Alexandria Engineering Journal (2022) 61, 3831–3872

HOSTED BY

ELSEVIER

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

Differential evolution: A recent review based on state-of-the-art works

Mohamad Faiz Ahmad [a], **Nor Ashidi Mat Isa** [a,*], **Wei Hong Lim** [b], **Koon Meng Ang** [b]

[a] *School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Pulau Pinang, Malaysia*
[b] *Faculty of Engineering, Technology and Built Environment, UCSI University, 56000 Cheras, Kuala Lumpur, Malaysia*

Received 30 April 2021; revised 30 August 2021; accepted 10 September 2021
Available online 23 September 2021

KEYWORD

Metaheuristics;
Differential Evolution;
Initialisation;
Mutation;
Crossover;
Selection;
Optimisation

Abstract Differential evolution (DE) is a popular evolutionary algorithm inspired by Darwin's theory of evolution and has been studied extensively to solve different areas of optimisation and engineering applications since its introduction by Storn in 1997. This study aims to review the massive progress of DE in the research community by analysing the 192 articles published on this subject from 1997 to 2021, particularly studies in the past five years. The methodology used to search for relevant DE papers and an overview of the original DE are firstly explained. Recent advances in the modifications proposed to enhance the effectiveness and efficiency of the original DE are reviewed by analysing the strengths and weaknesses of each published work, followed by the potential applications of these DE variants in solving different real-world engineering problems. In contrast to most existing DE review papers, additional analyses are performed in this survey by investigating the impacts of various parameter settings on given DE variants to identify their optimal values required for solving certain problem classes. The qualities of modifications incorporated into selected DE variants are also evaluated by measuring the performance gains achieved in terms of search accuracy and/or efficiency against the original DE. The additional surveys conducted in this study are anticipated to provide more insightful perspectives for both beginners and experts of DE research, enabling their better understanding about current research trends and new motivations to outline appropriate strategic planning for future development works.

1. Introduction

Different sophisticated engineering systems have been designed and deployed in various industries in response to the phenomenal changes brought by the Fourth Industrial Revolution [1,2]. Given the ubiquitous nature of optimisation, most of these real-world engineering systems can be formulated as complex models that consist of challenging characteristics

* Corresponding author.
E-mail address: ashidi@usm.my (N.A.M. Isa).

Peer review under responsibility of Faculty of Engineering, Alexandria University.

https://doi.org/10.1016/j.aej.2021.09.013
1110-0168

such as nonlinearity, multimodality, discontinuousness and non-differentiability. To solve these complex models successfully, a decision-making process known as optimisation is generally used to search for the best combinations of decision variables that can maximise or minimise the predefined objective functions while satisfying all technical and non-technical constraints within a reasonable period of time. Conventionally, numerous mathematical programming methods such as linear programming [3], dynamic programming [4] and Newton's methods [5] were adopted by practitioners to solve these optimisation problems. Each of these conventional optimisers can perform well in certain types of problems only and it can be challenging to select an appropriate optimiser without having good priori knowledge on the characteristics of optimisation problems to be tackled. In addition, these conventional optimisers suffer from common drawbacks such as limited global strength, poor guessing of the initial solution and strong dependency on gradient information, which can further restrict their effectiveness in tackling various modern engineering optimisation problems that have increasing complexity. In view of the limitations of conventional optimisation methods, there is an urgent need to develop generic and yet more intelligent schemes that are able to solve different types of complex optimisation problems effectively without having to know the nature of a given problem a priori.

Metaheuristic search algorithms (MSAs) are envisioned as promising candidates to solve challenging modern optimisation problems by leveraging their search mechanisms inspired by different natural phenomena. These MSAs are able to perform searches with different levels of exploration and exploitation strengths during the optimisation process to locate the global or near-global optimum solution. Exploration is able to promote population diversity because it involves the process of discovering diverse solutions within the search space, while exploitation involves the refinement of information discovered so far by focusing on the search process within the vicinities of best solutions. The proper balancing of exploration and exploitation searches serves as a cornerstone for all MSAs to solve different types of optimisation problems successfully. In contrast to mathematical programming methods, MSAs in general are more flexible and able to locate the near-global optima of given optimisation problems more efficiently without requiring substantial modifications of algorithmic frameworks and the derivative information of given problems. The stochastic nature of MSAs also enables them to exhibit better robustness in handling local entrapment issues that are commonly encountered in real-world global optimisation problems. Given MSAs' desirable features such as simplicity, flexibility, robustness and efficiency, trends in artificial intelligence fields have been emerging in the past decades to develop new MSAs with enhanced search performances to solve different types of optimisation problems.

Depending on the sources of inspiration used to generate new solutions, the existing MSAs can be divided into four major categories [6]: (a) evolutionary algorithms (EAs), (b) swarm intelligence algorithms, (c) human-based algorithms and (d) physics-based algorithms. In general, the development of EAs is motivated by Darwin's theory of evolution and the 'survival of the fittest' concept. Different genetic operations such as crossover, mutation and selection were utilised by these EAs in producing new offspring solutions with better quality. Swarm intelligence algorithms are inspired by the collective behaviours of animals or insects such as searching for food sources and finding mates for reproduction purposes. Referring to the local information and interaction with search environments, intelligent behaviours can be demonstrated by these swarm members in a decentralised manner. For human-based algorithms, their search operators emulate different human activities or any human behaviour such as thinking, learning, talking and teaching. Finally, physics-based algorithms refer to those with search mechanisms inspired by various types of physical laws such as thermodynamics, electromagnetism, trigonometry and gravity. Table 1 presents the taxonomy of MSAs based on their source of inspiration and some notable examples associated with each branch of the MSA.

1.1. Overview of differential evolution

Among MSAs that were developed in the past few decades, differential evolution (DE) proposed by Storn et al. [30] is considered one of the most popular optimisers to solve complex optimisation problems. DE belongs to the EA family and is a population-based method that is widely used to solve various types of optimisation problems. It generates new offspring by recombining solutions under certain conditions, unlike other EAs that produce offspring by perturbing the solutions with scaled difference vectors. The current individual solution will be replaced if it is outperformed by the new offspring solution [31]. DE is considered a robust and simple algorithm because its search process is governed by few algorithm-specific parameters, such as scaling factor and crossover rate. Similar to other EAs, DE can produce new offspring solutions through three mechanisms: mutation, crossover and selection. Mutation and crossover are commonly observed to have a greater impact on the algorithm's search performance [32].

Although substantial amounts of MSAs inspired by different natural phenomena were proposed in the past decades, DE remains one of the most popular MSAs used by researchers

Table 1 Taxonomy of metaheuristic search algorithms.

MSAs	Algorithm name
Evolutionary Algorithms	Evolutionary programming (EP) [7], genetic algorithm (GA) [8], co-evolving algorithm (CEA) [9], genetic programming (GP) [10], differential evolution (DE) [11], human evolutionary model (HEM) [12], bio-inspired optimization (BIO) [13]
Swarm Intelligence Algorithms	Particle swarm optimization (PSO) [14], ant colony optimization (ACO) [15], artificial bee colony (ABC) [16], island bat algorithm (iBA) [17], whale optimization algorithm (WOA) [18], firefly algorithm (FA) [19]
Human-based Algorithms	Jaya algorithm (JA) [20], human-inspired algorithm (HIA) [21], teaching–learning-based optimization (TLBO) [22], socio evolution and learning optimization (SELO) [23], cognitive behavior optimization (CBO) [24]
Physics-based Algorithms	Gravitational search algorithm (GSA) [25], electro-magnetism optimization (EMO) [26], multi-verse optimizer (MVO) [27], sine cosine algorithm (SCA) [28], nuclear reaction optimization (NRO) [29]

and practitioners to tackle diverse sets of real-world optimisation problems due to its several competitive advantages. Firstly, the implementation of DE is simpler and more straightforward than that of most other MSAs [33]. This desirable feature enables those practitioners that might not have strong programming competency to make simple adjustments on the coding of DE to solve their domain-specific problems. Secondly, despite its simplicity, DE is able to demonstrate more promising optimisation ability than other MSAs in solving diverse types of optimisation problems that have challenging features such as nonlinearity, multimodal and non-separability. Thirdly, different DE variants have emerged as the top three best-performing optimisers in most Congress of Evolutionary Computation (CEC) competitions since 2005 [34], implying the potential of DE variants to solve different real-world applications with competitive search accuracy, search robustness and convergence speed. Lastly, compared with other MSAs that might also perform well in challenging optimisation problems, DE has the more desirable feature of low space complexity [33]. In other words, DE has better scalability than some existing MSAs in handling large-scale and computational expensive optimisation problems due to its lower storage requirements.

1.2. Existing review works on differential evolution and their limitations

Substantial efforts have been made by several researchers to review the massive progresses made in research areas of DE from different perspectives since its conceptualisation in 1997 [33–39]. Neri and Tirronen [35] are one of the pioneers in reviewing the progresses of DE research, focusing on the survey of various modified DE structures with additional components that were published up to 2009 and their performance evaluation in solving selected conventional and rotated problems with different dimensional sizes. A few crucial aspects that were not highlighted in their review work include engineering applications, analysis of current trends, open challenges and future directions of DE research. Two review papers were published by Das and Suganthan [33] and by Das et al. [36] to rigorously survey the advances of DE research up to 2011 and 2016, respectively. The major topics covered in these review papers are modifications made on the existing DE variants to solve different types of optimisation environments and engineering applications. Other interesting issues such as theoretical analyses, parallelisation and future directions of DE were also discussed.

The scope of study for the review paper published by Jebaraj et al. [37] is relatively niche because it focused on the applications of DE to solve static and dynamic economic or emission dispatch problems that were published up to 2016. Problem formulation in terms of objective function as well as the equality and inequality constraints of various economic or emission dispatch problems were firstly described, followed by an overview of existing DE variants that were designed to solve these problems. In contrast to most existing works that surveyed modifications made on DE variants and their engineering applications, the review paper of Opara and Arabs [38] focused on theoretical analyses of DE works. Substantial amounts of theoretical studies such as convergence characteristic, computational complexity, population diversity and population dynamics models of DE published up to 2019 were discussed in this paper comprehensively. Javaid [39] also published a review paper in the same year, but it only covered the DE variants and their applications in energy management problems that were published up to 2016. Recently, Bilal et al. [34] published a review paper that covered the recent progress of DE research published up to 2018. Apart from the modifications made of original DE, this review paper also performed comprehensive bibliometric analysis of DE to report its publication statistics on the basis of different journal quartiles and publishers.

Although numerous DE review papers have been published since the last decade, these works have their unique strengths and limitations. First, most DE review papers such as [33,35,36] only covered the relevant works published up to 2016. Bilal et al. [34] surveyed the recent advances of DE variants up to 2018, yet the engineering applications of DE were not up to the date and only covered until 2013. Certain review papers such as [37–39] have a relatively narrower scope of study and might therefore have limited readership among novice researchers or other interested parties that intend to know more about DE from more fundamental and broader perspectives. Some review papers such as [34,37–39] did not describe the methodology, databases and choice of keywords used for data collection. In fact, the presence of this information is highly recommended to enable researchers who are performing similar surveys in other research areas. Finally, all the aforementioned DE review papers [33–39] did not perform any surveys to investigate the impacts of various parameter settings on given DE variants to find out their optimal values required for solving certain problem classes. The performance gains achieved by these modified DEs against the original DE in terms of search accuracy and/or efficiency when they are applied to solve given optimisation problems were also not further analysed.

1.3. Significance and contributions of this work

In this paper, we present an updated review based on the state-of-art DE works, aiming to address the limitations of existing DE review papers. Our review paper aims to cover broader scopes of DE's philosophy to generate greater readership among the researchers or any other intended parties with different competency levels. Initially, the systematic methodology, including adoption of database, choice of keywords, classification and verification of research articles used during the data collection, is described for the intended readers to have a better understanding of the origin of selected papers. Apart from an overview of basic search mechanisms (i.e. mutation, crossover and selection) in original DE, the recent advances of modifications made in enhancing the effectiveness and efficiency of DE are also covered up to 2021, where the strengths and limitations of each DE variants are critically discussed. In contrast to existing works, our review paper also performs additional analysis to investigate the impacts of different parameter settings on given DE variants to identify the best parameter values used for solving particular problem classes. Furthermore, our review paper investigates the qualities of modifications incorporated into selected DE variants by measuring the performance gains achieved in terms of search accuracy and/or efficiency against the original DE.

The additional survey analyses conducted in our review paper are anticipated to be able to offer more new insights to DE researchers with different competency levels in understanding the latest trends of DE research, hence providing them the motivation to outline appropriate strategic planning for their future development works.

The summary of the contributions of this review paper of DE can be presented as follows:

- Detailed descriptions are given of the systematic methodology used for data collection of related papers, including the definition of research, choice of keywords and online databases used for article searching, verification processes used for excluding irrelevant articles and research analyses required to summarise the findings from existing DE works.
- Recent progress made on the modifications of DE algorithms, including those innovations introduced into modified initialisation, mutation, crossover, selection and/or hybridisation schemes up to 2021, were described comprehensively and analysed critically. The latest engineering applications of DE variants, including those of niche areas such as solving differential equation systems, were also covered in our review paper.
- Additional survey analyses are conducted in this review paper to investigate the influences of different parameter settings on given DE variants to solve certain problem classes optimally. To the best of the authors' knowledge, a detailed survey of the optimal parameter settings for DE in different types of optimisation problems was not considered in any previous DE review paper even though the overall search performances of DE were governed by these control parameters.
- The qualities of modifications made on DE variants in terms of search accuracy and/or efficiency to solve given problems based on pre-specified system configurations are also measured and compared with the original DE. Although these types of analyses have not been conducted in any DE review paper so far, we anticipate that these analyses can be beneficial for both expert and novice researchers by offering them new insights in discovering new directions for the modifications of DE.
- Open research challenges from different perspectives are elaborated by referring to the additional survey analyses performed on the state-of-art DE variants published in recent years. The corresponding future research directions that have the potential to address each open research challenge are also unveiled at the end of this review paper.

The remaining sections of this paper are presented as follows: Section 2 presents the research methodology used to conduct a systematic literature survey of DE. Section 3 discusses the basic concepts of the traditional DE algorithm. Section 4 discusses the strategies used to enhance the search performance of DE through the modification of the initialisation, mutation, crossover, selection and hybrid schemes. The latest engineering applications of DE are also reported. Section 5 discusses the additional performance analyses of DE in terms of survey analysis, benchmark functions used, frequently used parameter settings, widely used performance measures and performance studies. Sections 6 and 7 present the open research challenges and future research directions, respectively. Lastly, Section 8 concludes this paper.

2. Research methodology

In this section, the methodology and procedures used to perform the systematic review of existing works related to DE are explained. As shown in Fig. 1, an organized research framework is developed and expressed in four major steps, namely, research definition, article search, article verification and research analysis.

First step: Research definition

Step 1.1: Define the research area. The research areas are the enhanced technique and applications of DE algorithms to solve various problems, as shown in Fig. 2.
Step 1.2: Define the research goal. The research goal is to provide a systematic research scheme and propose ideas to enhance the existing DE algorithms.
Step 1.3: Define the research scope. The research scope is to review the optimisation process of DE algorithm articles with full text.

Second step: Article search

Step 2.1: Define the search terms. Only articles published within 2010 to 2021 were selected as the main references. Other articles published before 2010 are still considered general references given that DE was initially conceptualised by Storn et al. [30] in 1997. The articles were gathered as references by using six key terms, as shown in Fig. 2. All these research areas are the main focuses of the review process presented in Section 4.
Step 2.2: Search in online databases. The online journal databases used for searching research articles based on the choices of keywords are listed as follows:

- IEEE Xplore
- Web of Science
- Taylor & Francis
- ScienceDirect
- Springer
- Wiley
- Scopus

Step 2.3: Select only English language articles. Only related articles written in English language were considered, and those with other languages were excluded from this review. A total of 192 articles were selected for the survey purpose.

Third step: Article verification

The articles gathered in the second step were thoroughly verified by all co-authors. Only the articles related to DE were selected.

Step 3.1: Filter and remove irrelevant articles. Articles that generally focused on the general nature-inspired based algorithms but did not have any significant relation with DE were rejected except for the 53 general research articles used to explain the basic concept of MSAs.
Step 3.2: Double verification process for each article. The remaining selected articles were determined if they lie in the areas of focus. In this review, our focused parts were exploitation and exploration processes, convergence rate, computation time, search accuracy and robustness of an algorithm, as shown in Fig. 3. However, other outstanding

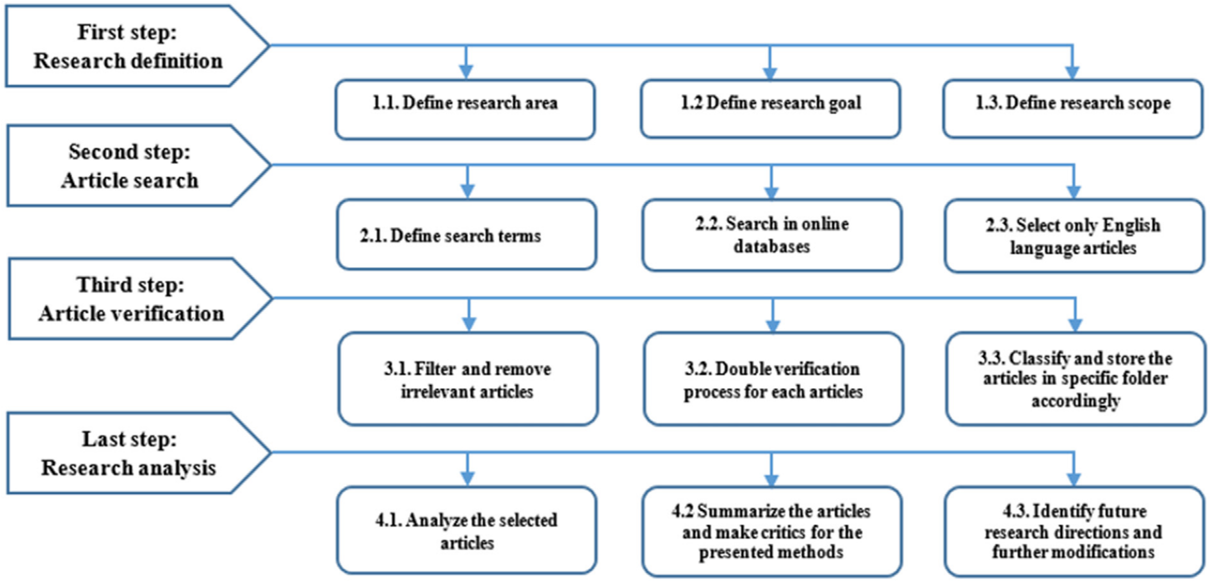

Fig. 1 Research framework.

