res. 67, 155–173.
Das, S., Abraham, A., Chakraborty, U.K., Konar, A., 2009. Differential evolution using a neighborhood-based mutation operator. IEEE Trans. Evol. Comput. 13 (3), 526–553.
Das, S., Konar, A., 2009. Automatic image pixel clustering with an improved differential evolution. Appl. Soft Comput. J. 9 (1), 226–236.
Das, S., Mullick, S.S., Suganthan, P.N., 2016. Recent advances in differential evolution-An updated survey. Swarm Evol. Comput. 27, 1–30.
Das, S., Suganthan, P.N., 2011. Differential evolution: A survey of the state-of-the-art. IEEE Trans. Evol. Comput. 15 (1), 4–31.
Datta, D., Figueira, J.R., 2013. A real-integer-discrete-coded differential evolution. Appl. Soft Comput. J. 13 (9), 3884–3893.
De Falco, I., Della Cioppa, A., Maisto, D., Tarantino, E., 2008. Differential evolution as a viable tool for satellite image registration. Appl. Soft Comput. J. 8 (4), 1453–1462.
De Melo, V.V., Botazzo Delbem, A.C., 2012. Investigating Smart Sampling as a population initialization method for Differential Evolution in continuous problems. Inf. Sci. (Ny). 193, 36–53.
De Melo, V.V., Carosio, G.L.C., 2013. Investigating multi-view differential evolution for solving constrained engineering design problems. Expert Syst. Appl. 40 (9), 3370–3377.
Deng, G., Gu, X., 2012. A hybrid discrete differential evolution algorithm for the no-idle permutation flow shop scheduling problem with makespan criterion. Comput. Oper. Res. 39 (9), 2152–2160.
Deng, W., Yang, X., Zou, L., Wang, M., Liu, Y., Li, Y., 2013. An improved self-adaptive differential evolution algorithm and its application. Chemom. Intell. Lab. Syst. 128, 66–76.
Di Carlo, M., Vasile, M., Minisci, E., 2015. Multi-population inflationary differential evolution algorithm with adaptive local restart. pp. 632–639.
Dixon, L.C.W., Price, R.C., 1989. Truncated Newton method for sparse unconstrained optimization using automatic differentiation. J. Optim. Theory Appl. 60 (2), 261–275.
Do, D.T.T., Lee, S., Lee, J., 2016. A modified differential evolution algorithm for tensegrity structures. Compos. Struct. 158, 11–19.
Dong, B., Zhou, A., Zhang, G., 2016. A hybrid estimation of distribution algorithm with differential evolution for global optimization, In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, pp. 1–7.
Dos Santos, G.S., Luvizotto, L.G.J., Mariani, V.C., Coelho, L.D.S., 2012. Least squares support vector machines with tuning based on chaotic differential evolution approach applied to the identification of a thermal process. Expert Syst. Appl. 39 (5), 4805–4812.
Dragoi, E.N., Curteanu, S., Galaction, A.I., Cascaval, D., 2013. Optimization methodology based on neural networks and self-adaptive differential evolution algorithm applied to an aerobic fermentation process. Appl. Soft Comput. J. 13 (1), 222–238.
Du, J.X., Huang, D.S., Wang, X.F., Gu, X., 2007. Shape recognition based on neural networks trained by differential evolution algorithm. Neurocomputing 70 (4–6), 896–903.
Du, W., Leung, S.Y.S., Tang, Y., Vasilakos, A.V., 2017. Differential evolution with event-triggered impulsive control. IEEE Trans. Cybern. 47 (1), 244–257.
Du Plessis, M.C., Engelbrecht, A.P., 2012. Using Competitive Population Evaluation in a differential evolution algorithm for dynamic environments. European J. Oper. Res. 218 (1), 7–20.
Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory. In: MHS'95 Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43.
El Dor, A., Clerc, M., Siarry, P., 2012. Hybridization of differential evolution and particle swarm optimization in a new algorithm: DEPSO-2S. Swarm Evol. Comput. 7269, 57–65.
El Ela, A.A.A., Abido, M.A., Spea, S.R., 2011. Differential evolution algorithm for optimal reactive power dispatch. Electr. Power Syst. Res. 81 (2), 458–464.
Elsayed, S.M., Sarker, R.A., Essam, D.L., 2013a. An improved self-adaptive differential evolution algorithm for optimization problems. IEEE Trans. Ind. Inform. 9 (1), 89–99.
Elsayed, S.M., Sarker, R.A., Essam, D.L., 2013b. Self-adaptive differential evolution incorporating a heuristic mixing of operators. Comput. Optim. Appl. 54 (3), 771–790.
Epitropakis, M.G., Tasoulis, D.K., Pavlidis, N.G., Plagianakos, V.P., Vrahatis, M.N., 2011. Enhancing differential evolution utilizing proximity-based mutation operators. IEEE Trans. Evol. Comput. 15 (1), 99–119.
Fan, H., Lampinen, J., Dulikravich, G.S., 2003. Improvement to Mutation Donor Formulation of, Optimization. pp. 1–12.
Fan, T.E., Shao, G.F., Ji, Q.S., Zheng, J.W., dong Liu, T., Wen, Y.H., 2016. A multi-populations multi-strategies differential evolution algorithm for structural optimization of metal nanoclusters. Comput. Phys. Comm. 208, 64–72.
Fan, Q., Zhang, Y., 2016. Self-adaptive differential evolution algorithm with crossover strategies adaptation and its application in parameter estimation. Chemom. Intell. Lab. Syst. 151 (1550), 164–171.
Fatih Tasgetiren, M., Suganthan, P.N., Pan, Q.K., 2010. An ensemble of discrete differential evolution algorithms for solving the generalized traveling salesman problem. Appl. Math. Comput. 215 (9), 3356–3368.
Fister, I., Tepeh, A., Fister, I., 2016. Epistatic arithmetic crossover based on Cartesian graph product in ensemble differential evolution. Appl. Math. Comput. 283, 181–194.
Fogel, L.J., Owens, A.J., Walsh, M.J., 1965. Artificial intelligence through a simulation of evolution. In: Biophysics and {C}ybernetic {S}ystems: {P}roc. of the 2nd {C}ybernetic {S}ciences {S}ymposium, pp. 131–155.
Gämperle, R., Müller, S.D., Koumoutsakos, P., 2002. A parameter study for differential evolution. Adv. Intell. Syst. Fuzzy Syst. Evol. Comput. 10, 293–298.
Georgilakis, P.S., 2009. Differential evolution solution to transformer no-load loss reduction problem. IET Gener. Transm. Distrib. 3 (10), 960–969.
Ghosh, A., Das, S., Chowdhury, A., Giri, R., 2011. An improved differential evolution algorithm with fitness-based adaptation of the control parameters. Inf. Sci. (Ny). 181 (18), 3749–3765.
Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Read. Addison-Wesley, pp. 611–616.
Gong, W., Cai, Z., 2013. Differential evolution with ranking-based mutation operators. IEEE Trans. Cybern. 43 (6), 2066–2081.
Gong, W., Cai, Z., Wang, Y., 2014. Repairing the crossover rate in adaptive differential evolution. Appl. Soft Comput. J. 15, 149–168.
Gong, T., Tuson, A.L., 2007. Differential evolution for binary encoding. Adv. Soft Comput. 39 (1), 251–262.
Goudos, S.K., Baltzis, K.B., Antoniadis, K., Zaharis, Z.D., Hilas, C.S., 2011. A comparative study of common and self-adaptive Differential Evolution strategies on numerical benchmark problems. Proced. Comput. Sci. 3, 83–88.
Goudos, S.K., Gotsis, K.A., Siakavara, K., Vafiadis, E.E., Sahalos, J.N., 2013. A multi-objective approach to subarrayed linear antenna arrays design based on memetic differential evolution. IEEE Trans. Antennas Propag. 61 (6), 3042–3052.
Griewank, A.O., 1981. Generalized descent for global optimization. J. Optim. Theory Appl. 34 (1), 11–39.
Guo, S.M., Hsu, P.H., Yang, C.C., Tsai, J.S.H., 2016. Constrained min–max optimization via the improved constraint-activated differential evolution with escape vectors. Expert Syst. Appl. 46, 336–345.
Guo, H., Li, Y., Li, J., Sun, H., Wang, D., Chen, X., 2014. Differential evolution improved with self-adaptive control parameters based on simulated annealing. Swarm Evol. Comput. 19, 52–67.
Guo, S.M., Tsai, J.S.H., Yang, C.C., Hsu, P.H., 2015. A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set. In: 2015 IEEE Congr. Evol. Comput. CEC 2015 - Proc., pp. 1003–1010.
Guo, S.M., Yang, C.C., 2015. Enhancing differential evolution utilizing eigenvector-based crossover operator. IEEE Trans. Evol. Comput. 19 (1), 31–49.
Hachicha, N., Jarboui, B., Siarry, P., 2011. A fuzzy logic control using a differential evolution algorithm aimed at modelling the financial market dynamics. Inf. Sci. (Ny). 181 (1), 79–91.
Hans-Paul, 1981. Numerical Optimization of Computer Models. Wiley.
He, X., Zhang, Q., Sun, N., Dong, Y., 2009. Feature selection with discrete binary differential evolution. In: 2009 International Conference on Artificial Intelligence and Computational Intelligence, Vol. 4, AICI 2009, pp. 327–330.