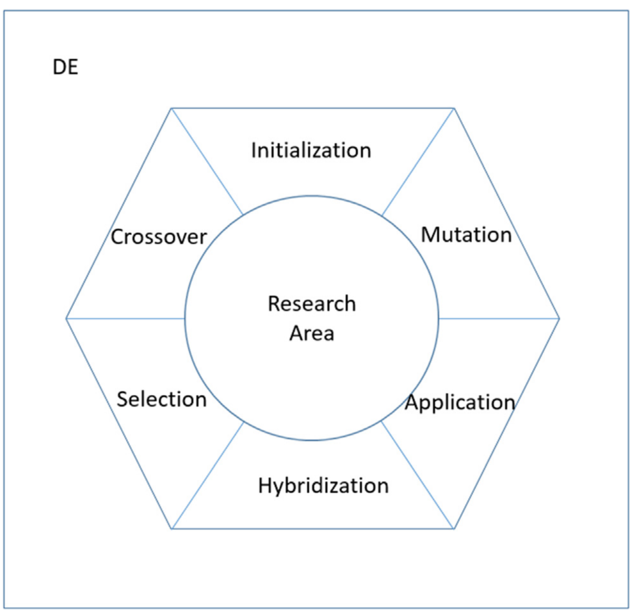

Fig. 2 Research areas.

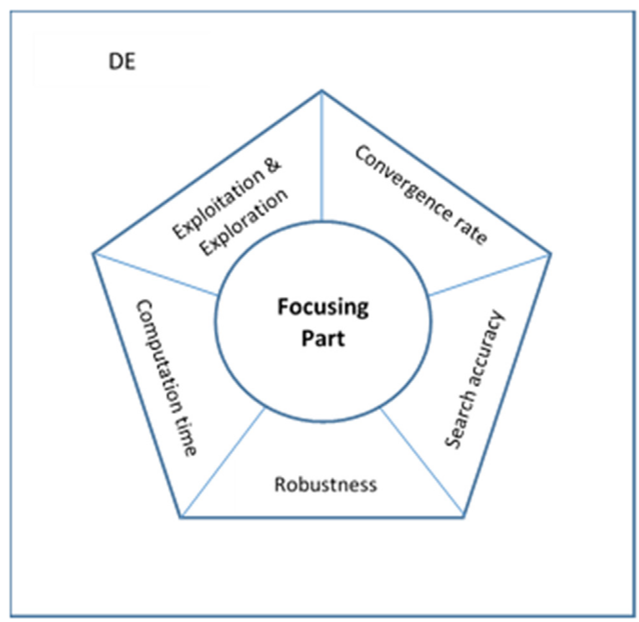

Fig. 3 Focused review parts.

achievements of researchers were also included as extra information that can be value-adding to this review paper. The articles were mainly selected from high-impact journals and several book chapters in explaining the basic theory of the DE algorithm. A few conference papers were also included as extra references for a general review of DE algorithms. Fig. 4 shows the tabulation of publications used in this review. On the basis of the chart, most of the chosen articles were published between 2016 and 2020 because the latest advances of DE and modification techniques proposed in these variants were surveyed. The DE and non-DE algorithms are designated with blue and orange colours, respectively, as shown in Fig. 4. Some non-DE research papers were included in this review process to provide additional information to readers such as the basic concept of other MSAs, as well as some open research challenges and future research directions that might be applicable to outline the future development plans of DE research.

Step 3.3: Classify and store the articles in a specific folder accordingly. The collected articles were classified into a specific folder according to the nature of their topics. This process was necessary because it ensures that the information related to our research is well organised.

Final step: Research analysis

Step 4.1: Analyse the selected articles. We analysed the selected articles to obtain important information such as

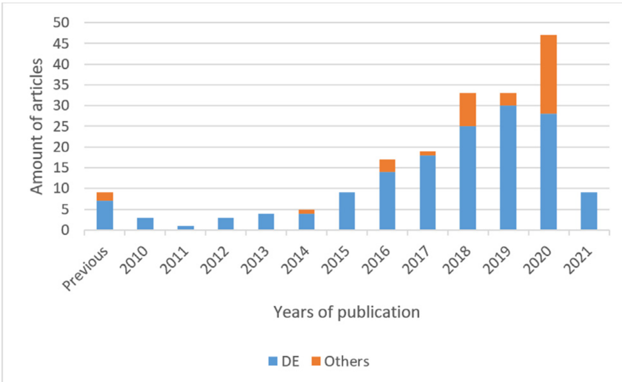

Fig. 4 Number of all cited articles.

modification strategies introduced in the original DE and their respective strengths and limitations, types of engineering applications, dimensional size of benchmark functions, optimisation results in terms of the search accuracy and efficiencies, parameter settings and qualities of modifications for different DE variants, and performance gains in terms of algorithm's accuracy and convergence speed.

Step 4.2: Summarise the articles and make critiques for the presented methods. We listed the strengths and limitations of the introduced techniques, as well as the qualities of modifications imposed in selected DE variants to solve different optimisation problems or any other engineering applications. Some articles are used as supplementary references to explain the fundamental theory and basic concept of DE algorithms.

Step 4.3: Identify future research directions and further modifications. Finally, we outlined the potential future development works for DE algorithms on the basis of the current research trends and open research challenges observed from the current survey analyses.

3. Basic concept of DE algorithms

The algorithmic framework of basic DE consists of four phases, namely, initialisation, mutation, crossover and selection, as shown in Fig. 5. Initialisation is a one-time process, while the remaining three mechanisms are repeated in the search process of DE in a D-dimensional solution space until the termination criteria are satisfied.

3.1. Initialisation

Initialisation is the first process that occurs in DE to search for a global optimum solution located in a D-dimensional of real parameter space. The initial solutions for a given multidimensional optimisation problem consist of NP real-valued parameter vectors, where NP represents the population size of DE. During the t-th iteration, each i-th individual solution of DE can be represented as a D-dimensional vector as (1)

$$X_i^t = (X_{i,1}, X_{i,2}, \cdots, X_{i,D}) \tag{1}$$

where $i = 1,2,\ldots,NP$. The initial population condition starts at $t = 0$. The initial candidate solutions can be generated during the initialisation stage on the basis of the lower and upper limit boundaries of the solution search space represented by (2) and (3), respectively, as follows:

$$X_{min} = (X_{min,1}, X_{min,2}, \cdots, X_{min,D}) \tag{2}$$

$$X_{max} = (X_{max,1}, X_{max,2}, \cdots, X_{max,D}) \tag{3}$$

For each i-th DE solution, the j-th dimensional component can be initialised by randomly generating a value in between the upper limit of $X_{max,j}$ and lower limit of $X_{min,j}$ as shown in (4)

$$X_{i,j}^{(0)} = X_{min,j} + rand_{i,j}[0,1]\left(X_{max,j} - X_{min,j}\right) \tag{4}$$

where $rand_{i,j}[0,1]$ is a uniform distribution that can generate any real value between 0 and 1.

3.2. Mutation

In biological terms, mutation is defined as an instant change of characteristic observed from a chromosome gene. In the context of evolutionary computation, mutation is a random perturbation process performed on selected decision variables. In DE philosophy, a mutant or donor vector denoted as Y_i^t is constructed from a mutation process on the basis of a given target vector of X_i^t [34,35]. Generally, the DE mutation strategy can be represented as the format 'DE/*/n' where n refers to the number of difference vectors involved and * represents the target vector considered during the mutation process. The search mechanisms of five commonly used mutation strategies in DE are represented as follows:

DE/rand/1:

$$Y_i^t = X_{r_1}^t + F\left(X_{r_2}^t - X_{r_3}^t\right) \tag{5}$$

DE/rand/2:

$$Y_i^t = X_{r_1}^t + F\left(X_{r_2}^t - X_{r_3}^t\right) + F\left(X_{r_4}^t - X_{r_5}^t\right) \tag{6}$$

DE/best/1:

$$Y_i^t = X_{best}^t + F\left(X_{r_1}^t - X_{r_2}^t\right) \tag{7}$$

DE/best/2:

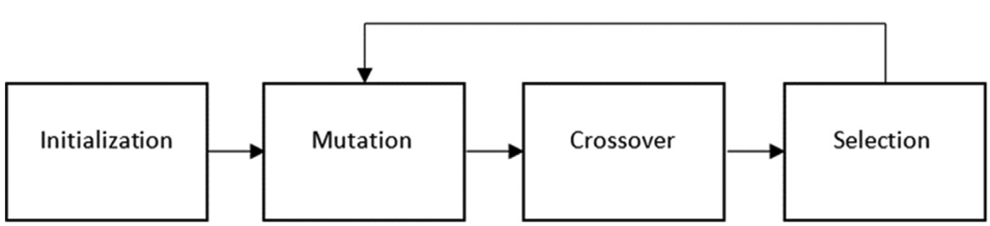

Fig. 5 DE consecutive phases.

$$Y_i^t = X_{best}^t + F\left(X_{r_1}^t - X_{r_2}^t\right) + F\left(X_{r_3}^t - X_{r_4}^t\right) \tag{8}$$

DE/current-to-best/1:

$$Y_i^t = X_i^t + F\left(X_{best}^t - X_i^t\right) + F\left(X_{r_1}^t - X_{r_2}^t\right) \tag{9}$$

where r_1 is the population index of the DE solution selected as the base vector; r_2, r_3, r_4 and r_5 are the population indices of DE solutions randomly selected to construct the mutant vector, where r_1, r_2, r_3, r_4, $r_5 \in [1, NP]$ and $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$; X_{best}^t imply that the best individual solution in the DE population is selected as the target vector; and F is a scaling factor used to control the mutation process and has a value in the range between [0,1]. Choosing the appropriate value for F is crucial to achieve proper balancing of exploration and exploitation searches of the algorithm to prevent undesirable drawbacks such as premature convergence or slow convergence speed.

3.3. Crossover

In this phase, both the mutant and target vectors cross their components together in a probabilistic manner to produce a trial vector (offspring). This crossover process allows the target solution to inherit the attributes of the donor solution or mutant. Two commonly used crossover operators are known as uniform crossover and exponential crossover. The uniform crossover scheme is controlled by a crossover rate (CR) that has a value between [0,1]. The trial solution generated by uniform crossover can be defined in (10) as follows:

$$Z_i^t = \begin{cases} Y_{i,j}^t & \text{if } rand_{i,j}[0,1] \leq CR \text{ or } j = k \\ X_{i,j}^t & \text{Otherwise} \end{cases} \tag{10}$$

where $rand_{i,j}$ is a random number lies in range [0,1] and $k \in \{1, 2, \cdots, D\}$ is a randomly selected dimension index to ensure at least one dimensional component of trial solution Z_i^t is inherited from the donor vector $Y_{i,j}^t$.

For exponential crossover, an integer $n \in \{1, 2, \cdots, D\}$ is randomly chosen as the starting point of the dimension index for a target vector to perform crossover with the mutant or donor vector. Another integer $L \in \{1, 2, \cdots, D\}$ denotes the number of dimensional components to be inherited from the donor or mutant vector to form the trial solution. Referring to the values of n and L, the trial solution (Z_i^t) can be obtained from (11) as follows:

$$Z_i^t = \begin{cases} Y_{i,j}^t & \text{if} j = \langle n \rangle_D, \langle n+1 \rangle_D, \cdots, \langle n+L-1 \rangle_D \\ X_{i,j}^t & \text{Otherwise} \end{cases} \tag{11}$$

where $\langle \cdot \rangle_D$ indicates a modulus function of D. Exponential crossover reportedly performs better on certain types of optimisation problems such as those with the presence of linkages between neighbouring decision variables.

3.4. Selection

The selection process enables DE to determine the survival of a target (parent) or a trial (offspring) solution in the next iteration (X_i^{t+1}) of the search process while retaining the population size of DE in every generation. Once the new population is formed in the next generation, the iterative processes of mutation, crossover and selection are performed continuously until the termination criteria are satisfied. Two types of selection exist, namely, local and global [40]. The selection process of DE is mathematically described as follows:

$$X_i^{t+1} = \begin{cases} Z_i^t & \text{if } f(Z_i^t) \leq f(X_i^t) \\ X_i^t & \text{Otherwise} \end{cases} \tag{12}$$

where $f(\cdot)$ is an operator used to determine the objective function or fitness value of an individual solution. If the latest trial vector of Z_i^t produces a better objective function value, then the current target vector X_i^t will be replaced by Z_i^t in the next iteration. The selection process of DE can be implemented through synchronous and asynchronous modes. The DE population can be updated simultaneously during the synchronous mode, whereas the asynchronous mode can be used to update the DE population individually.

4. Enhanced schemes of DE

4.1. Enhanced initialisation scheme

Population initialisation is a crucial task because it can govern the quality of the final solution and the convergence speed of the algorithm. The selection of this non-repeatable process is vital because the quality of the initial population obtained tends to affect the overall flow of the DE algorithm.

An initialisation approach used to generate the initial population of the candidate solution with opposition-based learning was proposed by Rahnamayan et al. [41]. Three essential stages were considered in this initialisation scheme. Firstly, a randomly distributed population denoted as $P(n)$ was obtained from the uniform distribution generator, where n refers to population size. An opposite population of $OP(n)$ was then calculated based on opposition-based learning strategy. Lastly, the initial population was determined by selecting the best n individual solutions evaluated from a combined population set of $\{P(n) \cup OP(n)\}$. Fig. 6 shows the DE with (a) random population initialisation (DEr) and (b) opposition-based population initialisation (DEo). A set of unimodal and multimodal benchmark functions were used for performance comparison, and reports indicate that the DE incorporated with opposition-based initialised scheme can achieve performance gains of 10% in terms of convergence rate. However, the proposed method was unable to solve the benchmark functions with high dimensional sizes in a robust manner.

Ozer et al. [42] proposed a chaotically initialised DE (CIDE) algorithm with faster convergence rate and better robustness against premature convergence. A total of seven chaotic maps — logistic, circle, Gauss, Henon, sinus, sinusoidal and tents — were selected to initialise the DE population. Simulation results showed that the DE algorithms incorporated with chaotic maps for initialisation process tend to demonstrate better optimisation performance than that initialised using classical random sequences. Furthermore, the DE algorithms initialised with both circle and sinus maps were reported to outperform those of remaining chaotic maps in terms of final solution quality due to their more promising capability to flee from the local solutions. The idea of parallel computing was suggested in this study to reduce the computational overhead brought by chaotic initialisation schemes.

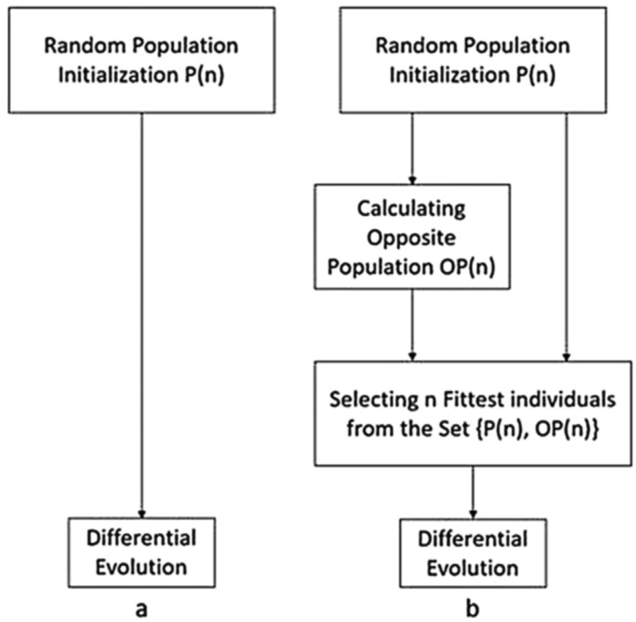

Fig. 6 DE with (a) random population initialisation and (b) opposition-based population initialisation.

Melo et al. [43] proposed smart sampling DE (SSDE), which is able to locate the promising solution area in the search space by leveraging the strength of machine learning. The proposed initialisation process was performed in different regions of the search space, and it aimed to assist DE in identifying the solution regions with higher chances to locate the global optimum. SSDE started with a randomly initialised population in the boundary regions of the search space. The solutions with promising fitness values were chosen iteratively based on a specific threshold until the stopping criteria determined based on the minimum window size or resampling number were satisfied. The K-nearest neighbour (kNN) was utilised to classify the newly generated solutions based on their fitness, and inferior solutions were discarded. Finally, the regions containing promising solutions were partitioned by a rule-based learner. SDE was compared with three DE variants with different modified initialisation schemes, namely, opposition-based DE (ODE), quasi-oppositional DE (QODE) and uniform-quasi-opposition DE (UQODE). Extensive simulation results revealed the advantages of the proposed initialisation scheme incorporated into SSDE in locating the global optima of different optimisation problems in terms of efficiency with a higher success rate (i.e. 96%) as compared with those of the original DE (i.e. 80%).

Zhu et al. [44] designed an adaptive population tuning scheme (APTS) to dynamically control the population size of DE based on the desired population distribution and searching status. The proposed dynamic population strategy enabled excessive individuals to be identified on the basis of their ranking orders and then discarded from population clusters. Particularly, the redundancy monitor variable increased if fitness enhancement for successive generations was observed. The stagnation monitor variable increased if no fitness improvement was detected from the individual solution, and the termination process can be triggered at the predefined threshold value. The population cut strategy was applied to prevent the overgrowth of the population size before it reached the upper boundary limit, whereas the population increment technique was employed when the population size was close to the lower boundary limit to ensure that adequate individuals were allocated for the search process. APTS was integrated with multiple DE variants and their performances were evaluated against that of five other state-of-the-art DE variants by using CEC 2005 test functions. Simulation results showed that the DE incorporated with APTS outperformed other DE variants in solving 100 dimensional problems without incurring excessive complexity and runtime.

Poikolainen et al. [45] presented a cluster-based population initialisation (CBPI) sampling technique for DE to predict the most promising regions of the decision space. A set of initial points were randomly sampled within the decision space and fine-tuned with two local search strategies. These refined solutions were subsequently grouped into different sets of promising solution areas by using K-means clustering algorithm. In the final stage, all clusters' best individuals were saved as the initial population of DE, and other population members can be sampled based on the fitness-based probabilistic criterion to occupy the population size. CBPI was used as the initialisation scheme of six DE variants to solve different benchmark and real-world problems. These simulation results showed that CBPI is able to improve the performance of DE.