Hedar, A.-R., 2019. Global optimization test problems. [Online]. Available: http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.html. (Accessed 03 October 2019).
Hultmann Ayala, H.V., dos S. Coelho, L., Mariani, V.C., Askarzadeh, A., 2015. An improved free search differential evolution algorithm: A case study on parameters identification of one diode equivalent circuit of a solar cell module. Energy 93, 1515–1522.
Ibarra, O.H., Kim, C.E., 1977. Heuristic algorithms for scheduling independent tasks on nonidentical processors. J. ACM 24 (2), 280–289.
Ilonen, J., Kamarainen, J.-K., Lampinen, J., 2003. Differential evolution training algorithm for feed-forward neural networks. Neural Process. Lett. 17 (1), 93–105.
Islam, S.M., Das, S., Ghosh, S., Roy, S., Suganthan, P.N., 2012. An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. IEEE Trans. Syst. Man Cybern. B 42 (2), 482–500.
Jiang, L., Qiang, H., 2013. Differential evolution algorithm with base vector group. In: Proceedings of the 32nd Chinese Control Conference, Xi'an, pp. 8006–8009.
Joshi, R., Sanderson, A.C., 1999. Minimal representation multisensor fusion using differential evolution. IEEE Trans. Syst. Man Cybern. A 29 (1), 63–76.
Kadhar, K.M.A., Baskar, S., Amali, S.M.J., 2015. Diversity controlled self adaptive differential evolution based design of non-fragile multivariable PI controller. Eng. Appl. Artif. Intell. 46, 209–222.
Kaelo, P., Ali, M.M., 2007. Differential evolution algorithms using hybrid mutation. Comput. Optim. Appl. 37 (2), 231–246.
Kao, Y., Chen, C.-C., 2013. A differential evolution fuzzy clustering approach to machine cell formation. Int. J. Adv. Manuf. Technol. 65 (1247), 9–12–1259.
Kapadi, M.D., Gudi, R.D., 2004. Optimal control of fed-batch fermentation involving multiple feeds using Differential Evolution. Process Biochem. 39 (11), 1709–1721.
Koutny, T., 2016. Using meta-differential evolution to enhance a calculation of a continuous blood glucose level. Comput. Methods Programs Biomed. 133, 45–54.
Kukkonen, S., Lampinen, J., 2005. GDE3: The third evolution step of generalized differential evolution. pp. 443–450.
Kumar, P., Kumar, S., Pant, M., Singh, V.P., 2012. Interpolation based mutation variants of differential evolution. Int. J. Appl. Evol. Comput. 3 (4), 34–50.
Kumar, P., Pant, M., 2013. Noisy source recognition in multi noise plants by differential evolution. In: Proceedings of the 2013 IEEE Symposium on Swarm Intelligence. SIS 2013-2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013, pp. 271–275.
Kumar, P., Pant, M., Abraham, A., 2011. Two enhanced differential evolution variants for solving global optimization problems. In: Proceedings of the 2011 3rd World Congress on Nature and Biologically Inspired Computing, NaBIC 2011, No. 1, pp. 201–206.
Kumar, S., Pant, M., Ray, A., 2011. Differential evolution embedded Otsu's method for optimized image thresholding. In: Proceedings of the 2011 World Congress on Information and Communication Technologies, WICT 2011, pp. 325–329.
Lai, J.C.Y., Leung, F.H.F., Ling, S.H., 2009. A new differential evolution with wavelet theory based mutation operation. In: 2009 IEEE Congress on Evolutionary Computation, CEC 2009, pp. 1116–1122.
Lampinen, J., Zelinka, I., 1999. Mechanical engineering design optimization by differential evolution. In: New Ideas in Optimization. Mc Graw-Hill, UK, pp. 127–146.
Le Hai, B., Ashida, T., Thawonmas, R., Rinaldo, F., Computer, I., 2012. A hybrid differential evolution method and its application to the physical travelling salesman problem. pp. 2–3.
Lee, M.-C., Cho, S.-B., 2012. Interactive differential evolution for image enhancement application in smart phone. In: 2012 IEEE Congr. Evol. Comput., pp. 1–6.
Li, X., Hu, C., Yan, X., 2013a. Chaotic differential evolution algorithm based on competitive coevolution and its application to dynamic optimization of chemical processes. Intell. Autom. Soft Comput. 19 (1), 85–98.
Li, H., Jiao, Y.C., Zhang, L., 2011. Hybrid differential evolution with a simplified quadratic approximation for constrained optimization problems. Eng. Optim. 43 (2), 115–134.
Li, Y., Wang, Y., Li, B., 2013b. A hybrid artificial bee colony assisted differential evolution algorithm for optimal reactive power flow. Int. J. Electr. Power Energy Syst. 52 (1), 25–33.
Li, X., Yin, M., 2012. Optimal synthesis of linear antenna array with composite differential evolution algorithm. Sci. Iran. 19 (6), 1780–1787.
Li, C., et al., 2008. Benchmark Generator for CEC 2009 Competition on Dynamic Optimization. Univ. Leicester, Univ. Birmingham, Nanyang Technol. Univertsity, pp. 1–13.
Li, G., et al., 2016. A novel hybrid differential evolution algorithm with modified CoDE and JADE. Appl. Soft Comput. J. 47, 577–599.
Li-bao, Deng, et al., 2016. A hybrid mutation scheme-based discrete differential evolution algorithm for multidimensional Knapsack problem. In: 2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC). pp. 1009–1014.
Liang, J.J., et al., Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report 201212.34 (2013): 281–295.
Liao, T.W., 2010. Two hybrid differential evolution algorithms for engineering design optimization. Appl. Soft Comput. J. 10 (4), 1188–1199.
Liao, J., Cai, Y., Chen, Y., Wang, T., Tian, H., 2014. Improving differential evolution with ring topology-based mutation operators. In: Proceedings - 2014 9th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2014, pp. 103–109.
Lin, Q., Zhu, Q., Huang, P., Chen, J., Ming, Z., Yu, J., 2015. A novel hybrid multi-objective immune algorithm with adaptive differential evolution. Comput. Oper. Res. 62, 95–111.
Liu, H., Cai, Z., Wang, Y., 2010. Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl. Soft Comput. J. 10 (2), 629–640.
Liu, J., Lampinen, J., 2002. On setting the control parameter of the differential evolution method. In: Proceedings of the 8th International Conference on Soft Computing, MENDEL, pp. 11–18.
Liu, J., Lampinen, J., 2005. A fuzzy adaptive differential evolution algorithm. Soft Comput. 9 (6), 448–462.
Liu, F., Qi, Y., Xia, Z., Hao, H., 2009. A discrete differential evolution algorithm for the job shop scheduling problem. In: 2009 World Summit on Genetic and Evolutionary Computation, 2009 GEC Summit - Proceedings of the 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation, GEC'09, 2009, Table 1, pp. 879–882.
Long, W., Liang, X., Huang, Y., Chen, Y., 2013. A hybrid differential evolution augmented Lagrangian method for constrained numerical and engineering optimization. CAD Comput. Aided Des. 45 (12), 1562–1574.
Maciel, L., Gomide, F., Ballini, R., 2016. A differential evolution algorithm for yield curve estimation. Math. Comput. Simulation 129, 10–30.
Magoulas, G.D., Plagianakos, V.P., Vrahatis, M.N., 2004. Neural network-based colonoscopic diagnosis using on-line learning and differential evolution. Appl. Soft Comput. J. 4 (4), 369–379.
Mahdavi, S., 2017. Enhancing discrete differential evolution by conducting election.
Mallipeddi, R., Suganthan, P.N., 2010. Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies. In: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). In: LNCS, vol. 6466, pp. 71–78.
Mandal, K.K., Chakraborty, N., 2013. Parameter study of differential evolution based optimal scheduling of hydrothermal systems. J. Hydro-Environ. Res. 7 (1), 72–80.
Maulik, U., Saha, I., 2010. Automatic fuzzy clustering using modified differential evolution for image classification. IEEE Trans. Geosci. Remote Sens. 48 (9), 3503–3510.
Menchaca-Mendez, A., Coello, C.A.C., 2009. A new proposal to hybridize the nelder–mead method to a differential evolution algorithm for constrained optimization. In: 2009 IEEE Congr. Evol. Comput., pp. 2598–2605.
Meng, Z., Pan, J.S., Kong, L., 2018. Parameters with Adaptive Learning Mechanism (PALM) for the enhancement of Differential Evolution. Knowl-Based Syst. 141, 92–112.
Mesejo, P., Ugolotti, R., Di Cunto, F., Giacobini, M., Cagnoni, S., 2013. Automatic hippocampus localization in histological images using Differential Evolution-based deformable models. Pattern Recognit. Lett. 34 (3), 299–307.
Mlakar, U., Potočnik, B., Brest, J., 2016. A hybrid differential evolution for optimal multilevel image thresholding. Expert Syst. Appl. 65, 221–232.
Mohamed, A.W., 2015. An improved differential evolution algorithm with triangular mutation for global numerical optimization. Comput. Ind. Eng. 85, 359–375.
Mukherjee, R., Debchoudhury, S., Kundu, R., Das, S., Suganthan, P.N., 2013. Adaptive differential evolution with locality based crossover for dynamic optimization. In: 2013 IEEE Congress on Evolutionary Computation, CEC 2013, pp. 63–70.
Nakib, A., Daachi, B., Siarry, P., 2012. Hybrid differential evolution using low-discrepancy sequences for image segmentation. In: Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2012, pp. 634–640.
Nearchou, A.C., 2008. A differential evolution approach for the common due date early/tardy job scheduling problem. Comput. Oper. Res. 35 (4), 1329–1343.
Neri, F., Tirronen, V., 2010. Recent advances in differential evolution: A survey and experimental analysis. Artif. Intell. Rev. 33 (1–2), 61–106.
Noktehdan, A., Karimi, B., Husseinzadeh Kashan, A., 2010. A differential evolution algorithm for the manufacturing cell formation problem using group based operators. Expert Syst. Appl. 37 (7), 4822–4829.
Noman, N., Iba, H., 2008. Differential evolution for economic load dispatch problems. Electr. Power Syst. Res. 78 (8), 1322–1331.
Nwankwor, E., Nagar, A.K., Reid, D.C., 2013. Hybrid differential evolution and particle swarm optimization for optimal well placement. Comput. Geosci. 17 (2), 249–268.
Omran, M.G.H., Engelbrecht, A.P., 2006. Self-adaptive differential evolution methods for unsupervised image classification. In: 2006 IEEE Conference on Cybernetics and Intelligent Systems, pp. 966–973.
Omran, M.G.H., Engelbrecht, A.P., 2009. Free search differential evolution. In: 2009 IEEE Congress on Evolutionary Computation, CEC 2009, pp. 110–117.
Onwubolu, G., Davendra, D., 2006. Scheduling flow shops using differential evolution algorithm. European J. Oper. Res. 171 (2), 674–692.
Opara, K., Arabas, J., 2013. Censoring mutation in differential evolution. In: Proceedings of the 2013 IEEE Symposium on Differential Evolution, SDE 2013-2013 IEEE Symposium Series on Computational Intelligence, No. 2, SSCI 2013, pp. 54–60.
Ortiz, A., Cabrera, J.A., Nadal, F., Bonilla, A., 2013. Dimensional synthesis of mechanisms using differential evolution with auto-adaptive control parameters. Mech. Mach. Theory 64, 210–229.