An initialisation scheme based on Cauchy mutation and clustering method was proposed by Bajer et al. [46] to generate an initial DE population that has good solution qualities without requiring the domain knowledge of given optimisation problems. Clustering method was firstly employed to identify different areas of solution search spaces with relatively promising qualities. Cauchy mutation was then performed around these cluster centres to generate other good solutions, whereas uniform distribution was also applied to generate other population members to maintain the diversity level of solutions. Performance analyses revealed that the DE variants integrated with this proposed initialisation scheme has a higher convergence speed than those with the conventional initialisation scheme, but the former approach tends to incur additional computational complexity due to the presence of the clustering process. Furthermore, a challenging task is to determine the optimal ratio between solution members generated using the Cauchy mutation and uniform random distribution to achieve a proper trade-off between the exploration and exploitation searches of the algorithm in solving different types of optimisation problems.

An adaptive multi-population DE with dynamic population reduction scheme known as tribe DE-dynamic reduction (sTDE-dR) was proposed to address premature convergence and improve the solution qualities in solving real-world applications [47]. The population of sTDE-dR was partitioned into several tribes, and the ensemble of different mutation and crossover schemes was utilised to evolve these tribe members. A competitive success-based scheme was designed to determine the life cycle and participation ratio of each tribe in the next iteration on the basis of their respective mean success value. A dynamic reduction scheme was also incorporated to reduce its population size of sTDE-dR during the search process. The CEC 2014 benchmark functions with different characteristics were used to compare the performance of sTDE-dR with that of other classical DE algorithms, and the former algorithm performs more competitively in terms of robustness and convergence speed.

Mustafi et al. [48] proposed a hybrid algorithm using GA and DE to prevent the premature convergence issue of the K-mean algorithm by enhancing the quality of its seed initialisation. Specifically, the GA framework was used to determine the original seed points, while the heuristic of DE was leveraged to search for the prerequisite number of clusters to improve the choice of centroids used by the K-mean clustering algorithm. The performance of the proposed hybrid GA-DE in optimising the K-mean algorithm was evaluated using text clustering application and compared with the K-mean algorithm implemented with standard parameters. Despite demonstrating better performance, the K-mean algorithm optimised with hybrid GA-DE tends to suffer from the drawback of high computational times. On the basis of the aforementioned reviews, Table 2 summarises the research related to enhanced initialisation schemes.

4.2. Enhanced mutation scheme

Islam et al. [49] proposed an adaptive DE algorithm with a new mutation strategy to improve its search performance in tackling complex numerical optimisation problems. A new mutation operator, namely, DE/current-to-gr_best/1, was modified from the conventional mutation scheme of DE/current-to-best/1 and the associated mechanism can be expressed in (13)

$$\overrightarrow{V}_{i,G} = \overrightarrow{X}_{i,G} + F_i(\overrightarrow{X}_{gr\_best,G} - \overrightarrow{X}_{i,G} + \overrightarrow{X}_{r_1^g,G} - \overrightarrow{X}_{r_2^1,G}) \tag{13}$$

where $\overrightarrow{X}_{gr\_best,G}$ is a percentage of randomly chosen vectors from group of population, and $\overrightarrow{X}_{r_1^g,G}$ and $\overrightarrow{X}_{r_2^1,G}$ are two different vectors. The best solution from the population was randomly picked from the latest generation to interrupt the target vector to reduce the probability of premature convergence. The proposed framework was compared with other six DE variants by using CEC 2005 benchmark functions in 30, 50 and 100 dimensions. The results showed that the new mutation method can enhance the search performance of DE tremendously in solving various types of optimisation problems. The proposed mutation method was also incorporated with other DE variants such as JADE and jDE. However, their performance studies were limited to fixed group sizes of candidate pools in selecting the best individual solutions over generations.

Gong et al. [50] presented the ranking-based DE mutation operator, where parents in the mutation operators were randomly chosen based on their current population rankings. The probability of each candidate solution to be selected for the mutation process is directly proportional to its fitness value. Performance analyses were performed by comparing the proposed method with competitive jDE algorithm with various mutation operators, other DE variants and self-adaptive parameters. The results showed that the ranking-based mutation operators were able to improve the performance of classical and advanced DE algorithms in terms of their exploitation ability. However, the promising performance of the proposed framework is limited to small-scale problems only.

Fan et al. [51] proposed a discrete mutation control parameters self-adaptive DE (DMPSADE) algorithm to address the issue of balancing exploration and exploitation search when tackling different optimisation problems. In this proposed mutation scheme, each individual solution was assigned a unique mutation control parameter. The performance of DMPSADE was evaluated with CEC 2005 benchmark functions in different dimensional sizes and compared with the existing DE variants. Extensive simulation results revealed that DMPSABE outperforms its competitors in most optimisation problems, but it is outperformed by the Rank-Code, MDE_pBX and Rank-JADE when dealing with 30 dimensional problems.

Ali et al. [52] proposed a multi-population DE balanced ensemble of mutation strategies (mDE-bES) to tackle large-scale optimisation problems with enhanced population diversity. The individuals were partitioned into multiple subgroups to prevent the stagnation of the main swarm, where each subgroup was evolved using different mutation strategies. A novel mutation scheme was also proposed to produce good-quality solutions by leveraging useful information obtained from the best or any randomly selected individuals from a given population topology. The fitness evaluation of mDE-bES was divided into different epochs, where information exchange between subgroups can be facilitated at the end of every epoch through the swapping of subgroup members. Performance evaluation was conducted by comparing mDE-bES with other optimisers in solving 19 large-scale global optimisation benchmarks in dimensional sizes set in between 50 and 1,000. Despite having promising performance in solving large-scale problems, the parameter settings of mDE-bES were obtained through manual tuning process instead of adaptive ones.

Hamza et al. [53] proposed a new mutation strategy inspired by the constraint consensus (CC) method and incorporated into DE to reduce the constraint violation of new solutions generated during the searching process. In contrast to most existing works that considered the constraint violation issue only during the ranking and selection processes of solutions, the CC method directly participated in the mutation process to reduce the number of new solutions that violated the predefined constraints. Notably, the CC-based mutation scheme was applied only on certain infeasible solutions during the search process to maintain the population diversity with reduced runtime. Simulation studies revealed that the CC-based mutation scheme can improve DE in terms of solution quality with a significantly reduced runtime up to 44.7% and 10.6% for problems with dimensional sizes of 10 and 30, respectively. Nevertheless, the CC-based mutation scheme is ineffective in handling multimodal problems due to the rapid loss of population diversity during the initialisation stage to identify the feasible search space of given optimisation problems.

Ho-Huu et al. [54] proposed an improved DE (IDE) that is able to evolve the target vectors in population by using multi-mutation operators known as 'rand/1′', 'rand/2′', 'best/1′' and 'best/2′'. A selection scheme was devised in the mutation stage to assign a unique mutation operator to each individual solution to maintain the balance between the global and local search abilities of the algorithm. Furthermore, an elitist selection scheme was proposed to replace the conventional selection scheme during the selection process of IDE, aiming to accelerate the convergence rate of the algorithm. The performance of IDE was evaluated using five benchmark functions and the optimisation of size and shape for truss structures subjected

Table 2 Summary of research on enhanced DE initialisation scheme.

Author	Technique introduced	Dimensional sizes	Results	Performance metrics	Merits	Limitations
[41]	Opposition-based initialization method	2, 3, 4, 5, 6, 10, 20, 30 and 100	The convergence rate of the DE algorithm with opposition-based initialization was enhanced by 10% as compared with the DE algorithm with random population initialization.	Acceleration rate	The proposed initialization method improved diversity level of initial population, hence increasing convergence rate of algorithm.	Significant performance degradation of acceleration rate can be observed in higher dimensional problems, i.e., for $D > 10$
[42]	Chaotically Initialized DE (CIDE)	2, 5 and 10	The presence of complex and yet dynamic initialization methods can improve the quality of solutions in solving optimization problems.	Not stated	Reduced probability of premature convergence due to improvement of global search capability.	Only two out of seven chaotic maps can increase the solution quality in solving benchmark functions. The capability of other chaotic maps in generating solution with better quality to solve real-world applications were not investigated.
[43]	Smart Sampling DE (SSDE)	10, 20, 30, 40 and 60	Success rate and success performance of SSDE were proven to outperform original DE by 16% and 76%, respectively.	NFC, Success rate (SR), Success performance (SP)	SSDE has better efficacy to find initial populations of superior quality when compared to three other DE variants with oppositional learning.	High computational cost was incurred due to the utilization of machine learning techniques to perform the smart sampling approach.
[44]	Adaptive population tuning scheme (APTS)	30, 100	The proposed JADE-APTS achieved a good performance in 30 dimensional problems and best performance in 100 dimensional problems.	Mean error and standard deviation	Simplicity in implementation. Population size can be adaptively adjusted to increase the probability of locating global optimum.	Effect of control parameters were not carefully studied. High tendency of status monitor to discard the candidate solutions that have temporary inferior performance but can be potentially useful in long terms.
[45]	Cluster-Based population initialization (CBPI)	10, 30, 50, 100 and 1000	The proposed algorithm is efficient and able to improve the performance of DE framework consistently.	Average fitness and standard deviation	The presence of intra-cluster and inter-cluster mutation strategies can improve explorative behavior of algorithm.	Additional computational cost can be incurred by clustering process. In addition, the results were not explicitly exploited.
[46]	A population initialization method for evolutionary algorithms based on clustering and Cauchy deviates	2, 4, 10, 30 and 50	The proposed initialization method achieved more rapid convergence rate speed than the conventional initialization approach.	Average execution time	Convergence rate of proposed DE variant was increased notably when compared to the DE variants with conventional population initialization method and DE with opposition-based initialization method.	High computational complexity. Difficulty in estimating optimal ratios between the solutions generated by Cauchy mutation and uniform distribution when solving different problem types.
[47]	Success Tribe DE-dynamic Reduction (sTDE-dR)	10, 30, 50 and 100	sTDE-dR solved unimodal, multimodal, composite and hybrid function efficiently.	Mean error, standard deviation, and Friedman test	Increased probability of generating good offspring solutions due to promising ability of each tribe in disseminating the best solution to future tribe.	Inferior performance of algorithm in solving the hybrid functions with higher dimensional sizes.

Table 2 (*continued*)

Author	Technique introduced	Dimensional sizes	Results	Performance metrics	Merits	Limitations
[48]	A hybrid approach using genetic algorithm and the DE heuristic	2, 3, 4, 5	The proposed method outperformed the classical versions of k-means algorithm	Silhouette score and DB-Index score	Initial population with the higher diversity level was generated due to the better coverage of initial solutions in the search space, hence generating the most desirable centroids for k-means clustering algorithm.	The execution time of proposed algorithm were significantly slower than that of the conventional k-means clustering algorithm.

to frequency constraints. While the solution accuracies of IDE are similar to that of the original DE, the former algorithm was reported to have better performance than the latter one in terms of computational cost.

Opara et al. [55] advocated that the existing DE mutation operators can be generalised through parameter tuning because these algorithms are different only in terms of the range and direction of the difference vector performed on a given base vector. A mathematical framework was therefore developed to represent different mutation schemes in terms of expectation vectors and covariance matrices of mutants' distribution. A generalised scaling factor was also introduced to adjust the mutation range, enabling the expectation vectors and covariance metrices of expected mutants' distribution to simulate the statistics of two mutation operators, i.e. DE/best/1 and DE/rand/1. The proposed framework was evaluated using CEC 2013 benchmark functions in different dimensional sizes. Comparable performances were found in different transformations of scaling factors, especially for DE variants with two difference vectors.

Mohamed et al. [56] proposed an enhanced fitness adaptation DE (EFADE) with a novel triangular mutation operator to handle the global optimisation problems with better balancing of exploration and exploitation searches. Three solutions were firstly chosen randomly during the triangular mutation process. The difference vectors between the best, better and worst solutions with each randomly selected solution were then calculated to construct the convex combination vector of triplet. Two parameter adaptation schemes were also introduced to adjust the control parameters of EFADE effectively without requiring any additional parameters or priori knowledge of given problems. The performances of EFADE were compared with those of 12 DE variants by using CEC 2013 benchmark functions, and EFADE can outperform its peers in terms of solution quality and robustness. Although EGADE can perform well in solving problems with low and medium dimensional sizes, the effectiveness of modifications that were made to deal with large-scale optimisation problems remains unexplored.

Yu et al. [57] proposed several mutation operators to handle the feasible and infeasible solutions of DE separately when dealing with constrained multi-objective optimisation problems that consider the goals of objective optimisation and constraint handling equally crucial. For infeasible solutions, randomisation processes based on major and minor mutations were proposed at the early and later stages of optimisation, respectively, to balance the exploration and exploitation searches. For feasible solutions, different modifications were proposed for mutation operators to rank these solutions based on their Pareto domination status and diversity values to balance the convergence and diversity. Nineteen benchmark functions and a real-world problem known as combined economic emission dispatch were used to examine the performance of proposed method with other peer algorithms such as reference vector-guided EA, speed-constrained multi-objective PSO, archive-based hybrid scatter search, reference point-based many-objective and multi-objective EA based on decomposition. Numerical results showed that the proposed method outperformed the others, having better solution diversity and higher convergence rate in solving problems with low dimensional sizes of 2 and 10. However, the robustness of the proposed method in solving problems with higher dimensional sizes remains unexplored.

Ramadas et al. [58] proposed a reconstructed mutation strategy for DE (RDE) to solve multilevel thresholding problems based on fuzzy entropy. Apart from the scaling factor, two additional control parameters with complemented values were incorporated into the mutation operator to enhance convergence speed and prevent the premature convergence of the algorithm. The inclusion of two additional control parameters in RDE also enabled greater emphasis on the value of the donor vector and prevented it from collecting the neighbourhood of global best solutions. Simulation studies showed that RDE can obtain better thresholding results within a shorter processing time when dealing with weather images with different hazard severity levels, thus enabling more accurate analysis of weather conditions.

Wang et al. [59] proposed a self-adaptive mutation DE algorithm based on PSO (DEPSO) to address the slow convergence and high tendency of premature convergence exhibited by the original DE. An improved DE/rand/1 mutation scheme was introduced based on elite archive strategy to promote the global exploration search of DEPSO. Meanwhile, the convergence speed of DEPSO was enhanced by incorporating another PSO-based mutation scheme. The performance of DEPSO was evaluated using a set of benchmark functions with different dimensional sizes (i.e. 30 and 100) and a real-world optimisation problem known as arrival flight scheduling. DEPSO successfully improved its convergence speed in solving simple optimisation problems without sacrificing the population diversity. The robustness of their self-adaptive mutation strategy with a relatively simple structure in dealing with more complicated optimisation problems remains questionable.

Xiao et al. [40] presented a multi-strategy different dimensional mutation DE (MDMDE) to address the slow convergence speed and premature convergence of the conventional algorithm. A different dimensional mutation strategy was firstly proposed to enhance population diversity, where each dimensional component of the mutant vector is contributed by the base vector and difference vector from different dimensions, as described in (14)

$$v_{i,j}^{g+1} = x_{r3,n}^{g} + F \cdot \left(x_{r1,m}^{g} - x_{r2,m}^{g} \right) \tag{14}$$

where $i \neq r_1 \neq r_2 \neq r_3, j \neq n \neq m$, $v_{i,j}^{g+1}$ is the mutation vector, $x_{r3,n}^{g}$ is the base vector and $x_{r1,m}^{g} - x_{r2,m}^{g}$ is a different vector from distinct dimensions. A multi-strategy mutation scheme was also designed to enhance the convergence speed of MDMDE by dividing the overall optimisation process into four generation units. The first- and third-generation units of MDMDE adopted a conventional mutation strategy of 'DE/best/1′ that promoted exploitation search, whereas the new dimensional mutation strategy was utilised in the second- and fourth-generation units under the presence of dynamic mutation factor to prevent the stagnation of population in local optima regions. A new crossover rate scheme varied based on the cosine function was also introduced to further enhance the robustness of MDMDE towards premature convergence. The proposed MDMDE was applied to solve eight simple test functions, and it outperformed its peer algorithms in terms of solution accuracy and convergence speed. Nevertheless, the performance evaluation of the current study might not be sufficient to fully explore the full potential of MDMDE due to the small number of test functions and peer algorithms used. Furthermore, the idea of different dimensional mutation strategy proposed in this study might not be applicable for most real-world optimisation problems because the decision variables encoded in different dimensions tend to have different search ranges.

Prabha et al. [60] presented a new mutation vector, namely, the DE haemostasis operator (DEHeO) inspired by the haemostasis process in the human body to enhance the population diversity of the algorithm to prevent premature convergence. Fitness criterion was firstly employed to divide the main population of DEHeO into revision and remainder pools used to store the best and better candidates, respectively. Both pools performed searching independently based on the newly designed haemostatic mutation operator to introduce diverse new solutions at every iteration. Further population diversity can be supplemented by the haemostatic mutation operator by removing the pressure from all individuals in generating potential solutions. The performance of DEHeO was compared with that of other DE variants by using the Comparing Continuous Optimisers (COCO) platform, and the proposed algorithm was reported to have better solution accuracies and convergence rates. Nevertheless, laborious processes of manual parameter tuning were needed to determine the optimal parameter settings of DEHeO to achieve these performance gains.

Bidgoli et al. [61] proposed a new version of generalised DE (GDE4) with an ordered mutation operator to solve many-objective optimisation problems more effectively. In contrast to conventional DE, which performed mutation on randomly selected individual solutions, GDE4 ranked the candidate solutions by using the concepts of non-dominated sorting and crowding distance before mutation. The best-ranked solution among three randomly selected solutions was considered the parent, whereas the remaining two are identified as the second and third candidate solutions for the mutation process, respectively. With these ordered vectors, the summation and subtraction operators of DE mutation enable the new mutant solution to inherit the quality of the best and better candidate solutions by being attracted towards the first and second vectors and being repelled from the third one. The standard benchmark functions of CEC 2017 were used to evaluate the performance of GDE4, and it was reported to outperform its competitor in most test functions, implying the effectiveness of using ordered vectors for mutation. The potential of other ranking strategies has yet to be explored in this study.