Pan, Q.K., Wang, L., Gao, L., Li, W.D., 2011. An effective hybrid discrete differential evolution algorithm for the flow shop scheduling with intermediate buffers. Inf. Sci. (Ny). 181 (3), 668–685.

Pant, M., Ali, M., Singh, V.P., 2008. Differential evolution with parent centric crossover. In: Proceedings - EMS 2008 European Modelling Symposium, 2nd UKSim European Symposium on Computer Modelling and Simulation, pp. 141–146.

Pant, M., Ali, M., Singh, V.P., 2009. Differential evolution using quadratic interpolation for initializing the population. In: 2009 IEEE International Advance Computing Conference, No. March, IACC 2009, pp. 375–380.

Pant, M., Thangaraj, R., Grosan, C., Abraham, A., 2008. Hybrid differential evolution - Particle Swarm Optimization algorithm for solving global optimization problems. In: 2008 Third Int. Conf. Digit. Inf. Manag., pp. 18–24.

Parouha, R.P., Das, K.N., 2016. A memory based differential evolution algorithm for unconstrained optimization. Appl. Soft Comput. 38, 501–517.

Peñuñuri, F., Cab, C., Carvente, O., Zambrano-Arjona, M.A., Tapia, J.A., 2016. A study of the Classical Differential Evolution control parameters. Swarm Evol. Comput. 26, 86–96.

Pham, H.A., 2016. Truss optimization with frequency constraints using enhanced differential evolution based on adaptive directional mutation and nearest neighbor comparison. Adv. Eng. Softw. 102, 142–154.

Piotrowski, A.P., 2013. Adaptive memetic differential evolution with global and local neighborhood-based mutation operators. Inf. Sci. (Ny). 241, 164–194.

Powell, M.J.D., 1962. An iterative method for finding stationary values of a function of several variables. Comput. J. 5 (2), 147–151.

Prauzek, M., Krömer, P., Rodway, J., Musilek, P., 2016. Differential evolution of fuzzy controller for environmentally-powered wireless sensors. Appl. Soft Comput. J. 48, 193–206.

Qian, B., Wang, L., xian Huang, D., Wang, X., 2009. An effective hybrid DE-based algorithm for flow shop scheduling with limited buffers. Int. J. Prod. Res. 47 (1), 1–24.

Qin, A.K., Huang, V.L., Suganthan, P.N., 2009. Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. Evol. Comput. 13 (2), 398–417.

Qin, A.K., Suganthan, P.N., 2005. Self-adaptive differential evolution algorithm for numerical optimization. In: 2005 IEEE Congress on Evolutionary Computation, Vol. 2, pp. 1785–1791.

Qiu, X., Xu, J.X., Tan, K.C., Abbass, H.A., 2016. Adaptive cross-generation differential evolution operators for multiobjective optimization. IEEE Trans. Evol. Comput. 20 (2), 232–244.

Rahnamayan, S., Tizhoosh, H.R., 2008. Image thresholding using micro Opposition-Based Differential Evolution (Micro-ODE). In: 2008 IEEE Congress on Evolutionary Computation, CEC 2008, pp. 1409–1416.

Rahnamayan, S., Tizhoosh, H.R., Salama, M.M.A., 2007. A novel population initialization method for accelerating evolutionary algorithms. Comput. Math. Appl. 53 (10), 1605–1614.

Rahnamayan, S., Tizhoosh, H.R., Salama, M.M., 2008. Opposition-based differential evolution. Stud. Comput. Intell. 143 (1), 155–171.

Raj, C.T., Srivastava, S.P., Agarwal, P., 2008. Differential evolution based optimal control of induction motor serving to textile industry. Int. J. Comput. Sci. 35 (2), 1–8.

Raj, C.T., Thangaraj, R., Pant, M., Bouvry, P., Abraham, A., 2012. Design optimization of induction motors with differential evolution algorithms with an application in textile spinning. Appl. Artif. Intell. 26 (9), 809–831.

Rane, T.D., Dewri, R., Ghosh, S., Mitra, K., Chakraborti, N., 2005. Modeling the recrystallization process using inverse cellular automata and genetic algorithms: Studies using differential evolution. J. Phase Equilib. Diffus. 26 (4), 311–321.

Reddy, K.N., Ravi, V., 2013. Differential evolution trained kernel principal component WNN and kernel binary quantile regression: Application to banking. Knowl.-Based Syst. 39, 45–56.

Reed, H.M., Nichols, J.M., Earls, C.J., 2013. A modified differential evolution algorithm for damage identification in submerged shell structures. Mech. Syst. Signal Process. 39 (1–2), 396–408.

Reynoso-Meza, G., Sanchis, J., Blasco, X., Herrero, J.M., 2011. Hybrid de algorithm with adaptive crossover operator for solving real-world numerical optimization problems. In: 2011 IEEE Congr. Evol. Comput. CEC 2011, Vol. 1, No. 2, pp. 1551–1556.

Rochenberg, I., 1973. Evolution Strategy: Optimization of Technical Systems By Means of Biological Evolution. Fromman-Holzboog, Stuttgart, Germany.

Ronkkonen, J., Kukkonen, S., Price, K.V., 2005. Real-Parameter Optimization with Differential Evolution. In: 2005 IEEE Congress on Evolutionary Computation, Vol. 1, pp. 506–513.

Rosenbrock, H.H., 1960. An automatic method for finding the greatest or least value of a function. Comput. J. 3 (3), 175–184, https://doi.org/10.1093/comjnl/3.3.175.

Sabat, S.L., Kumar, K.S., Rangababu, P., 2011. Differential evolution algorithm for motion estimation. Multi-Discip. Trends Artif. Intell. 7080, 309–316.

Sacco, W.F., De Moura Meneses, A.A., Henderson, N., 2013. Some studies on differential evolution variants for application to nuclear reactor core design. Prog. Nucl. Energy 63, 49–56.

Sacco, W.F., Henderson, N., 2014. Differential evolution with topographical mutation applied to nuclear reactor core design. Prog. Nucl. Energy 70, 140–148.

Sahoo, D.K., Sahu, R.K., Sekhar, G.T.C., Panda, S., 2016. A novel modified differential evolution algorithm optimized fuzzy proportional integral derivative controller for load frequency control with thyristor controlled series compensator. J. Electr. Syst. Inf. Technol. (2016).

Sakr, W.S., EL-Sehiemy, R.A., Azmy, A.M., 2017. Adaptive differential evolution algorithm for efficient reactive power management. Appl. Soft Comput. J. 53, 336–351.