Deng et al. [62] proposed a DE with two dynamic speciation-based mutation strategies (DSM-DE) to solve single-objective optimisation problems more effectively. Dynamic speciation technique was firstly performed in DSM-DE to partition the population dynamically into multiple numbers of species with each species seed considered as centres. The best vector of each species was also considered base vectors of two proposed mutation vectors, i.e. 'DE/seeds-to-seeds' and 'DE/seeds-to-rand', with greater explorative and exploitative strengths, respectively. 'DE/seeds-to-seeds' considered another two species seeds randomly chosen from other species to construct the difference vector, while two random individuals were randomly selected by 'DE/seeds-to-rand' to determine the difference vector as in (15) and (16)

$$\overrightarrow{V}_{i,j,g} = \overrightarrow{X}_{i,seedi,g} + F_{i,j,g}.(\overrightarrow{X}_{i,seedr1,g} - \overrightarrow{X}_{i,seedr2,g}) \tag{15}$$

$$\overrightarrow{V}_{i,j,g} = \overrightarrow{X}_{i,seedi,g} + F_{i,j,g}.(\overrightarrow{X}_{i,r1,g} - \overrightarrow{X}_{i,r2,g}) \tag{16}$$

where $\overrightarrow{X}_{i,seedi,g}$ is the seed of the target vector; $F_{i,j,g}$ is the mutation factor for each generation; *seedr1*, *seedr2* are the indices of two randomly selected species seeds; and *r1* and *r2* are the indices of two randomly chosen solutions from the DE population. The performance of DSMDE was compared with that of its peer algorithms by using CEC 2014, CEC 2015 and Lennard-Jones potential problems. DSMDE has competitive search accuracy and efficiency due to the proper balancing of global and local searches brought by 'DE/seeds-to-seeds' and 'DE/seeds-to-rand'. Despite its promising performance, DSMDE is sensitive to species sizes and it requires a manual parameter tuning process to obtain an optimal species size.

Sun et al. [63] proposed a novel Gaussian mutation and modified common mutation schemes in their proposed GPDE to produce new mutant vectors collaboratively and adaptively by referring to their respective cumulative scores. A periodic function was adopted to generate the scaling factor to achieve proper balancing of exploration and exploration strengths. The population diversity of GPDE was further enhanced through the fluctuant crossover rate obtained from Gaussian function to ensure its robust performance in dealing with complex problems. Thirty benchmark functions from CEC 2014 and four real-world problems were employed to compare the performance of GPDE with that of seven DE variants. Despite its promising search performance, GPDE is more sensitive to the changes of crossover rate due to its greater impacts on population sizes, thus requiring a more careful parameter tuning process to achieve optimal settings.

Attia et al. [64] proposed an enhanced DE variant abbreviated as MSaDE with self-adapting control parameters and multi-mutation strategies, aiming to improve its solution accuracy and convergence speed. The proposed MSaDE consists of three mutation operators with unique control parameters. 'DE/rand/1′ and 'DE/best/1′ can promote the global search and local search abilities of MSaDE, respectively. The third mutation technique was designed to offer better balancing of exploration and exploitation searches by generating new mutant vectors from the average values of those obtained from the two aforementioned mutation operators. Notably, only one mutation strategy was selected to form the trial vector in each generation based on a selection probability influenced by the locations of the worst, current and best individuals in each generation. Selected benchmark functions from CEC 2013 and CEC 2005 were used for performance comparison purposes. Despite performance gains achieved by MSaDE, some performance degradation can be observed when dealing with multimodal and composition functions, implying the limitations of their modifications in handling the optimisation problems with a more complicated fitness landscape. On the basis of the aforementioned reviews, Table 3 summarises the research related to enhanced mutation schemes.

4.3. Enhanced crossover scheme

Guo et al. [65] presented a rotationally invariant operator known as eigenvector-based crossover operator to address optimisation problems with rotated fitness landscapes more effectively. A rotated coordinate system was firstly constructed by referring to the eigenvector information of the covariance matrix of the population. The offspring solutions can then be generated by the parents that are randomly selected from standard or rotated coordinate systems to prevent the rapid diversity loss of the population. Eight DE variants incorporated with the eigenvector-based crossover operator were evaluated using the selected benchmark functions from CEC 2011, CEC 2012 and CEC 2013. Significant performance improvement of these DE variants was observed, especially when dealing with non-separable unimodal functions. Nevertheless, the effects of parameter settings such as population size and dimensionality on the performance of the proposed method remain unexplored.

Cai et al. [66] proposed a hybrid linkage crossover (HLX) scheme to leverage the problem-specific linkage information between pairs of variables for more effective guidance of the search process. An improved differential grouping technique was firstly incorporated into HLX to adaptively extract different groups of tightly interactive variables known as building blocks (BBs). Two group-wise crossover operators, abbreviated as GbinX and GorthX, were then proposed to guide the crossover process without disrupting the tight linkage structures of BBs. Accordingly, the proposed HLX scheme can be easily adapted into the existing DE variants to achieve more promising optimisation performance. Two sets of benchmark functions adopted from CEC 2005 and CEC 2012 were employed to investigate the effectiveness of HLX in improving the optimisation performance, parameter sensitivity and scalability of all involved DE variants.

Hui et al. [67] proposed an ensemble and arithmetic recombination-based speciation DE (EARSDE) to solve the multimodal optimisation problems. In contrast to the conventional approach, the arithmetic recombination-based neighbourhood speciation technique incorporated into EARSDE can enhance exploration without having to suffer from any radius parameterisation issues. The proposed EARSDE was reported to outperform 11 peer algorithms in terms of efficiency and robustness when evaluated using 29 multimodal benchmark functions. The proposed speciation technique adopted by EARSDE was proven to be more generalisable in practical situations for being able to identify the peaks and troughs of highly irregular fitness landscape regions.

Xu et al. [68] proposed a superior–inferior (SI) crossover strategy and a superior–superior (SS) crossover strategy to improve the diversity of the DE population. The SI scheme is triggered to enhance the exploration strength of an algorithm if the population diversity is too low. At the same time, the exploitation search is promoted through the SS scheme if the population is diversified. Both the SS and SI schemes are adaptable into typical binomial and exponential crossover operators and can hence be incorporated into various DE frameworks. Simulation studies using CEC 2005 benchmark functions showed that the search performance of DE variants is highly dependent on the parameter settings of both SI and SS crossover schemes.

Fan et al. [69] presented a crossover adaptation strategy in self-adaptive differential evolution (CSA-SADE) to improve the performance of DE. Each CSA-SADE individual was envisioned to have a unique crossover strategy, mutation strategy and control parameters that can be changed adaptively by referring to its latest search progress. The proposed CAS-SADE was compared with eight advanced EAs, and it was proven competitive in solving the CEC 2005 benchmark functions and kinetic parameter estimation problem of mercury oxidation due to its enhanced exploitation capability.

Fister et al. [70] proposed an epistatic arithmetic crossover operator into an ensemble DE variant known as eXEDE. Unlike the ordinary arithmetic crossover, the epistatic arithmetic operators considered the impact of epistatic genes in the context of evolutionary computation by expressing the epistatic as a graph product of two linear graphs represented by the candidate solutions that are involved in the crossover process. The performances of eXEDE in solving the CEC 2014 benchmark functions are compared with those of three DE variants, and the presence of epistatic arithmetic crossover operator is proven promising in enhancing the search performance of DE significantly.

Qiu et al. [71] presented a multiple exponential crossover operator that enables the formation of a new solution through the combination of multiple segments of target and mutant vectors. Theoretical and empirical analyses showed that this semi-consecutive crossover operator is not only equipped with the strengths of both conventional binomial and exponential crossover operators but also demonstrates better capability in handling the subset of tightly interactive variables. The proposed exponential crossover was implemented in six classical and one DE variants, and the performance evaluations reported its outstanding search performance in solving problems with unknown variable interrelations as compared with the conventional binomial crossover approach.

Table 3 Summary of research on enhanced DE mutation scheme.

Author	Technique introduced	Dimensional sizes	Results	Performance metrics	Merits	Limitations
[49]	An adaptive DE algorithm with novel mutation and crossover strategies	30, 50 and 100	The proposed method is able to solve various large-scale optimization problems with improved search performance.	Mean fitness and standard deviation	More explorative mutation strategy was proposed to preserve population diversity. Biased parent selection strategy was incorporated into crossover operation to promote more exploitative behavior. These strategies reduced the probability of algorithm to suffer with the premature convergence in dealing with large-scale optimization problem.	The performance evaluations only focused on the constant group size over generations.
[50]	DE with ranking based mutation operators	30, 50, 100 and 200	The performance of classical and advanced variants of DE algorithms was improved with the adoption of ranking-based mutation operators.	Error value	Enhanced exploitative behavior through ranking-based mutation operators. Simplicity in implementation without increasing the complexity of algorithm significantly.	The proposed ranking-based mutation operators might not effective in improving the performance of DE variants in solving certain types of problems due to the excessive level of exploitation search introduced.
[51]	Self-adaptive DE with discrete mutation control parameters (DMPSADE)	30, 50 and 100	The average optimization performance of the proposed DMPSADE algorithm was better than other DE variants in solving problems with 50 and 100-dimensional sizes.	Mean fitness, standard deviation, Wilcoxon signed rank test and Friedman test	The proposed method allowed each optimized variable to have different mutation control parameters and adaptive adjustment of mutation strategy via competition, leading to better preservation of population diversity.	The proposed algorithm did not show significant performance gains over other DE variants in solving benchmark problems with 30-dimensional size. Encoding strategy of the proposed method increased the complexity of algorithm, leading to longer execution time.
[52]	Multi-population DE with balanced ensemble of mutation strategies (mDE-bES)	50, 100, 200, 500 and 1000	The proposed method has an excellent performance in solving persistent global problems efficiently	Mean error, standard deviation, computation time, Wilcoxon rank sum test, Friedman aligned rank test	Multi-population strategy promoted exploitative strategy, whereas information sharing scheme among different subpopulations was adopted to reduce the drastic loss of population diversity over the generations. Different mutation strategies and control parameters were also assigned to each subpopulation to	The performance of proposed algorithm was not thoroughly verified with the adaptive adjustment of control parameters.

Table 3 (*continued*)

Author	Technique introduced	Dimensional sizes	Results	Performance metrics	Merits	Limitations
					maximize the coverage of all individual solutions in search space.	
[53]	Constraint consensus mutation DE	10 and 30	The proposed algorithm was reported to have promising performance in solving constrained optimization problems in terms of fitness value and computational time. The computational time of proposed algorithm in solving 10 and 30 dimensional problems were 33.7% and 10.6% faster than those of conventional DE, respectively.	Fitness value and computational time.	Faster computational time as compared to conventional DE in solving the constrained optimization problems.	High tendency of the proposed algorithm to suffer with rapid loss of population diversity during the initialization stage, leading to the inferior performance when dealing with multi-modal constrained problems.
[54]	Adaptive DE mutation scheme	Not stated	The proposed IDE algorithm was reported to have promising performance in dealing with truss structure application problems with the presence of multiple complex nonlinear constraints.	Weight values, diversity index	Better preservation of population diversity through multi-mutation operator selection scheme.	Increased execution time due to the presence of elitist selection scheme.
[55]	A probabilistic perspective comparison of mutation strategies in DE	10, 30 and 50	The accuracy of Gaussian approximation were improved with the increasing numbers of difference.	Mean error and standard deviation.	Offered a new perspective, i.e., Gaussian approximation, in comparing DE with and other optimization algorithms	The transformations of scaling factors did not influence the performance much, especially for the case of multiple difference vectors.
[56]	Enhanced fitness-adaptive DE algorithm (EFADE)	10, 30 and 50	The proposed method EFADE outperformed other DE variants in terms of quality, stability, and robustness of solutions.	Wilcoxon signed rank test and Friedman test	The triangular mutation scheme can improve the exploration and exploitation searches simultaneously, without sacrificing its convergence speed.	Only small- and medium-scale problems were considered in the performance evaluations. The capability of the proposed algorithm in solving large-scale problem remain unexplored.
[57]	DE mutation operators for constrained multi-objective optimization	2 and 10	The proposed algorithm was reported to have promising performance in solving CMOPs and real-world problem.	Mean inverted generational distance (IGD), mean hyper-volume, standard deviation, and Friedman test	Gauss mutation was implemented to tackle infeasible solutions more effectively, while DE mutation operation was performed to handle feasible solutions more robustly.	The performance evaluation of the proposed algorithm only considered the small-scale optimization problem. The capability in dealing with large-scale problems were not investigated.
[58]	Segmentation of weather radar DE	25 and 50	The proposed RDE technique has the fastest processing time	Mean fitness, processing time and Friedman test	Faster computation time as compared to the conventional DE	Less thorough performance evaluations of image

(*continued on next page*)

Table 3 (*continued*)

Author	Technique introduced	Dimensional sizes	Results	Performance metrics	Merits	Limitations
	(RDE)		as compared to conventional DE algorithm in solving the image thresholding task.		algorithm in solving image thresholding tasks	thresholding because only two sets of images were used for performance comparison with conventional DE.
[59]	Self-adaptive mutation DE-particle swarm optimization (DEPSO)	30 and 100	The overall performance of DEPSO surpassed the conventional DE and PSO algorithms.	Mean fitness, standard deviation, success rate, mean number of evolution generations, Wilcoxon rank sum test, Friedman test and Kruskal-Wallis test.	The improved DE mutation strategy and mutation strategy inspired by PSO can improve the convergence rate of proposed algorithm effectively.	The capability of the proposed algorithm in solving more complex optimization problems was not thoroughly investigated.
[40]	Multi-strategy differential dimensional mutation DE	2 and 50	The proposed algorithm outperformed the compared DE variants in terms of fitness value and convergence rate.	Minimum, mean, median fitness values and standard deviation	Notable performance gain in terms of the convergence rate was achieved by the proposed algorithm against well-established DE variants.	The performance evaluates were conducted using simple single-objective test functions and compared with only two DE variants. Insufficient test functions, real-world problems and statistical analysis were used to further analyze the performance of the proposed algorithm.
[60]	DE with biological-based mutation operator (DEHeO)	2, 3, 5, 10, 20 and 40	The proposed DEHeO surpassed the other DE variants in terms of convergence rate.	Comparing Continuous Optimizer (COCO) framework	Global search ability was enhanced by the proposed mutation strategy. The pool assignment scheme can achieve the better balancing of exploitation and exploration search behaviors without discarding the individual solutions with relatively worse fitness value.	The parameters of proposed mutation scheme were fixed and cannot be adjusted adaptively.
[61]	The Generalized DE with Ordered Mutation (GDE4)	5, 10 and 15	The proposed GDE4 algorithm was reported to outperform its previous version of GDE3 algorithm in solving MaF test suite from CEC 2017.	Mean, median, worst and best inverse generational distance	Information such as non-dominance level and crowding distance of three randomly selected individual from search space were utilized by the proposed mutation scheme to improve the balancing of exploration and exploitation search behaviors of proposed algorithm.	The effectiveness of proposed algorithms is only tested with smaller dimensional problems and its robustness in problem with larger dimensional sizes remained unexplored.
[62]	Dynamic speciation-based mutation DE (DSM-DE)	30, 50 and 100	DSM-DE outperformed the compared DE variants in terms of search accuracy and	Mean error, standard deviation, Wilcoxon signed rank test and Friedman test.	The proposed mutation strategy, i.e., *DE/seeds-to-seeds* and *DE/seeds-to-rand*, can enhance the	The methodology used to determine the predefined species size of proposed algorithm was not clearly stated.

Table 3 (*continued*)

Author	Technique introduced	Dimensional sizes	Results	Performance metrics	Merits	Limitations
			convergence rate.		exploitation and explorative search behaviors of algorithm, respectively, leading to the improvement of convergence rate of algorithm.	
[63]	Gaussian mutation and dynamic parameter adjustment DE (GPDE)	30, 50 and 100	The proposed algorithm showed its competitive performance in solving limited numbers of benchmark problems.	Mean fitness, standard deviation and Wilcoxon rank sum test.	The proposed periodic scaling factor had shown promising potential to improve the exploitation and exploration abilities of proposed algorithm.	No notable performance gain in term of convergence rate was achieved by the proposed algorithm as compared to its peer algorithms.
[64]	Enhanced DE Algorithm with Multi-Mutation Strategies and Self-Adapting Control Parameters	30 and 50	The proposed algorithm was reported to outperform all the compared DE variants in solving benchmark functions. Furthermore, its convergence rate was reported to be better than its competitors.	Mean error, standard deviation, Wilcoxon signed rank test and Friedman test.	The proposed mutation strategy adopted three search directions that were assigned with different weightages of explorative and exploitative search behaviors in generating new solutions for achieving better preservation of population diversity.	Notable performance degradation can be observed from proposed algorithm when dealing with the multimodal and composition functions. The performance of proposed algorithm in solving real-world problems remain unexplored.

Deng et al. [72] proposed a new DE variant that consists of rotating crossover operator (RCO) with multi-angle searching strategy, aiming to reduce the likelihood of generating inferior offspring solutions by expending the search space tactically. Unlike conventional binomial crossover scheme, the trial vector of RCO can be generated diversely within the circle regions around the donor and target vectors by referring to the self-adaptive crossover parameter and rotation control vectors that followed the Levy distribution. A comparison analysis was conducted between JADE-RCO with other five enhanced DE variants on a group of test functions in CEC 2013. Simulation results showed that DE-RCO outperformed the compared DE variants in terms of search accuracy and convergence rate, with performance gains in the range of 57% to 96%. DE-RCO implies the feasibility of developing different variants of multi-angle search strategy with an efficient parameter selection scheme to enhance the search performance of other DE variants.

Ghosh et al. [73] proposed a switched parameter DE with success-based mutation and modified BLX crossover (SWDE_Success_mBLX) to solve scalable optimisation problems without sacrificing the simplicity of its algorithmic framework. A simple control parameter selection strategy that enables the random and uniform switching of mutational scale factor and crossover rate within their feasible ranges was firstly proposed. A success-based switching strategy was incorporated to determine the mutation schemes of each solution on the basis of its search performance history. The crossover of each target-donor pairs was performed using a binomial crossover or a modified BLX (mBLX) crossover via a probability-based selection scheme. The latter mBLX crossover scheme facilitated the search in the region between and beyond the dimensional bounds established by the target-donor pairs to balance the exploration and exploitation searches. Two sets of benchmark functions, namely, CEC 2010 and CEC 2013, were used to evaluate the overall performance of SWDE_Success_mBLX in solving moderate- and large-scale optimisation problems, ranging from 30 to 1,000 dimensions. The results showed that the proposed SWDE method outperformed other advanced DE variants in terms of robustness and searching capability efficiency. The robustness of SWDE_Success_mBLX in solving large-scale optimisation problem could be improved by adjusting the population size intelligently through dynamic population reduction or multi-population schemes.

Mohamed et al. [74] proposed a DE variant known as adaptive novel DE (ANDE) to tackle large-scale continuous optimisation problems. A novel adaptation scheme was introduced in ANDE to adjust the crossover rate of each solution based on its past searching experience to achieve the proper trade-off between enhanced convergence speed and good preservation of population diversity. Performance evaluation using 20 large-scale global optimisation benchmark functions showed that the proposed ANDE outperformed seven compared algorithms in terms of search accuracy, robustness and efficiency.

Zhou et al. [75] proposed a modified JADE with sorting crossover rate (JADE_sort) with enhanced search ability. A sorting CR mechanism was firstly incorporated into JADE_sort by assigning the fitter individual solutions with lower CR values to ensure the propagation of these promising solutions into the next generation. To further improve the exploration ability of JADE_sort, a better scheme retention mechanism was introduced during the crossover process to preserve the components of the mutant vector originated from better offspring, aiming to facilitate a deeper search around promising individual solutions. The proposed JADE_sort was compared with nine DE variants in solving the CEC 2005 benchmark problems. Although the JADE_sort was proven to have better global and local search abilities compared with its peers, the simulation studies revealed its weakness in handling hybrid composition functions due to the rapid loss of population diversity.

Alswaitti et al. [76] proposed a variance-based DE algorithm with an optional crossover (VDEO) to solve data clustering optimisation problems with improved convergence speed and solution quality. A single-solution representation approach was firstly adopted by VDEO to overcome the limitation of population-based solution techniques in initialising and formulating the clustering optimisation problems. A switchable scheme that consists of two mutation schemes and a vector-based estimation of the mutation factor were then incorporated into VDEO to achieve a proper balancing of exploration and exploitation searches. Considering that a mutant solution might have better fitness than its trial counterpart, the proposed scheme accelerated the convergence speed of the algorithm by determining the fitness between the mutant and trial solutions before proceeding to the selection process. The clustering performance of the proposed VDEO was compared with that of four DE variants in solving 15 benchmark datasets extracted from the UCI repository and validated using non-parametric statistical analysis. The results showed that VDEO achieved 11.98% average improvement in classification accuracy and cluster density against the competing peers. Despite the promising clustering performance demonstrated by VDEO, the control parameters of the algorithm, such as the mutation factor and crossover rate, are not optimised. Furthermore, the correlation between the input data characteristics and some control parameters, such as probability threshold and degree of randomisation, remains unexplored. On the basis of the aforementioned reviews, Table 4 summarises the research related to enhanced crossover schemes.

4.4. Enhanced selection scheme

Sallam et al. [77] proposed a landscape-based adaptive operator selection DE (LSAOS-DE) to solve the wide ranges of benchmark functions more efficiently and effectively. An adaptive operator selection scheme was introduced, where both the fitness landscape information and the performance track records of each mutation operator in generating fitter offspring were considered in selecting the most appropriate mutation operator to evolve the entire LSAOS-DE population during the optimisation process. The performances of LSAOS-DE in solving the CEC 2014 and CEC 2015 benchmark functions were compared with those of two state-of-the-art multi-method algorithms, four multi-operator algorithms and three single-operator algorithms. The results showed that LSAOS-DE outperformed other advanced DE algorithms in almost all cases. Thus, more than one landscape measure could be used to select the best mutation operator for each solution.

Tian et al. [78] presented a DE with improved individual-based parameter setting and selection strategy (IDEI). A combined mutation strategy that consists of two mixed mutation strategies is firstly incorporated into IDEI to guide the search process of each individual by referring to their respective individual-based parameter settings. A diversity-based selection strategy was also incorporated into IDEI as a weighted fitness function by referring to the fitness value and position of the target and trial individual. The proposed diversity selection strategy not only aims to enhance the population diversity, but it also leverages the large amount of exploration information found to locate better solutions. The performance of IDEI was compared with that of eight peer algorithms in solving the CEC 2005 and CEC 2014 benchmark functions, as well as a real-world optimisation problem. While it demonstrated competitive search performance in the majority of complex optimisation problems, the proposed IDEI has a relatively slow convergence speed in solving some unimodal functions.

Guo et al. [79] observed that the conventional one-to-one selection scheme tends to deteriorate the convergence speed of DE by unfairly rejecting the trial vectors with better fitness than most other current population members, especially if the corresponding target vector was even better. A novel subset-to-subset (STS) selection operator was proposed to enhance the convergence speed of DE by randomly partitioning the target and trial populations into several subsets of populations. For each subset, the best individual solutions among the subset of target and trial populations are identified by referring to their fitness values. Under the presence of the STS selection operator, the trial vectors with better fitness values were expected to have higher chances to survive in the next generation. Extensive simulation studies were conducted to compare the STS selection operator with four other survival selection schemes, and the proposed approach emerged as a more reliable selection scheme. Furthermore, the proposed STS selection improved the search accuracy and convergence speed of all DE variants significantly when it was incorporated into these algorithms.

Qu et al. [80] presented a modified multi-objective DE (MODE) to simultaneously minimise the pollution emission and fuel cost of the dynamic economic emission dispatch (DEED) problem incorporated with wind power plant. An ensemble of selection methods that incorporates non-dominated sorting and summation-based sorting was designed, enabling MODE to perform effectively in different stages of the searching process and in different types of optimisation problems. A heuristic constraint handling technique was further developed to locate all solution members in a feasible search space. Simulation studies revealed that the proposed MODE delivered good performance in solving both standard benchmark functions and DEED problems with and without wind factor.

Rakshit [81] proposed a DE integrated with noise handling policies (NDE) to enhance its optimisation robustness in dealing with solution search spaces consisting of stochastic noise. A stochastic learning automata (SLA) was firstly incorporated

Table 4 Summary of research on enhanced DE crossover scheme.

Author	Technique introduced	Dimensional Sizes	Results	Performance metrics	Merits	Limitations
[65]	Enhancing DE Utilising Eigenvector-Based Crossover Operator	30 and 50	Significant performance improvement was observed from proposed algorithm in dealing with the unimodal functions.	Mean fitness, standard deviation, Wilcoxon rank sum test	The proposed crossover operations allowed the offspring to be properly distributed corresponding to the fitness landscape, and to be directed towards the global optimum without affecting the search capabilities.	Lacking of clear explanations between the effect of dimensionality with population size of proposed algorithm.
[66]	Hybrid linkage crossover for DE (HLX-DE)	10, 30, 50, 100 and 200	High performance of HLX for the DE algorithms in terms of convergence speed as compared to four DE variants, original DE algorithm and advanced DE variants.	Mean error, standard deviation, Wilcoxon signed rank test and Friedman test.	A group-wise binomial crossover and a group-wise orthogonal crossover were designed to guide the crossover process of DE more effectively, enabling the better balancing of exploration and exploitation strengths and the enhanced convergence rate of proposed algorithm.	Slightly performance degradations were observed when the hybrid linkage crossover mechanism was used by different DE variants to solve the hybrid composite functions.
[67]	Ensemble and arithmetic recombination based speciation DE (EARSDE)	1, 2, 3 and 10	EARSDE outperformed the compared optimisation algorithms, in term of efficiency and robustness, in solving multimodal functions.	Success rate, peak ratio, average number of peak and Wilcoxon test	Speciation was performed with the arithmetic recombination and ensemble strategy to improve the exploitative and explorative search behaviors of algorithm, respectively.	The performance of proposed algorithm to solve real-world optimisation problems are unknown.
[68]	DE based superior-inferior (SI) and superior-superior (SS) crossover strategy	30, 50 and 100	The adoption of self-adaptive SI mechanism in DE variants can improve their optimisation performances in solving the unimodal, basic and expanded multimodal functions at 30-dimensional size.	Mean error and standard deviation	Enhanced exploration and exploitation strengths of proposed algorithm by the SI crossover and SS crossover operators, respectively.	Performance degradation of the proposed algorithm can be observed when solving the hybrid composition functions. The ability of SI method to enhance performance of DE variants in solving large-scale and complex problems are questionable.
[69]	Crossover strategies adaptation with self-adaptive DE (CSA-SADE)	30 and 50	The proposed algorithm was reported to outperform five well-established DE variants and three non-DE algorithms, in terms of search accuracy.	Mean error, standard deviation, Wilcoxon rank sum test and Friedman test	The proposed self-adaptive mechanism can improve population diversity by allowing each individual to have the unique combination of crossover strategy, mutation strategy and control parameters.	The scalability of proposed method was not thoroughly analysed with different set of test functions at higher dimensions.
[70]	Epistatic crossover ensemble DE (eXEDE)	10, 30 and 50	The proposed method was observed to outperform the compared DE variants in solving the test functions at 30 and 50 dimensions.	Mean error, standard deviation, Wilcoxon signed rank test, Friedman test and Nemenyi test	The proposed epistatic arithmetic crossover was proven able to improve the search accuracy of the DE algorithm in solving single-objective test functions.	The generated offspring was highly dependent on Cartesian graph product.

(*continued on next page*)

Table 4 (*continued*)

Author	Technique introduced	Dimensional Sizes	Results	Performance metrics	Merits	Limitations
[71]	Various exponential crossover	50 and 100	The proposed method had competitive performance in solving unimodal, basic and expanded multimodal functions at 50 and 100 dimensions.	Mean error, standard deviation and Wilcoxon rank sum test	The proposed multiple exponential crossover recombination scheme increased the robustness of the algorithm in handling different types of problems.	Manual and tedious parameter tuning processes were needed to obtain the optimal combinations of control parameter settings.
[72]	DE Rotating crossover operator (DE-RCO)	30 and 60	The implementation of the proposed rotating crossover operator in DE variants can improve their optimisation performances in terms of search accuracy and convergence rate.	Mean error, standard deviation and Wilcoxon rank sum test	RCO operator is simple to be implemented and can be easily incorporated into different DE variants.	Influences of different parameter settings on algorithm's performances were not thoroughly studied.
[73]	Switchable parameter DE (SWDE)	30, 50, 1000 and 2000	The proposed algorithm outperformed the compared DE variants in solving single-objective optimisation problems at 30, 50, 1000 and 2000 dimensions.	Mean error, standard deviation and Wilcoxon rank sum test	The proposed mechanism can enhance the scalability and robustness of DE in dealing with the large-scale optimisation problems.	Significantly high execution time incurred by proposed algorithm in solving large-scale optimisation problems due to its excessive computational complexity.
[74]	DE with adaptive crossover strategies	1000	The proposed method outperformed other seven optimisation algorithms in solving large scale optimisation problems.	Median, mean error, standard deviation and Wilcoxon signed rank test	The proposed self-adaptive crossover scheme enabled each individual to have different crossover rates in generating new solutions based on past experiences of these individuals in search space.	Th execution time of the proposed method in solving large-scale problem was not investigated. The scalability of the proposed method in solving small-scale optimisation problems (10, 30, 50 dimensions) remained unknown.
[75]	Assemble sorting crossover rate (CR)	30 and 50	The proposed sorting crossover rate mechanism was observed to improve the performance of proposed JADE_sort algorithm over the original JADE algorithm in solving single objective optimisation problem.	Mean fitness, Wilcoxon sum rank test	Better parent solutions have higher probability to retain their schemes into the offspring solutions, hence improving the overall quality of next populations.	The quality of the generated scheme was extremely depended on the latest iteration offspring.
[76]	Variance-based DE with optional CR (VDEO)	Not stated	Average performance improvement up to 11.98% in terms of classification accuracy was reported by the proposed algorithm over the compared DE variants in performing data clustering task on 15 datasets from UCI Machine Learning repository.	Best, worst, median, mean fitness, standard deviation and Friedman test	The proposed switchable DE mutation scheme can balance the search behavior of DE algorithm. The proposed multidimensional mutation factor can enhance the quality of offspring solutions. The convergence rate of algorithm was improved by the optional crossover strategy.	The adoption of multiple proposed mechanisms tends to increase the computational time of algorithm in performing data clustering task.

into NDE to identify an appropriate sample size of solutions in largely noise-affected areas to achieve accurate fitness estimation without incurring additional computation complexity. A new fitness estimation strategy was also proposed by considering the weighted average of all fitness samples to reduce the influences of noisy minority fitness samples. An adaptive mutation rate was designed to select the solutions from relatively less noisy regions for the mutation process. Finally, a niching strategy was incorporated to address the deceptive effect of noise signal in the fitness landscape during the selection phase of NDE, hence ensuring proper trade-off between population diversity and quality. Two sets of benchmark functions, i.e. CEC 2013 functions contaminated with noise signals and CEC 2010 noisy benchmark functions, were used for performance evaluation, and NDE was reported to have better robustness and convergence speed against its competitors. Despite its promising performance, more effective strategies of state quantisation and selection of reward functions were needed for NDE to provide more accurate sampling from noisy regions for fitness estimation. On the basis of the aforementioned reviews, Table 5 summarises the research related to enhanced selection schemes.

4.5. Hybridisation of DE algorithm

Hybridisation is another popular approach used to enhance the search performance of DE by leveraging the strengths of search operators obtained from other computational intelligence algorithms. In this section, we focus on the growing trends in the past six years (i.e. 2016–2021) in research that hybridised DE with other computational intelligence algorithms. Some popular computational intelligence algorithms considered to hybridise with DE are artificial neural network (ANN), PSO, fuzzy logic (FL), CS, ABC, WOA, FA, ACO and GA. Forty-three articles were written on hybridisation of the DE algorithm [59,82–123].

4.5.1. DE with ANN

Mason et al. [82] proposed a hybrid algorithm known as multi-objective ANN with DE, which can approximate and solve DEED problems simultaneously to minimise the cost and emission of the system. Kumar et al. [83] proposed an adaptive DE and ANN to improve the quality of resource scaling decisions made by the cloud prediction system through the fine-tuning of system parameters. The fine-tuned cloud prediction system was tested and compared with the conventional back-propagation ANN model by using two servers' benchmark datasets. Dahou et al. [84] presented an ANN-based language sentiment classifier in which the best configuration of the network architecture and parameters were optimised by DE. Some crucial parameters of the language sentiment classifier that needed to be fine-tuned by DE include the convolution filter sizes, neuron numbers, dropout rate and initialisation rate. Li et al. [85] hybridised the DE into a backpropagation ANN to solve the transient electromagnetic inversion problems commonly encountered in geophysical applications. The proposed hybrid algorithm was reported to have more competitive performances in term of stability, accuracy, robustness and speed when compared with two geoelectric models using two typical benchmark sets. Jiang et al. [86] proposed a simpler way to train the feedforward ANN by using the collective intelligence-based DE to optimise its network structure and parameters. With its ability to generate more diverse solution vectors by considering multiple best individuals from the current population via linear combination model, CIDE has better performance in training ANN. Majhi et al. [87] presented an evapotranspiration prediction model by hybridising the DE with a radial basis function ANN, and it was then applied to predict the climate changes of a moist humid area in east–central India. Saporetti et al. [88] developed a hybrid surrogate model of DE and ANN to classify the petrophysical data automatically to improve the procedures of reservoir characterisation in the oil industry. The optimal architecture and parameter settings of ANN (e.g. types of regularisation, activation function and optimiser) were determined by DE to produce a robust classifier.

4.5.2. DE with PSO

Moharam et al. [89] proposed a hybrid algorithm known as aging leader and challengers PSO and DE (ALC-PSODE) to fine-tune the parameter settings of PID controller. The concepts of aging leader and challengers adopted in ALC-PSODE were useful in addressing the premature convergence issues of PSO and DE, hence producing a PID controller with robust performance. Zhang et al. [90] hybridised DE and PSO for path planning of a mobile robot, which was formulated as a constrained multi-objective optimisation problem. The effectiveness of the hybrid DE and PSO in solving the path planning problem was verified by an extensive amount of simulation studies. Song et al. [91] optimised the design of a 3D wind turbine system using PSO and DE, aiming to maximise the output power generated and minimise the cost incurred. Wang et al. [59] proposed self-adaptive mutation DEPSO to maintain the population diversity by leveraging the promising global search ability of DE/rand/1 mutation strategy. Boks et al. [92] presented a modular hybridisation of DE and PSO, where a total of 800 hybrid variants were produced based on 16 different original variation operators and 4 selection operators considered in their study. Dash et al. [93] hybridised DE and PSO to optimise the design and fabrication of sharp edge filers that can produce good sharp edge frequency response during the filtering process. Choi et al. [94] proposed a hybrid algorithm consisting of DE and PSO in solving global optimisation problems. A modified initialisation scheme was introduced in this hybrid algorithm to produce a better-quality initial population that can lead to better optimisation results.

4.5.3. DE with FL

Ebtehaj et al. [95] proposed a hybrid model known as ANFIS-DE to predict the parameter of Froude number that restricted the velocity effort in the non-deposition sediment transport problem. Sahoo et al. [96] designed a PID controller by hybridising FL and DE to solve the load frequency control problem commonly encountered in interconnected power systems. The proper adjustment of two control parameters (i.e. crossover probability and step size) contributed to the performance gain of PID significantly. Dixit et al. [97] proposed a new image segmentation method by hybridising DE and FL to determine the optimised threshold values with reduced computational complexity. Sharma et al. [98] applied a DE and FL to solve energy-efficient clustering problems. Fuzzy clustering was the

Table 5 Summary of research on enhanced DE selection scheme.

Author	Technique introduced	Dimensional sizes	Results	Performance metrics	Merits	Limitations
[77]	Landscaped-based adaptive operator selection DE (LSAOS-DE)	10, 30 and 50	LSAOS-DE outperformed others DE variants in solving majority benchmark functions from CEC 2014 and CEC 2015.	Mean error, Wilcoxon signed-rank test, Friedman test	Fast convergence speed and good search accuracy.	Optimal parameter settings of proposed algorithm were determined manually.
[78]	Improved individual-based parameter setting selection strategy (IDEI)	30	Competitive search performance in majority of complex optimisation problems.	Mean error, standard deviation and Wilcoxon rank sum test	Diversity-based selection strategy was designed as a secondary guidance of searching process by enabling the individuals with temporary inferior fitness values to be selected for survival in the next iteration.	The proposed diversity-based selection strategy is computationally expensive and has high tendency to promote excessive explorative search behavior that can lead to significant reduction of the convergence rate.
[79]	STS selection operator	30	STS emerged as a more reliable selection scheme compared to other 4 competitive selection schemes.	Mean error, standard deviation, Wilcoxon rank sum test and Friedman test	The proposed subset-to-subset selection is proven effective to improve the convergence rate of DE algorithm.	The scalability of the proposed method was not thoroughly investigated with different dimensional sizes.
[80]	Adopted multi-objective DE (MODE)	Not stated	The proposed algorithm generated good performance on the DEED problem	Mean value of R indicator	Provides better power emission value as compared to the other optimisation algorithm.	The computational times incurred by proposed method might be infeasible for real-world application.
[81]	Improved DE for noisy optimisation	10, 20, 30, 40 and 50	NDE outperformed all its contesters in term of search capability in different dimensions and noise cancelation.	Mean function error value, standard deviation, Wilcoxon rank sum test and Friedman test	The proposed method can improve the convergence rate and search accuracy of DE.	Less efficient in solving the optimisation problems with complex fitness landscapes.

idea of new cluster creation to select the best node within each cluster. Jamali et al. [99] proposed a hybrid algorithm with multi-objective DE and FL to solve Pareto optimisation problems. FL inferences were used to dynamically adjust the mutation factor of DE to balance exploration and exploitation searches. Babanezhad et al. [100] demonstrated the feasibility of hybridising DE and FL to predict the nanofluid characteristics and pattern flow of natural heat transfer in Cu-water. Karimi et al. [101] hybridised DE and FL to develop the models used to estimate the efforts required for developing computer software.