Salehinejad, H., Rahnamayan, S., Tizhoosh, H.R., 2017. Micro-differential evolution: Diversity enhancement and a comparative study. Appl. Soft Comput. J. 52, 812–833.

Sallam, K.M., Elsayed, S.M., Sarker, R.A., Essam, D.L., 2017. Two-phase differential evolution framework for solving optimization problems. In: 2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016.

Salman, A., Engelbrecht, A.P., Omran, M.G.H., 2007. Empirical analysis of self-adaptive differential evolution. European J. Oper. Res. 183 (2), 785–804.

Sauer, J.G., Coelho, S., 2008. Discrete differential evolution with local search to solve the traveling salesman problem : Fundamentals and Case studies, Search.

Schumer, M.A., Steiglitz, K., 1968. Adaptive step size random search. IEEE Trans. Automat. Control 13 (3), 270–276.

Segura, C., Coello Coello, C.A., Hernández-Díaz, A.G., 2015. Improving the vector generation strategy of differential evolution for large-scale optimization. Inf. Sci. (Ny). 323, 106–129.

Segura, C., Coello Coello, C.A., Segredo, E., León, C., 2014. On the adaptation of the mutation scale factor in differential evolution. Optim. Lett. 9 (1), 189–198.

Sethanan, K., Pitakaso, R., 2016a. Improved differential evolution algorithms for solving generalized assignment problem. Expert Syst. Appl. 45, 450–459.

Sethanan, K., Pitakaso, R., 2016b. Differential evolution algorithms for scheduling raw milk transportation. Comput. Electron. Agric. 121, 245–259.

Sharma, T.K., Pant, M., 2011a. Self adaptive mutation step size in differential evolution algorithm. In: Proceedings of the 2011 World Congress on Information and Communication Technologies, WICT 2011, pp. 171–175.

Sharma, T.K., Pant, M., 2011b. Differential operators embedded artificial bee colony algorithm. In: Modeling Applications and Theoretical Innovations in Interdisciplinary Evolutionary Computation, Vol. 2, No. 3. pp. 149–163.

Shih, M.Y., Conde Enríquez, A., Hsiao, T.Y., Torres Treviño, L.M., 2017. Enhanced differential evolution algorithm for coordination of directional overcurrent relays. Electr. Power Syst. Res. 143, 365–375.

Silva, D.O., Vieira, L.G.M., Lobato, F.S., Barrozo, M.A.S., 2012. Optimization of the design and performance of hydrocyclones by Differential Evolution technique. Chem. Eng. Process. Process Intensif. 61, 1–7.

Sindhya, K., Ruuska, S., Haanpää, T., Miettinen, K., 2011. A new hybrid mutation operator for multiobjective optimization with differential evolution. Soft Comput. 15 (10), 2041–2055.

Sivasubramani, S., Swarup, K.S., 2012. Multiagent based differential evolution approach to optimal power flow. Appl. Soft Comput. J. 12 (2), 735–740.

Sotiroudis, S.P., Goudos, S.K., Gotsis, K.A., Siakavara, K., Sahalos, J.N., 2013. Application of a composite differential evolution algorithm in optimal neural network design for propagation path-loss prediction in mobile communication systems. IEEE Antennas Wirel. Propag. Lett. 12, 364–367.

Srinivasa Reddy, A., Vaisakh, K., 2013. Shuffled differential evolution for large scale economic dispatch. Electr. Power Syst. Res. 96, 237–245.

Stanarevic, N., 2012. Hybridizing artificial bee colony (ABC) algorithm with differential evolution for large scale optimization problems. Int. J. Math. Comput. Simul. 6 (1), 194–202.

Storn, Rainer, 1996. On the usage of differential evolution for function optimization. In: Proceedings of North American Fuzzy Information Processing. IEEE.

Storn, R., Price, K., 1996. Minimizing the real functions of the ICEC'96 contest by differential evolution. In: Proc. IEEE Conf. Evol. Comput., pp. 842–844.

Storn, R., Price, K., 1997. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11 (4), 341–359.

Subudhi, B., Jena, D., 2008. Differential evolution and levenberg marquardt trained neural network scheme for nonlinear system identification. Neural Process. Lett. 27 (3), 285–296.

Suganthan, P.N., et al., 2005. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Tech. Report, Nanyang Technol. Univ. Singapore, 2005 KanGAL Rep. 2005005, IIT Kanpur, India, (2014).

Sun, Y., 2017. Symbiosis co-evolutionary population topology differential evolution. In: Proceedings - 12th International Conference on Computational Intelligence and Security, No. 1, CIS 2016, pp. 530–533.

Sun, G., Cai, Y., 2017. A novel neighborhood-dependent mutation operator for differential evolution. In: Proceedings - 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017, Vol. 1, pp. 837–841.

Sun, Y., Li, Y., Liu, G., Liu, J., 2012. A novel differential evolution algorithm with adaptive of population topology. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). In: LNCS, vol. 7473, pp. 531–538.

Takahama, T., Sakai, S., 2006. Solving nonlinear constrained optimization problems by the ε constrained differential evolution. In: 2006 IEEE Int. Conf. Syst. Man, Cybern., Vol. 198, No. 2, pp. 51–72.

Takahama, Tetsuyuki, Sakai, Setsuko, 2010. Constrained Optimization By the ε Constrained Differential Evolution with an Archive and Gradient-Based Mutation. IEEE.

Tanabe, R., Fukunaga, A., 2013. Success-history based parameter adaptation for Differential Evolution. In: 2013 IEEE Congr. Evol. Comput., No. 3, CEC2013, pp. 71–78.