4.5.4. DE with CS

Mlakar et al. [102] hybridised CS and DE (CSDE) to select the optimal threshold values required to perform multilevel segmentation on grayscale images. The proposed hybridisation framework was established by including a reset strategy adopted from CS within the DE loop. Lin et al. [103] proposed a new hybrid CSDE to solve the protein–ligand docking problem. This method achieved performance gains of 9% to 15% in terms of success rate when compared with the original CS by using two benchmark functions. Zhang et al. [104] proposed a new hybrid CSDE to solve several constrained engineering problems. Notably, their proposed hybrid algorithm can separate the main population into two subpopulations for better information exchange, and each subpopulation was evolved independently by CS and DE. Chi et al. [105] proposed a new CSDE to solve the logistics distribution centre location problems, and Xi et al. [106] presented hybrid CSDE with enhanced population diversity and local search ability to address several numerical optimisation problems.

4.5.5. DE with ABC

Gao et al. [107] proposed a hybrid algorithm of ABC and DE with enhanced convergence speed by leveraging the search experience of each solution in previous iterations. An initialisation scheme modified with chaotic theory and oppositional-based learning was also incorporated to improve the global convergence characteristic of this hybrid algorithm. Zhou et al. [108] proposed a hybridised DE with ABC (DE-caABC) to improve the search performance in solving the selection process of cloud manufacturing and service composition. Haohao et al. [109] presented a hybrid algorithm of ABC with DE to solve the path planning and obstacle avoidance problems commonly encountered in quadrotor applications. Specifically, ABC was responsible for enhancing the global search ability of the hybrid algorithm, whereas DE was used to introduce additional diversity to prevent the population from falling into local optima. Najari et al. [110] proposed to hybridise ABC and DE to address the modelling and optimisation of carbon dioxide (CO_2). The first part of this study focused on the estimation of kinetic parameters by using the proposed hybrid algorithm, followed by an investigation of the reactor's performance on the basis of the distribution of hydrocarbons. Zorarpaci et al. [111] developed a new feature selection method by hybridising ABC and DE to solve data classification problems in which data privacy was the main concern. The proposed feature selection method was able to reduce the number of queries dispatched count to the database with different levels of privacy without compromising the classification accuracy of the system significantly.

4.5.6. DE with WOA

Xiong et al. [112] proposed a hybrid WOA and DE to solve the parameter estimation problem for solar model application. The performance of the proposed hybrid algorithm in this modelling problem was compared with that of the original WOA and DE algorithm under different environmental conditions such as weather, temperatures and irradiances. Dhabal et al. [113] presented a hybrid WOA and DE for image enhancement by improving the pixel intensity. It is realised using a cost function with global and local information.

4.5.7. DE with FA

Sarbazfard et al. [114] proposed a hybrid algorithm of FA and DE to solve several global optimisation problems in the real world. Ghosh et al. [115] presented a novel hybrid FA and DE (HFA-DE) algorithm to solve the job scheduling task in the computation system grid by maximising the utilisation rates of resources and minimising the processing cost. Anuradha et al. [116] hybridised FA with DE in developing a computationally efficient clustering technique for multi-agent systems. Rosić et al. [117] proposed an adaptive hybrid FA and DE to solve passive target localisation with proper balancing local exploitation and global exploration searches during optimisation problems.

4.5.8. DE with ACO

Rahmat et al. [118] hybridised ACO with DE to optimise the economic load dispatch of a power system integrated with renewable energy to minimise the operating cost. Zhang et al. [119] presented a hybrid algorithm of DE and ACO to learn the optimal structure of Bayesian network to enhance its convergence speed and learning accuracy. Xie et al. [120] proposed a hybrid algorithm of ACO and DE to address popular issues encountered in the cloud computing resource scheduling problem such as long processing time and uneven distribution of computing resources.

4.5.9. DE with GA

Trivedi et al. [121] hybridised GA with DE as hGADE to solve the unit commitment scheduling problem. A heuristic was incorporated into the population initialisation scheme to further enhance the performance of hGADE. Thakshaayene et al. [122] presented another hybrid algorithm of GA and DE to solve unit commitment problems, and the obtained solutions were compared with those of the conventional dynamic programming method. Li et al. [123] proposed a multi-objective optimisation algorithm by hybridising GA and DE to solve cloud computing applications.

Table 6 Hybridisation of DE algorithm with other AI algorithms.

Method hybridized	Years	Authors
DE with ANN	2018	[82]
	2018	[83]
	2019	[84]
	2020	[85]
	2020	[86]
	2021	[87]
	2021	[88]
DE with PSO	2016	[89]
	2018	[90]
	2018	[91]
	2019	[59]
	2020	[92]
	2020	[93]
	2021	[94]
DE with FL	2017	[95]
	2018	[96]
	2018	[97]
	2019	[98]
	2020	[99]
	2020	[100]
	2021	[101]
DE with CS	2016	[102]
	2018	[103]
	2019	[104]
	2019	[105]
	2020	[106]
DE with ABC	2016	[107]
	2017	[108]
	2018	[109]
	2019	[110]
	2020	[111]
DE with WOA	2018	[112]
	2020	[113]
DE with FA	2017	[114]
	2018	[115]
	2020	[116]
	2021	[117]
DE with ACO	2017	[118]
	2018	[119]
	2019	[120]
DE with GA	2016	[121]
	2017	[122]
	2018	[123]

4.5.10. Summary of hybridisation

Table 6 presents a list of hybridised methods derived from the DE algorithm, and this trend analysis shows that the hybridisation of DE with ANN, PSO and FL remains popular in the research community. Other computational intelligence algorithms such as WOA, FA, ACO and GA were observed as less favourable candidates to be hybridised with DE. Fig. 7 summarises the distributions of proposed hybrid DE variants according to year and the computational intelligence algorithms selected for hybridisation. Accordingly, ANN, PSO,

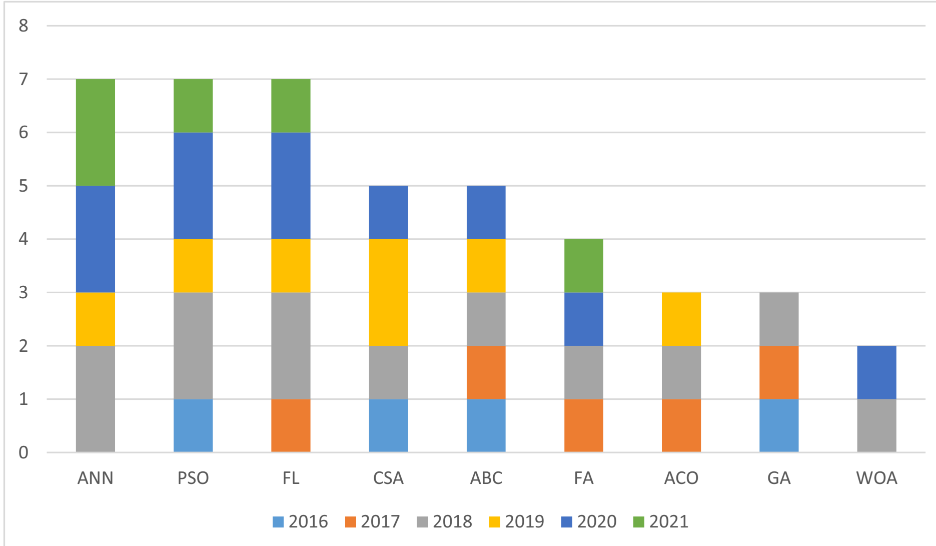

Fig. 7 Distributions of hybrid DE variants produced according to year and AI algorithms used for hybridisation.

FL and FA are the most famous computational intelligence algorithms selected to be hybridised with DE, as indicated by recent works published in 2020 and 2021. In contrast, ACO and GA are less popular choices of candidates used for hybridisation because no related works were published in 2020 or 2021.

Performance analyses of most hybrid DE algorithms were conducted by using the benchmark functions under frequently used control parameters. The performance measures of solution quality and convergence rate are not analysed due to the different natures of simulation results; hence, they are not suitable to be compared with other DE variants.

4.6. Applications of DE algorithm

The applications of the DE algorithm to solve different real-world engineering problems and standard benchmark functions are summarised in Table 7. Fifty-five articles related to the applications of original DE and its enhanced variants are covered [124–178]. Some of the published works will be summarised in another section for further performance analyses, whereas the remaining applications that do not fall in the scope of the study will be listed in this section only.

4.6.1. Prediction models

Onan et al. [124] proposed a multi-objective weighted voting ensemble classifier to solve text sentiment classification problems, where DE was applied to determine the appropriate weight values of each individual classifier on the basis of their predictive performance. The machine learning models included in their proposed ensemble method were support vector machine, logistic regression, linear discriminant analysis, naïve Bayes and Bayesian logistic regression. Hu et al. [125] applied DE to optimise the parameters and weights of least-square support vector machine. A multi-level regression model was then developed to predict the carbon efficiency to minimise the energy consumption incurred during the iron ore sintering process.

Peng et al. [126] presented a long short-term memory (LSTM) model optimised by DE to address the electricity price prediction task that can be formulated as time series and non-linear regression problems. The LSTM optimised by DE can produce higher prediction accuracy. Al-Sudani et al. [127] proposed the application of DE to optimise the design of a multi-variate adaptive regression spline developed using least square support vector regression. This prediction model was used to achieve more accurate forecasting of a streamflow pattern that plays crucial roles in effective planning and management of water resources. Both studies show the effectiveness of DE in optimising the parameters of the machine learning model to develop useful prediction models in different areas of applications.

4.6.2. Industrial control

Li et al. [128] applied DE to estimate the self-potential data widely used in geophysics with better solution quality and efficiency by optimising six parameters, namely, regional coefficients, polarisation angle, distance from the origin, depth of source and electrical dipole moment. Marčič et al. [129] proposed to use DE to simultaneously identify the mechanical, electrical and magnetic subsystem parameters of a dynamic model used to represent the objective functions and technical constraints of line-start interior permanent magnet synchronous motors. Kadhar et al. [130] proposed to use diversity controlled self-adaptive DE with local search to achieve the optimal design of a non-fragile multivariable proportional integral controller used in the wood and berry distillation process. Both studies demonstrated the potential of DE for various industrial processes.

Santucci et al. [131] proposed an abstract algebraic DE to optimise the total flow time of the permutation flow shop scheduling task. Some notable modifications introduced in this work include the heuristic-based initialisation scheme, discrete mutation operator, crossover operator inspired by the bubble sort algorithm, biased selection scheme and memetic restart scheme. Wang et al. [132] developed a new strategy to design a coaxial magnetic gear by referring to the theory related to

Table 7 Research on the applications of the DE algorithm.

Authors	Year	Applications
[124]	2016	Prediction
[125]	2018	Prediction
[126]	2018	Prediction
[127]	2019	Prediction
[128]	2012	Industrial control
[129]	2014	Industrial control
[130]	2015	Industrial control
[131]	2015	Industrial control
[132]	2019	Industrial control
[133]	2020	Industrial control
[134]	2017	Computational systems
[135]	2020	Computational systems
[136]	2020	Computational systems
[137]	2012	Electrical and power systems
[138]	2017	Electrical and power systems
[139]	2017	Electrical and power systems
[140]	2019	Electrical and power systems
[141]	2020	Electrical and power systems
[142]	2013	Feature selection
[143]	2017	Feature selection
[144]	2017	Feature selection
[145]	2018	Feature selection
[146]	2018	Feature selection
[147]	2018	Feature selection
[148]	2020	Feature selection
[149]	2020	Feature selection
[150]	2020	Feature selection
[151]	2013	Image processing
[152]	2017	Image processing
[153]	2018	Image processing
[154]	2019	Image processing
[155]	2020	Image processing
[156]	2017	Clustering
[157]	2019	Clustering
[158]	2019	Clustering
[159]	2019	Clustering
[160]	2020	Clustering
[161]	2018	Health care
[162]	2019	Health care
[163]	2019	Health care
[164]	2016	Path planning
[165]	2019	Path planning
[166]	2020	Path planning
[167]	2020	Path planning
[168]	2020	Path planning
[169]	2020	Path planning
[170]	2018	Wireless and sensor
[171]	2018	Wireless and sensor
[172]	2018	Wireless and sensor
[173]	2020	Wireless and sensor
[174]	2007	Differential equations
[175]	2013	Differential equations
[176]	2014	Differential equations
[177]	2019	Differential equations
[178]	2020	Differential equations

magnet magneto-motive force. DE was applied to search for optimal combinations of key parameters that have a significant impact on the modulation effects of magnetic gears, i.e. the thickness of permanent magnet, ratio between magnet arc to permanent magnet pole pitch and ratio between air slot opening to pole pitch. Nadimi-Shahraki et al. [133] proposed three trial vector producers (TVPs), namely, the global best history-based TVP, local random-based TVP and representative-based TVP, into their multi-trial vector-based DE (MTDE) to solve different numerical benchmark functions and four engineering design problems. Both the winner-based distribution policy and life-time archive were employed by MTDE to determine the most appropriate TVP for each subpopulation.

4.6.3. Computational systems

By leveraging the benefits of cloud computing technologies, Teijeiro et al. [134] proposed the parallel implementation of an enhanced DE by using a cloud platform known as Spark to address three-parameter estimation problems commonly found in the computational biology domains (i.e. Circadian model, NFKB model and three-step pathway model) more effectively without consuming excessive computational time. LaTorre et al. [135] applied DE in computational neuroscience model calibration. A simplified triggering technique with fixed-diameter axon was used to increase the computation processes. The finding showed that the proposed model managed to address the complex framework of nerve damages. Houssein et al. [136] proposed to apply an adaptive guided DE (AGDE) in searching for the optimal quantum cloning circuit parameters through the minimisation of cloning difference error value to improve the cloning fidelity. A new mutation operator was proposed to improve the convergence speed of AGDE by fully utilising information brought by population members with good, average and poor fitness. A self-adaptive scheme was also introduced to adjust the crossover rate of AGDE to ensure the balancing of exploration and exploitation searches.

4.6.4. Electrical and power systems

Azad et al. [137] proposed a modified DE to solve economic dispatch problems in power generation systems with minimum generation cost by incorporating a tournament selection scheme to perform pair-wise comparisons between the feasibility of solutions by referring to their degrees of constraint violations. Sakr et al. [138] proposed a modified DE (MDEA) to solve an optimal reactive power management problem encountered in an Egyptian power grid system to minimise the voltage deviation and transmission power losses simultaneously. A self-adaptive scaling factor was incorporated into MDEA to enhance its convergence speed without compromising population diversity, and a new updating strategy was also proposed to modify its penalty factor to handle different types of technical constraints effectively.

Majed et al. [139] designed a direct control strategy of distribution static synchronous compensator based on a harmonics elimination pulse-width strategy implemented using DE. In contrast to the conventional control strategy, HEPWM has a wider modulation index and is able to maintain a small harmonic distortion in output voltage. Biswas et al. [140] applied the linear population size reduction technique of success history-based adaptive DE (L-SHADE) presented to solve the parameter estimation problems of solar cells constructed from single-diode and double-diode models with minimum current–voltage errors. Ozyon et al. [141] produced the optimal solutions by using DE to solve the short-term operations of an electrical power system that consists of pumped-storage power generation units.

4.6.5. Feature selection

Ghosh et al. [142] proposed a self-adaptive DE (SADE) to address the feature subset selection problem of a hyperspectral image that suffers from high computational intensiveness and redundancy issues due to the presence of large numbers of neighbouring bands. A feature ranking technique known as Relief-F algorithm was incorporated to eliminate the duplicated features, followed by the employment of fuzzy k-nearest neighbour to perform classification tasks. Mlakar et al. [143] investigated the feasibility of applying a multi-objective version of DE (DEMO) to develop a wrapper-based feature selector (FS) for a facial expression recognition system to detect seven types of prototypical expression with minimum numbers of features and maximum accuracy of emotion recognition. Vivekanandan et al. [144] designed an FS method using a modified DE to select the critical features for a classifier constructed using the fuzzy analytical hierarchy and feed-forward neural network to predict the presence and absence of heart disease with better accuracy and shorter processing time.

Gutoski et al. [145] proposed a bio-inspired FS method with a traditional DE/rand/1/bin algorithm and combined with the K-means algorithm to solve unsupervised clustering problems with more promising values of Calinski-Harabasz and Silhouette scores, which are two metrics commonly used for cluster analyses. Nayak et al. [146] designed a feature selector for extreme learning machine to solve supervised classification problems by leveraging the promising capability of using an elitist-based multi-objective DE in optimising three objective functions related to selected feature number, classification accuracy and Minkowski score simultaneously. Yao et al. [147] incorporated two improved binary DE (BDE) algorithms into the Gaussian process regression to solve the variable selection problem in deploying the soft sensors required to monitor the process variables of industrial plants. Hancer [148] proposed a new multi-objective DE-based filter approach to perform feature selection by considering the filter criteria of fuzzy and kernel measures to obtain the optimal feature subset that can lead to the maximum predictive of responses.

Zhang et al. [149] proposed a self-learning multi-objective FS with binary DE (MOFS-BDE) to address the FS problems with the aim of maximising classification accuracy and minimising the number of selected features simultaneously. A novel binary mutation scheme based on probability difference was incorporated into MOFS-BDE to guide the solution members in locating the promising solution regions rapidly, and the self-learning capability of elite individuals in near-optimal solution regions was enhanced with a one-bit purifying search operator. The computational complexity of the selection process in MOFS-BDE was further reduced by using an efficient non-dominating sorting strategy based on crowding distance. Rivera-Lopez et al. [150] proposed a permutational-based DE algorithm to tackle feature subset selection problems without a fixed subset size to be defined in advance. The permutational-based mutation operator was designed to create new feasible solutions, and a repair-based recombination operator was proposed to maintain the population diversity during the evolution process.

4.6.6. Image processing

Sarkar et al. [151] performed a multilevel thresholding process on the 2D histogram to segment input images into several distinct regions efficiently through the maximisation of the Tsallis entropy using DE. Gong et al. [152] applied a DE to solve the superpixel segmentation problems that have become increasingly crucial in computer vision applications. The homogeneity of superpixels was achieved through the minimisation of within-superpixel errors, derivation of superpixel boundaries in natural images with boundary gradient and generation of superpixels that are human vision-friendly by using a regularisation term. Bhandari et al. [153] proposed a multi-level thresholding scheme based on the novel beta DE to address the colour segmentation problem. The optimal thresholding levels of colour images were determined via the maximisation of Tsallis and Kapur's entropy values.

Kaur et al. [154] applied a memetic DE to optimise the parameters of an intertwining logistic map to develop an image encryption technique with higher efficiency and security. The optimised intertwining map was able to produce encrypted images by generating appropriate secret keys used for encrypting the shuffled channels of colour images. Sui et al. [155] developed a parallel computation of DE (pcDE) to solve the image threshold segmentation problem with better performance and stability even in the presence of different noise signals. Two communication schemes, namely, the optimal elite strategy and mean elite strategy, were incorporated into pcDE to promote information exchange between different subpopulations by replacing the local optimal solution with the global optimal solution and the mean value of the local optimal solution in all subpopulations, respectively.

4.6.7. Data clustering

Tam et al. [156] proposed a new unsupervised learning technique known as automatic clustering differential evolution (ACDE), where DE was used to determine the optimal number of clusters. A heuristic approach was incorporated into ACDE to adjust the activation value of each cluster centroid based on the predefined quality metrics. Saini et al. [157] presented a fusion of self-organizing map (SOM) and multi-objective DE, abbreviated as SMODoc_clust, to automatically classify documents on the basis of Silhouette and Pakhira–Bandyopad hyay–Maulik indices. The multi-objective DE was used to determine the optimal cluster numbers of SMODoc_clust based on the new genetic operator inspired from SOM. Nguyen-Trang et al. [158] designed an adaptive elitist DE (aeDE) to determine the optimal values of cluster centres that can lead to better partitioning by minimising an internal validity measure that represents the compactness of established clusters. Apart from having promising global search ability, a spherical quadratic steepest descent method was also integrated into aeDE to enhance its local search ability to solve optimisation problems with better accuracy and less computational time.

Mustafa et al. [159] designed an adaptive memetic DE to improve the quality of data clustering by optimising the intra-cluster distance similarity measure. A neighbourhood selection heuristic and an adaptive DE mutation operator were integrated with a memetic algorithm to ensure that population diversity was maintained throughout the optimisation process and to guarantee the consistency of clustering results. Wu et al. [160] presented a clustering DE method assembled with crowding factors to promote the populations to eliminate the local optima. A novel clustering method, namely, k-means specialbased DE, was proposed, which enhances the diversity of the populations in the earlier stage, whereas the solution accuracy was gradually improved in the later stage.

4.6.8. Health care

Hamdi et al. [161] applied DE and support vector regression (SVR) model to predict blood glucose level in chronic disease patients. The validation was performed with real continuous glucose monitoring data of 12 patients and can be operated independently with high prediction accuracy. Wang et al. [162] proposed a complex harmonic regularisation with DE (CHR-DE) to address the biomarker selection problems that are crucial in combating cancer and genetic diseases. DE was used to optimise the hyperparameters of CHR, enabling the latter method to have a strong ability to select relevant biomarkers from gene expression data. Kaur et al. [163] designed an e-health data prediction method by applying a multi-objective DE to fine-tune the parameters of random forest technique for different medical applications, such as the diagnosis of lung cancer, skin cancer, blood cancer, breast cancer, diabetes, brain tumour and Ebola.

4.6.9. Path planning

Kok and Rajendran [164] applied DE to solve the path planning problem of an unmanned aerial vehicle with improved computational cost and better final output path via the fine-tuning of its four control parameters, namely, generation number, population size, scaling factor and crossover rate. Zamuda et al. [165] applied SHADE and L-SHADE to design an expert system that enabled higher-autonomy control of underwater glider path planning with more stable trajectories that are crucial for robotic and oceanographic applications. Chellaswamy et al. [166] proposed a method called railway track heath monitoring system DE (RHMDE) to check malfunctions in railway tracks for increasing safety measures. RHMDE can continuously report defects found on the track to the control station, hence reducing the need to conduct regular inspections.

Zuo et al. [167] proposed a case learning-based DE (CLDE) to solve an optimal design problem of interplanetary trajectory that is crucial for a space mission. Successful values of scaling factor and crossover rate were stored and retrieved by CLDE throughout the search process until these control parameters were no longer able to produce new offspring solutions with better fitness. Jain et al. [168] studied recent modifications made in DE to solve robotic path planning problems subjected to various constraints. Pan et al. [169] proposed a hybrid DE called CIJADE by combining the modified CIPDE (MCIPDE) and modified JADE (MJADE) to solve the path planning problem related to unmanned combat aerial vehicles. The main population of CIJADE was partitioned into inferior and superior subpopulations on the basis of fitness values and evolved independently using MJADE and MCIPDE, respectively. An external archive was incorporated into the mutation operator of MCIPDE to enhance its exploration capability, where a new crossover operator and dynamic strategy of determining elite size was designed in MJADE to achieve a better balancing of exploration and exploitation searches.

4.6.10. Wireless sensor network

Cespedes-Mota et al. [170] proposed a multi-objective DE applied on wireless sensor distribution network over various geometric shapes and areas to minimise energy usage and maximise network coverage area simultaneously. Cui et al. [171] presented a wireless sensor network (WSN) with DE and DV-Hop to enhance the localisation accuracy in four different network simulations. The estimated distance error can be further reduced by optimising the location estimation for all sensor nodes. Qin et al. [172] applied DE for WSNs to solve area coverage problems such as unbalanced energy consumption through the incorporation of compensation technique. Their proposed strategy covered 90% of network areas with high energy and good computation efficiency. Wu et al. [173] proposed a single-inverter wireless power transfer technique with improved DE to optimise modulated dual-frequency output and switching angle.

4.6.11. Differential equations

Differential equations are widely applied to formulate highly complex real-world application problems encountered in different sectors. However, numerical approximations of differential equations such as those approaches reported in [179–181] are tedious tasks. DE algorithms are envisioned as promising optimisation algorithms to solve these differential equations by locating the manipulated parameter vectors effectively. Chang [174] proposed the use of a DE algorithm to solve the identification problem of chaotic systems known as Chen and Lü [182] systems. The chaotic models of the Chen and Lü systems were described by the differential equations with manipulated parameter vectors that can be evolved by an optimisation algorithm to produce the optimal values. The simulation studies verified the effectiveness of the DE algorithm in locating the optimal parameters for Chen and Lü systems with few iterations. Moreover, a fast convergence rate of the DE algorithm was observed in minimising the cost function value.

In [175], a new DE algorithm was proposed to locate the feasible optimal parameters of biological dynamic system models described by using ordinary differential equations (ODEs) and continuous delay-differential equations (DDEs). This parameter estimation problem was formulated as an objective function with algebraic constraints through the incorporation of nonlinear programming and spline approximation approaches. The formulated objective function contained two major parts: the system parameter and the weight coefficient, which have essential roles in affecting the search behaviour of the algorithm. Although these two objectives have the same weightages when the conventional DE was adopted, the proposed algorithm was modified using a relaxation approach to emphasise the explorative search in the solution space of the system parameter because its main objective is to discover the unknown system parameters. The performance evaluations were performed by using two simulated studies and a biological application described by ODEs and DDEs. The proposed algorithm was more robust and efficient in locating the optimal parameter values of ODEs and DDEs. However, the performance of the proposed algorithm was not compared with that of other optimisation algorithms.

In [176], a new efficient differential evolution with population size varying scheme was introduced to solve partial differential equations. In contrast to the conventional DE with fixed population size, the population size of the proposed algorithm can be increased with a predefined maximum value. If the standard deviation of best fitness values is smaller than a predefined threshold value, then an additional individual solution will be generated using DE operators and added to the current population. Six partial differential equations (PDEs) were formulated as minimisation problems with equality constraints to evaluate the performance of the proposed algorithm in locating the approximate solutions of PDEs. The convergence rate of the proposed algorithm was twice that of other compared methods on the basis of reported function evaluation numbers.

Fateh et al. [177] introduced a differential evolution-based solver to produce optimal solutions for elliptic PDEs. Second-order elliptic equations with homogeneous and non-homogeneous forms were considered and formulated as the minimisation problems. The proposed DE-based solver was effective in solving elliptic PDEs benchmark problems with linear and nonlinear characteristics. The proposed method has a high convergence speed, producing the best fitness value for each problem in approximately 500 generations, and was also reported to solve PDEs with low computational cost, as indicated by its low processing time.

In [178], a novel mesh-free approach was introduced to solve ODEs by combining the improved Fourier periodic expansion function with the weighted least square method to reduce the approximation errors. A weighted residual method [183] was firstly applied to formulate the ODEs problems as an optimisation problem. Then, an adaptive DE algorithm was used to minimise the ODEs' residuals and the boundary condition errors. The proposed method was implemented with five optimisation algorithms to solve 20 types of ODEs with initial value problems and boundary value problems. The SHADE algorithm achieved the best accuracy in producing the optimal solutions for the majority of ODEs with average processing times.

4.6.12. Summary of applications

Table 7 presents the research related to the applications of DE variants, while Fig. 8 summarises the distributions of these studies according to their publication years and application domains. Eleven applications are listed in this survey, namely, the prediction, industrial control, computational systems, electrical power systems, feature selection, image processing, clustering, health care, path planning, wireless sensor and differential equations. Feature selection is identified as the most popular application of DE, with nine research papers having been published in this research domain. Other applications also received notable attention among researchers, with works related to these research domains having been published in recent years.

5. Performance analysis of DE

This section presents the performance analysis of DE based on the surveyed articles. This section is split into five subsections: benchmark functions, most frequently used parameter settings, most frequently used performance measures, performance studies of solution quality and trends in enhancing the DE algorithm.

5.1. Benchmark functions

The modifications of DE, namely, EDE, were evaluated with different datasets or benchmark functions. Table 8 shows the benchmark functions commonly used for evaluation purposes. Most researchers used EDE to solve optimisation problems, followed by the selection and clustering problems. The most common dimension size used for performance evaluation is 30D.

The benchmark selection process has a major impact on the results of EDE. Previously, most researchers used Rosenbrock, Rastrigin, Dejong, Griewank, Ackley, Sphere and various classical benchmark functions to solve their respective problems. Recent trends indicate that CEC benchmark suites become popular in 2011 until 2020. The latest benchmark being used is CEC 2017 because of its effectiveness in evaluating the search accuracy, search robustness and success performance of an algorithm. It also compatible for solving various func-

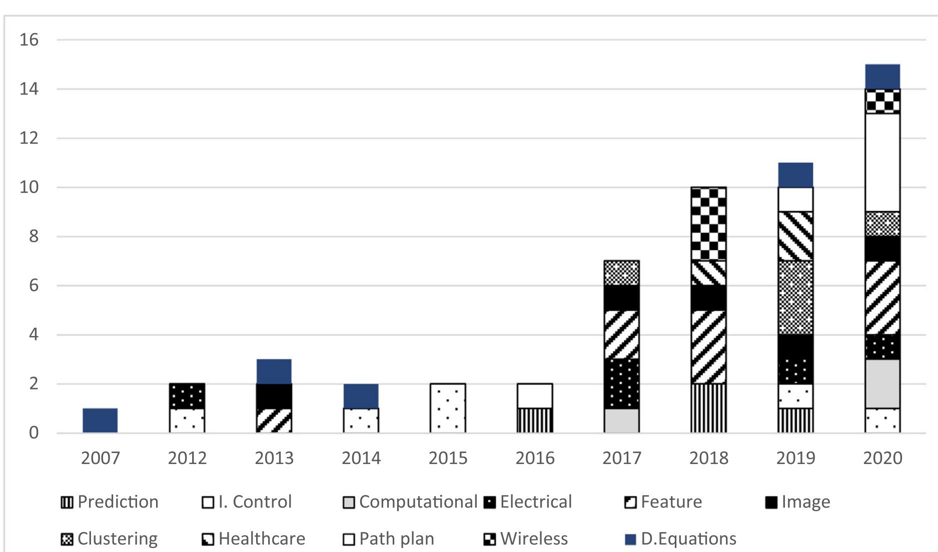

Fig. 8 Distributions of DE algorithm applications.

Table 8 Benchmark functions.

Author	Tasks	Dimensional size	Benchmark functions
[41]	Optimization	2,3,4,5,6,10,20,30,100	Numerical benchmark functions
[42]	Optimization	2, 5, 10	Rosenbrock, Dejong, Quartick, Rastrigin
[43]	Optimization	10, 20, 30, 40, 60	Ackley, Alpine, Griewank, Parabola, Rastrigin, Rosenbrock, Tripod
[44]	Optimization	30, 100	Sphere, Schwefel, Rosenbrock, Griewank, Ackley, Rastrigin, Weierstrass
[45]	Optimization	10, 30, 50, 100, 1000	CEC 2010, CEC 2013, BBOB 2010, Lennard-Jones Potential Minimization
[46]	Optimization	2, 4, 10, 30, 50	Unimodal, Step, Multimodal, Low dimensional
[47]	Optimization	10, 30, 50, 100	CEC 2014
[48]	Clustering	2, 3, 4, 5	BBC, Email
[49]	Optimization	30, 50, 100	CEC 2005
[50]	Optimization	30, 50, 100, 200	CEC 2005
[51]	Optimization	30, 50	CEC 2005
[52]	Optimization	50, 100, 200, 500, 1000	LGSO
[53]	Optimization	10, 30	CEC 2010
[54]	Optimization	Not reported	10, 37, 52, 72, 120 Bar planar truss structure
[55]	Optimization	10, 30, 50	CEC 2013
[56]	Optimization	10, 30, 50	CEC 2013
[57]	Optimization	2, 10	19 (Name not given)
[58]	Clustering	25, 50	Images (Morse, Iris, Hogg, Weather, Stock returns)
[59]	Optimization	30, 100	Mathematical model of arrival flights scheduling
[40]	Optimization	2, 50	Sphere, Schwefel, Quartic, Griewank, Rastrigin, Zakharov, Schaffer
[60]	Optimization	2, 3, 5, 10, 20, 40	BBOB 2015
[61]	Optimization	5, 10, 15	CEC 2017
[62]	Optimization	30, 50, 100	CEC 2014, CEC 2015
[63]	Optimization	30, 50, 100	CEC 2014
[64]	Optimization	30, 50	CEC 2005, CEC 2013
[65]	Optimization	30, 50	CEC 2011, CEC 2013, BBOB 2012
[66]	Optimization	30, 50	CEC 2005, CEC 2010, CEC 2012
[67]	Optimization	Not reported	Multimodal (name not given)
[68]	Optimization	50, 100	CEC 2005
[69]	Optimization	30, 50	CEC 2005
[70]	Optimization	10, 30, 50	CEC 2014
[71]	Optimization	50, 100	CEC 2005
[72]	Optimization	30, 60	CEC 2013
[73]	Optimization	30, 50, 1000	CEC 2010, CEC 2013
[74]	Optimization	1000	CEC 2010
[75]	Optimization	30, 50	CEC 2005
[76]	Clustering	2	Images (Iris, Haberman, New thyroid, Seeds, Lung cancer, Glass, Wine, Balance, Vowel, BTSCD, Heart, WDBC-Int, Dermatology, WDBC, Landsat)
[77]	Selection	10, 30, 50	CEC 2014, CEC 2015
[78]	Selection	10, 20, 30	CEC 2005, CEC 2014
[79]	Selection	30	CEC 2005, CEC 2014
[80]	Optimization	Not reported	Pareto fronts
[81]	Optimization	10, 20, 30, 40, 50	CEC 2013
[124]	Optimization	10	13 numerical benchmarks
[125]	Optimization	Not reported	Images (Lena, Vegetable, Ship)
[129]	Optimization	Not reported	Least Square Support Vector Regression (LSSVR)
[136]	Optimization	Not reported	Circadian
[139]	Optimization	Not reported	Constrained test problems (g06, g10)
[140]	Optimization	6, 15	IEEE 30, 57 bus system, WDN
[141]	Optimization	Not reported	Not reported
[142]	Optimization	3, 5	Diodes (single, double)
[145]	Selection	Not reported	Cohn Kanade, JAFFE, MMI
[146]	Selection	20, 30, 50, 100, 200, 500, 1000	Not reported
[147]	Selection	2, 10	Not reported
[148]	Selection	Not reported	Images (Iris, Glass, Breast cancer, Wine, Heart, Australian, Zoo, Vehicle, German numeric, WBCD, Dermatology, Ionosphere, Waveform, Lung cancer, Spambase, Sonar, Hill valley, Musk 1, DNA, Arrhythmia, Multiple features)
[149]	Selection	Not reported	Not reported
[150]	Clustering	Not reported	Real world and synthetic
[151]	Selection		NGSA Feature Selection (NSGAFS)

(*continued on next page*)

Table 8 (*continued*)

Author	Tasks	Dimensional size	Benchmark functions
[152]	Selection	Not reported	Arrhythmia, Australian, Cylinder-b, Dermatology, h-valley, Madelon, Musk-1, Optic, Semeion, Soybean, Splice, Vote, All-aml, Leukemia, Ovarian
[156]	Selection	Not reported	5 color images
[161]	Clustering	1, 2	Images (Iris, Wine, CMC, Glass, Cancer, Vowel)
[164]	Selection	Not reported	Breast cancer, Hepatocellular carcinoma, Colorectal cancer
[167]	Optimization	Not reported	Underwater glider path planning

tion types such as unimodal, multimodal, constrained, large-scale and hybrid. However, none of the researchers used CEC benchmark suites to evaluate the clustering task because the developed algorithms are generic.

5.2. Frequently used parameter settings

Parameter settings are one of the crucial parts to produce the optimum solution within the shortest running time. The performance of DE depends on the common parameter settings of number of population *(NP)*, mutation factor (*F*) and crossover rate (*CR*) [33,35]. All researchers worked on adjusting this value with a variety of numbers. The review based on Table 9 shows that the common value set on *NP* is 100, *F* and *CR* are 0.5. 'Not Reported' indicates that the researchers did not report the algorithm implementation values, while 'Not Applicable' means that the researchers did not perform any implementation. The total simulation runs are important because different results may be obtained in different runs. The lowest simulation runs used for performance evaluation is 10, and the highest is 100. Our survey shows that most of the researchers prefer to set 30 and 50 runs for their simulations.

5.3. Most widely used performance measures

The widely performance measures are recorded in Tables 10 and 11 for solution quality and convergence rate, respectively. The quality of algorithms cannot be judged based on a few runs because a result may vary when undergoing several runs. Additional independent runs need to be performed to obtain more accurate average results.

The experiment results in the reviewed papers were analysed and measured in various ways (best result overruns, average time over results, average results over time, etc.). However, some researchers considered only quality solutions and ignored the convergence rate, and vice versa. In addition, some researchers considered both solution quality and convergence rate. The performance measuring criteria rely on the research objective.