Tanabe, R., Fukunaga, A.S., 2014. Improving the search performance of SHADE using linear population size reduction. In: Proc. 2014 IEEE Congr. Evol. Comput., CEC2014, pp. 1658–1665.

Tasgetiren, M.F., Pan, Q.K., Liang, Y.C., 2009. A discrete differential evolution algorithm for the single machine total weighted tardiness problem with sequence dependent setup times. Comput. Oper. Res. 36 (6), 1900–1915.

Teo, J., 2005. With self-adaptive populations. pp. 1284–1290.

Teo, J., 2006. Exploring dynamic self-adaptive populations in differential evolution. Soft Comput. 10 (8), 673–686.

Thangaraj, R., Pant, M., Abraham, A., 2010. New mutation schemes for differential evolution algorithm and their application to the optimization of directional over-current relay settings. Appl. Math. Comput. 216 (2), 532–544.

Tian, M., Gao, X., 2018. An improved differential evolution with information intercrossing and sharing mechanism for numerical optimization. Swarm Evol. Comput. 1 (2017), 1–21.

Törn, A., Viitanen, S., 1994. Topographical global optimization using pre-sampled points. J. Global Optim. 5 (3), 267–276.

Ustun, D., Akdagli, A., 2017. A study on the performance of the hybrid optimization method based on artificial bee colony and differential evolution algorithms. pp. 4–8.

Uyar, A.Ş., Türkay, B., Keleş, A., 2011. A novel differential evolution application to short-term electrical power generation scheduling. Int. J. Electr. Power Energy Syst. 33 (6), 1236–1242.

Wang, Y., Cai, Z., Zhang, Q., 2011. Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans. Evol. Comput. 15 (1), 55–66.

Wang, Y., Cai, Z., Zhang, Q., 2012. Enhancing the search ability of differential evolution through orthogonal crossover. Inf. Sci. (Ny). 185 (1), 153–177.

Wang, B.C., Li, H.X., Li, J.P., Wang, Y., 2018b. Composite differential evolution for constrained evolutionary optimization. IEEE Trans. Syst. Man Cybern. 1–14.

Wang, Y., Li, B., Weise, T., 2010b. Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems. Inf. Sci. (Ny). 180 (12), 2405–2420.

Wang, Y., Li, B., Weise, T., 2010c. Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems. Inf. Sci. (Ny). 180 (12), 2405–2420.

Wang, L., Pan, Q.-K., Suganthan, P.N., Wang, W.-H., Wang, Y.-M., 2010a. A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems. Comput. Oper. Res. 37, 509–520, Hybrid Metaheuristics.

Wang, H., Rahnamayan, S., Sun, H., Omran, M.G.H., 2013. Gaussian bare-bones differential evolution. IEEE Trans. Cybern. 43 (2), 634–647.

Wang, X., Tang, L., 2016. An adaptive multi-population differential evolution algorithm for continuous multi-objective optimization. Inf. Sci. (Ny) 348, 124–141.

Wang, L., Wang, T., Wu, J., Chen, G., 2017b. Multi-objective differential evolution optimization based on uniform decomposition for wind turbine blade design. Energy 120, 346–361.

Wang, X., Xu, G., 2011. Hybrid differential evolution algorithm for traveling salesman problem. Proced. Eng. 15, 2716–2720.

Wang, Y., Xu, B., Sun, G., Yang, S., 2017a. A two-phase differential evolution for uniform designs in constrained experimental domains. IEEE Trans. Evol. Comput. 21 (5), 665–680.

Wang, Y., Yin, D.Q., Yang, S., Sun, G., 2018a. Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints. IEEE Trans. Cybern. 1–15.

Wang, J., Zhang, W., Zhang, J., Member, S., 2016. Cooperative differential evolution with multiple populations for multiobjective optimization. IEEE Trans. Cybern. 46 (12), 2848–2861.

Wei, W., Wang, J., Tao, M., 2015. Constrained differential evolution with multiobjective sorting mutation operators for constrained optimization. Appl. Soft Comput. J. 33, 207–222.

Worasucheep, C., 2015. An opposition-based hybrid artificial bee colony with differential evolution. In: IEEE Congress on Evolutionary Computation, Vol. 00, No. c, CEC, pp. 2611–2618.

Wu, G., Mallipeddi, R., Suganthan, P.N., Wang, R., Chen, H., 2016. Differential evolution with multi-population based ensemble of mutation strategies. Inf. Sci. (Ny). 329, 329–345.

Yan, X., Shi, H., 2011. A Hybrid Algorithm Based on Particle Swarm Optimization and Group Search Optimization, Vol. 1. pp. 13–17.

Yang, X.-S., 2009. Firefly Algorithms for Multimodal Optimization. Springer, Berlin, Heidelberg, pp. 169–178.

Yeh, M.F., Lu, H.C., Chen, T.H., Leu, M.S., 2017. Modified Gaussian barebones differential evolution with hybrid crossover strategy. In: Proceedings - International Conference on Machine Learning and Cybernetics, Vol. 1, pp. 7–12.

Yi, W., Zhou, Y., Gao, L., Li, X., Mou, J., 2016. An improved adaptive differential evolution algorithm for continuous optimization. Expert Syst. Appl. 44, 1–12.

Yildiz, A.R., 2013. A new hybrid differential evolution algorithm for the selection of optimal machining parameters in milling operations. Appl. Soft Comput. J. 13 (3), 1561–1566.

Yu, X., Cai, M., Cao, J., 2015. A novel mutation differential evolution for global optimization. J. Intell. Fuzzy Syst. 28 (3), 1047–1060.

Yu, X., Yu, X., Lu, Y., Yen, G.G., Cai, M., 2018. Differential evolution mutation operators for constrained multi-objective optimization. Appl. Soft Comput. 67, 452–466.

Yuan, X., Cao, B., Yang, B., Yuan, Y., 2008. Hydrothermal scheduling using chaotic hybrid differential evolution. Energy Convers. Manage. 49 (12), 3627–3633.

Yuan, X., Su, A., Nie, H., Yuan, Y., Wang, L., 2009. Application of enhanced discrete differential evolution approach to unit commitment problem. Energy Convers. Manage. 50 (9), 2449–2456.

Zaharie, D., 2007. A comparative analysis of crossover variants in differential evolution. Comput. Sci. Inf. Technol. 171–181.

Zamuda, A., Brest, J., 2015. Self-adaptive control parameters' randomization frequency and propagations in differential evolution. Swarm Evol. Comput. 25, 72–99.

Zamuda, A., Brest, J., Bošković, B., Žumer, V., 2011. Differential evolution for parameterized procedural woody plant models reconstruction. Appl. Soft Comput. J. 11 (8), 4904–4912.

Zeng, X., Wong, W.K., Leung, S.Y.S., 2012. An operator allocation optimization model for balancing control of the hybrid assembly lines using Pareto utility discrete differential evolution algorithm. Comput. Oper. Res. 39 (5), 1145–1159.

Zhang, J., Avasarala, V., Sanderson, A.C., Mullen, T., 2008. Differential evolution for discrete optimization: An experimental study on combinatorial auction problems. pp. 2794–2800.

Zhang, X., Chen, W., Dai, C., Cai, W., 2010. Dynamic multi-group self-adaptive differential evolution algorithm for reactive power optimization. Int. J. Electr. Power Energy Syst. 32 (5), 351–357.

Zhang, C., Chen, J., Xin, B., 2013. Distributed memetic differential evolution with the synergy of Lamarckian and Baldwinian learning. Appl. Soft Comput. J. 13 (5), 2947–2959.

Zhang, Z., Dong, Y., Gao, T., 2017. A hybrid method based on cuckoo search and krill herd optimization with differential evolution. In: Proceedings - 13th Web Information Systems and Applications Conference, WISA 2016 - In conjunction with 1st Symposium on Big Data Processing and Analysis, BDPA 2016 and 1st Workshop on Information System Security, No. December, ISS 2016, pp. 138–143.

Zhang, L., Jiao, Y., Li, H., Zhang, F., 2009b. Antenna optimization by hybrid differential evolution.

Zhang, C., Ning, J., Lu, S., Ouyang, D., Ding, T., 2009a. A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization. Oper. Res. Lett. 37 (2), 117–122.

Zhang, H., Rangaiah, G.P., 2012. An efficient constraint handling method with integrated differential evolution for numerical and engineering optimization. Comput. Chem. Eng. 37, 74–88.

Zhang, J., Sanderson, A.C., 2009. JADE: Adaptive differential evolution with optional external archive. IEEE Trans. Evol. Comput. 13 (5), 945–958.

Zhang, X., Yuen, S.Y., 2015. A directional mutation operator for differential evolution algorithms. Appl. Soft Comput. J. 30, 529–548.

Zhao, S.Z., Suganthan, P.N., 2013. Empirical investigations into the exponential crossover of differential evolutions. Swarm Evol. Comput. 9, 27–36.

Zhao, Z., Yang, J., Hu, Z., Che, H., 2016. A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric Latin hypercube design for unconstrained optimization problems. European J. Oper. Res. 250 (1), 30–45.

Zhou, Y., Li, X., Gao, L., 2013. A differential evolution algorithm with intersect mutation operator. Appl. Soft Comput. J. 13 (1), 390–401.

Zhou, S., Liu, M., Chen, H., Li, X., 2016. An effective discrete differential evolution algorithm for scheduling uniform parallel batch processing machines with non-identical capacities and arbitrary job sizes. Int. J. Prod. Econ. 179, 1–11.

Zhou, X.G., Zhang, G.J., 2018. Differential evolution with underestimation-based multimutation strategy. IEEE Trans. Cybern. 1–12.

Zhou, Yin-Zhi, et al., 2017. Adaptive differential evolution with sorting crossover rate for continuous optimization problems. IEEE Trans. Cybern. 47 (9), 2742–2753.

Zhu, W., Tang, Y., Fang, J.A., Zhang, W., 2013. Adaptive population tuning scheme for differential evolution. Inf. Sci. (Ny). 223 (2999), 164–191.

Zielinski, K., Laur, R., 2006. Constrained Single-Objective Optimization Using Differential Evolution. In: 2006 IEEE International Conference on Evolutionary Computation, pp. 223–230.

Zielinski, K., Weitkemper, P., Laur, R., Kammeyer, K.-D., 2006. Parameter study for differential evolution using a power allocation problem including interference cancellation, In: 2006 IEEE International Conference on Evolutionary Computation, No. 2, pp. 1857–1864.

ZorarpacI, E., Özel, S.A., 2016. A hybrid approach of differential evolution and artificial bee colony for feature selection. Expert Syst. Appl. 62, 91–103.

Zou, D., Gao, L., 2012. An efficient improved differential evolution algorithm. In: Chinese Control Conference, CCC, pp. 2385–2390.

Zou, D., Wu, J., Gao, L., Li, S., 2013. A modified differential evolution algorithm for unconstrained optimization problems. Neurocomputing 120, 469–481.