Most researchers tend to use success rate (SR), standard deviation (SD), mean square error (MSE), root mean square error (RMSE), Friedman rank, Wilcoxon and average fitness (AF) to evaluate the accuracy of the solution obtained during the termination of the algorithm.

5.4. Performance studies

This section presents any enhancement that is applied to the DE algorithm. All the reviewed articles used non-standard measures. A comparison between the final results of papers is unfair and impossible. Thus, an estimation comparison can be conducted among all results with a modified enhancement equation, I (%), as defined by

$$I(\%) = \left(1 - \left(\frac{M_{EDE}}{M_{DE}}\right)\right) \times 100 \tag{17}$$

where M_{EDE} and M_{DE} are metric values for EDE and DE, respectively. This equation is useful to calculate the improvement of solution quality and convergence speed rate. Several researchers compared EDE and DE, while others compared EDE with non-DE algorithms. However, a few researchers presented their research without a single comparison, thus presenting difficulty in computing the impact of EDE on the original algorithm. In addition, many researchers present final results as graphs. Thus, a graph comparison cannot be made and we reported the results as 'Graph representation'. The best case for a perfect comparison is to compare EDE with the original DE before comparing it with other non-DE algorithms.

Table 10 presents the computation of percentage improvement for solution quality. The comparison is made by comparing EDE with DE only. The highest improvement recorded is 100%, and the lowest is 0.05%. Some researchers performed a comparison with different dimensional problems. Thus, all the results need to be calculated separately because different dimensions give a different percentage value.

The computation comparison of the convergence rate of EDE over DE is presented in Table 11. The different system configurations (operating system, processor speed, RAM, etc.) of the computer used by the researchers might affect the speed of computation processes. Thus, a direct comparison of execution time is not fair because the systems run on different hardware specifications. In addition, half of the researchers did not report information on system configuration. Only seven successful complete improvements were calculated with sufficient information [41,43,44,74,138,149,156]. The enhancement made by [156] was the best as it reached 100%, and the lowest is [41] with only 10.35%, as shown in Table 11.

5.5. Trends in enhancing the DE algorithm

As highlighted in Fig. 3, the statistics show that the enhancement of DE starts to the gloom is between 2015 and 2021. As mentioned in the earlier part of this paper, DE consists of four main strategies. The most popular enhancement strategy is mutation, as the main part of this review reported on mutation, and it is followed closely by crossover. Mutation and crossover have a close relationship, with both of them showing steady increases throughout the years. The report on initialisation shows that the strategy glooms in 2010 to

Table 9 Parameter settings.

Author	Parameters	Simulation runs
[41]	NP = 100, F = 0.5, CR = 0.9	100
[42]	F = 0.6–1.9, CR = 0.5–0.8	100
[43]	Not reported	50
[44]	Not applicable	30
[45]	NP = 30, F = 0.7, CR = 0.5	50
[46]	Not applicable	50
[47]	Not reported	51
[48]	Not reported	100
[49]	NP = 100, F = 0.8, CR = 0.9	50
[50]	Not applicable	50
[51]	NP = 50, 75, 100, 125, 150, F = 0.5, CR = 0.9	50
[52]	NP = 60, F = 0.5–0.9, CR = 0.2, 0.9	25
[53]	NP = 100, F = 0.4, CR = 0.4, 0.75, 0.95	25
[54]	NP = 20, F = 0.8, CR = 0.9	30
[55]	NP = 10D, F1 = 0.3, 0.5, 0.7, 0.9, CR = 0.1, 0.3, 0.5, 0.7, 0.9	51
[56]	NP = 10, 15, 20, CR = 0.05–0.15, 0.9–1	51
[57]	NP = 200, F1 = 1, CR1 = 0.4, F2 = 0.7, CR2 = 0.8	Not reported
[58]	F = 0.6, CR = 0.8	Not reported
[59]	NP = 100, F = 0.5, 0.9, CR = 0.1, 0.9	30
[40]	NP = 50, CR = 0–1	30
[60]	NP = 100D	15
[61]	NP = 100D, 3000D, F = 0.5, CR = 1	51
[62]	NP = 5D, F = 0.6, CR = 0.8	50
[63]	NP = D, 5D, F = 0.05	50
[64]	NP = 50 & 100, F = 0.1–1, CR = 0.05–1	25, 30, 51
[65]	F = 0.7, CR = 0.5, p = 0.05	50
[66]	NP = 100, F = 0.5, CR = 0.9	30
[67]	F = 0.3, 0.5, 0.9, CR = 0.1, 0.5, 0.9	25
[68]	NP = 30 & 50, p = 0.1	30
[69]	Not applicable	30
[70]	NP = 100, F = 0.5, CR = 0.9	51
[71]	NP = 100, F = 0.5, CR = 0.5	30
[72]	NP = 5D, F = 0.4, CR = 0.5	30
[73]	NP = 100, F = 0.5 & 2, CR = 0 & 1	51
[74]	CR = 0.1, 0.5, 0.9	25
[75]	NP = 100, F = 0.5, CR = 0.5	25
[76]	NP = 100, F = 0.5, CR = 0.9	50
[77]	F = 0.5, CR = 0.5	51
[78]	NP = 20, F = 0.5, CR = 0.5	25
[79]	NP = 30, F = 0.5, CR = 0.5	30
[80]	Not reported	30
[81]	NP = 15–25	50
[124]	NP = 40	50
[125]	Not reported	Not reported
[129]	Not reported	Not reported
[136]	NP = 256, F = 0.9, CR = 0.8	20
[139]	NP = 10D, F = 0.5–1, CR = 0.8–1	30
[140]	F > 0, CR = 0.9–1	50
[141]	Not reported	Not reported
[142]	NP = 50, 80, F = 0.5, CR = 0.5	30
[145]	NP = 30	30
[146]	Not reported	Not reported
[147]	NP = 30, CR = 0.75, F = 0.01,	10
[148]	NP = 50, F = 0.3, CR = 0.9, GN = 150, Hidden neuron = 40	40
[149]	Not reported	10
[150]	NP = 50, F = 0.8, CR = 0.7	30
[151]	NP = 50, F = 0.5x*Rand*, CR = 0.3	30
[152]	F = 0.1514, CR = 0.8552	10, 30
[156]	F = Independent, CR = Independent	Not reported
[161]	NP = 50, F = 0.7, CR = 0.9	31
[164]	Not reported	50
[167]	NP = 32, 100	12

Table 10 Solution quality.

Author	Metric	EDE	DE	Improvement (%)
[41]	Acceleration rate	2,476,078	2,755,883	10.15
[42]	Not stated	83.33%	12.50%	70.83
[43]	Success rate	96%	80%	16.00
[44]	Mean fitness	95%	Not reported	–
[45]	Average fitness, SD	83.33%	16.67%	66.66
[46]	Average execution time	Graph representation	Graph representation	–
[47]	SD, Wilcoxon	10D = 3.5333	10D = 5.2667	32.91
		30D = 3.2167	30D = 6.7167	52.11
		50D = 2.8000	50D = 7.2500	61.38
		100D = 3.6833	100D = 7.5167	51.00
[48]	k-means	C2 = 0.53292	C2 = 0.64329	17.16
		C3 = 0.79893	C3 = 0.81023	1.39
		C4 = 0.82104	C4 = 0.8731	5.96
		C5 = 0.88265	C5 = 0.89101	0.94
[49]	Mean, SD	Not reported	Not reported	–
[50]	Error value	Not reported	Not reported	–
[51]	Mean, SD	Not reported	Not reported	–
[52]	Mean, SD	50D = 1.14E + 00	50D = 1.20E + 00	5
		100D = 2.85E + 00	100D = 4.20E + 00	32.14
		200D = 2.22E + 01	200D = 2.90E + 01	23.49
		500D = 3.04E + 01	500D = 9.60E + 01	68.33
		1000D = 1.94E + 02	1000D = 3.80E + 02	48.95
[53]	Average, Best, Mean	1308.48	1395.84	6.26
[54]	Mean	28.6530 Hz	28.6661 Hz	0.05
[55]	Mean, SD	Graph representation	Graph representation	–
[56]	Mean	2.13	Not reported	–
[57]	Mean, SD	1981.73	2003.35	1.08
[58]	Mean	I1 = 11.70	I1 = 12.43	5.87
		I2 = 14.30	I2 = 17.02	15.98
[59]	Friedman Rank	1.17	1.83	36.07
[40]	Mean, SD	Graph representation	Graph representation	–
[60]	Mean	Graph representation	Graph representation	–
[61]	Mean	Graph representation	Graph representation	–
[62]	Friedman Rank	CEC2014 = 1.6667	Not reported	–
		CEC 2015 = 1.8333		
[63]	Mean, SD	Graph representation	Graph representation	–
[64]	Average rank	1.73	2.57	32.68
[65]	Mean error	Graph representation	Graph representation	–
[66]	Mean, SD	Not reported	Not reported	–
[67]	Success rate	Graph representation	Graph representation	–
[68]	Mean, SD	Graph representation	Graph representation	–
[69]	Mean, SD, Wilcoxon	Graph representation	Graph representation	–
[70]	Mean, SD	33.33%	10%	23.33
[71]	Mean, SD	Not reported	Not reported	–
[72]	Mean, SD	96%	57%	39
[73]	Mean, SD	Graph representation	Graph representation	–
[74]	Mean, SD, Median	Not reported	Not reported	–
[75]	Mean	Graph representation	Graph representation	–
[76]	Mean, Median	Graph representation	Graph representation	11.98
[77]	Median, Mean	CEC2014 = 0.981004	Not reported	–
		CEC2015 = 0.946601		
[78]	Mean, SD, Wilcoxon	Not reported	Not reported	–
[79]	Mean, SD, Wilcoxon	3.46	Not reported	–
[80]	Mean	Graph representation	Graph representation	–
[81]	Mean, SD	Graph representation	Graph representation	–
[129]	RMSE, MAE	Graph representation	Graph representation	–
[136]	Mean, Median, SD	Graph representation	Graph representation	–
[139]	Wilcoxon	43057.83	43,213	0.36
[140]	Average, SD	Graph representation	Graph representation	–
[141]	Not stated	Graph representation	Graph representation	–
[142]	Mean, SD	1D = 17.90 s	Not reported	–
		2D = 18.60 s		
[145]	Recognition rate	100%	98.37%	1.63

Table 10 (*continued*)

Author	Metric	EDE	DE	Improvement (%)
[146]	Accuracy	83%	Not applicable	–
[147]	k-means	2 K = 0.23 10 K = 0.04	Not applicable	–
[148]	Average rank	2.25	Not applicable	–
[149]	RMSE	0.0026	0.0045	42.22
[150]	Not stated	2.16	Not reported	–
[151]	Friedman test	1.6099E-13	Not applicable	–
[152]	Accuracy	1.33	2.75	51.64
[156]	Unified average change intensity	Image representation	Image representation	–
[161]	Not stated	Graph representation	Graph representation	–
[164]	MSE	53.999	Not applicable	–
[167]	Median, SD	Graph representation	Graph representation	–

2015. The majority of reviewed articles on initialisation were taken from 2010 to 2015 because fewer researchers focused on this strategy recently. The least enhanced DE strategy is selection. Only five reviews were conducted on this strategy between 2017 and 2021. Even the trend shows that selection is the subject of the fewest studies, but researchers still appear to be interested in performing the selection task because all the related reviews were published in the past three years. In addition, the trend of using DE to solve real-world applications and enhancing the search performance of DE with hybridisation has increased recently.

6. Open research challenges

6.1. Encoding strategy

The encoding strategy plays a crucial role in representing the decision variables of given optimisation problems to be solved by the DE variants. Direct encoding strategy is commonly used in most literature, where each decision variable or phenotype is represented as a genome element of DE's candidate solution. While the concept of direct encoding is intuitive and explicit in handling common optimisation problems, it tends to suffer from the curse of dimensionality in the Big Data era, which requires solving sophisticated optimisation problems with extremely high numbers of decision variables [184], especially with the world defending against COVID-19 nowadays, which requires many data samples [185]. Under this scenario, DE algorithms might be struggling to solve large-scale optimisation problems with desirable accuracy or within the reasonable amount of time due to the excessive complexity involved.

6.2. Fitness value and search performance

The fitness value is commonly used by most DE variants to evaluate the quality of the new solution generated. The current solution of the DE solution can be replaced by the corresponding trial solution with a better fitness value. The fitness function of an optimisation problem needs to be constructed to compute the fitness value of any given DE solutions. Nevertheless, articulating an accurate fitness function for a complex optimisation problem is difficult because the landscapes tend to be deceptive towards local optimal solution regions and prevent the global optimum from being reached. The fitness-based selection also tends to suppress the survival of novel solutions with temporary poor performance at the beginning stage and yet can be successful in the long term if given sufficient generation [186]. A crucial task is to investigate alternative metrics other than fitness value that can be used to guide the searching process of DE to preserve novel solutions.

6.3. Optimisation problems

Most existing EAs such as DE were primarily designed to solve a single optimisation problem from scratch at one time. This concept does not truly reflect the actual scenario of real-world optimisation problems, which require several jobs to be handled at the same time. A new category of a problem known as multi-factorial optimisation (MFO) [187] has recently emerged to address this issue. In general, the MFO problem is characterised by the concurrent existence of multiple search spaces associated with different tasks that may or may not be interdependent of each other. One of the notable examples of MFO is the cloud computing platform, which receives different optimisation tasks from multiple users concurrently. To the best of the authors' knowledge, only a few studies have been conducted to leverage the benefits of EAs in designing multi-tasking optimisers that are able to solve multiple optimisation problems in MFO simultaneously with enhanced productivity.

6.4. Exploitation and exploration processes

Numerous algorithmic-specific parameters were introduced in the majority of DE variants to achieve a proper balancing of their exploitation and exploration searches. The parameter settings of these DE variants tend to be problem-dependent because different optimal combinations of parameters need to be determined to solve various types of optimisation with promising accuracy. Nevertheless, evaluating every possible combination of parameters exhaustively for each type of optimisation problems encountered is impractical because of the limitation of computation resources. The presence of inherent mathematical correlations between these newly introduced algorithmic-specific parameters can further complicate the parameter tuning process, thus being impractical for solving real-world applications.

6.5. 'No free Lunch' theorem

In 1997, Wolpert [188] proposed the 'no free lunch' (NFL) theorem, which states that search-based optimisation algorithms

Table 11 Convergence rate.

Author	Metric	EDE	DE	System configurations	Improvement (%)
[41]	NFC	2,080,795	2,321,045	Not reported	10.35
[42]	Not reported	Not reported	Not reported	Not reported	–
[43]	NFC	100,663	415,194	Not reported	75.76
[44]	Mean reliability	99.74	59.33	Not reported	40.41
[45]	Not reported	Not reported	Not reported	Not reported	–
[46]	NFE	Graph	Graph	Not reported	–
[47]	Not reported	Not reported	Not reported	PC with Core processor, 2.26 GHz, 4 GB RAM	–
[48]	Not reported	Not reported	Not reported	Laptop with Matlab R2015b, HP Quad core, 2.4 GHz	–
[49]	Not reported	Not reported	Not reported	Pentium Core 2 duo, 2.23 GHz, 2 GB RAM	–
[50]	Not reported	Not reported	Not reported	Not reported	–
[51]	Not reported	Graph	Graph	64 bit windows 7, Matlab R2012a	–
[52]	Not reported	Graph	Graph	Windows Java 1.7, Intel Xeon 8 cores, 2.4 GHz	–
[53]	Objective function	Graph	Graph	PC with Windows 7, Matlab, Core i7, 3.4 GHz, 8 GB RAM	–
[54]	Diversity index	Graph	Graph	Matlab	–
[55]	Not reported	Not reported	Not reported	Not reported	–
[56]	Not reported	Not reported	Not reported	Not reported	–
[57]	Diversity index	Graph	Graph	Not reported	–
[58]	Not reported	Not reported	Not reported	Matlab R2008b, Core i7, 12 GB RAM	–
[59]	Mean best fitness	Graph	Graph	Windows 7, Visual Studio 2008	–
[40]	Average function	Graph	Graph	PC with Matlab R2016a, Core i7, 1.8 GHz	–
[60]	Target pairs	Graph	Graph	Windows 7, Matlab R2013a, Core i5, 2.2 GHz, 4 GB RAM	–
[61]	Inverse generational distance	Graph	Graph	Not reported	–
[62]	Average function error	Graph	Graph	Windows 7, Matlab R2014a	–
[63]	Fitness error	Graph	Graph	Not reported	–
[64]	Mean error	Graph	Graph	Matlab R2007b, Core i3, 2.4 GHz, 4 GB RAM	–
[65]	Solution error	Graph	Graph	Not reported	–
[66]	NFE	Graph	Graph	Not reported	–
[67]	Not reported	Not reported	Not reported	Not reported	–
[68]	Mean fitness	Graph	Graph	Not reported	–
[69]	Not reported	Not reported	Not reported	64 bit windows 7, Matlab R2012a	–
[70]	Not reported	Not reported	Not reported	PC with Linux Ubuntu, Core i5, 3.2 GHz, 8 GB RAM	–
[71]	Value of CRm	Graph	Graph	PC with Core i7, 3.4 GHz, 16 GB RAM	–
[72]	Error fitness value	Graph	Graph	Not reported	–
[73]	Not reported	Not reported	Not reported	Not reported	–
[74]	Median	2.44E + 00	7.27E + 0.0	Not reported	66.44
[75]	Not reported	Not reported	Not reported	Not reported	–
[76]	Mean of best objective function	Graph	Graph	Not reported	–
[77]	Fitness	Fitness	Graph	Windows 7, Matlab R2014a, Core i7, 3.4 GHz, 16 GB RAM	–
[78]	Not reported	Not reported	Not reported	Matlab R2010a, Quad core, 2.83 GHz, 4 GB RAM	–
[79]	Success rate	Success rate	4.63E + 05	Not reported	–
[80]	Not reported	Not reported	Not reported	Not reported	–

Table 11 (*continued*)

Author	Metric	EDE	DE	System configurations	Improvement (%)
[81]	Not reported	Not reported	Not reported	Not reported	–
[129]	Not reported	Not reported	Not reported	Not reported	–
[136]	Time	Graph	Graph	Microsoft Azure public cloud	–
[139]	Not reported	Not reported	Not reported	Microsoft Visual Studio 9.0, Core 2 Duo, 2.5 GHz, 4 GB RAM	–
[140]	Time	4.6	41	Matlab R2012a	88.78
[141]	Not reported	Not reported	Not reported	Not reported	–
[142]	Not reported	Not reported	Not reported	Matlab, Core i5, 2.7 GHz, 4 GB RAM	–
[145]	Not reported	Not reported	Not reported	Not reported	–
[146]	Not reported	Not reported	Not reported	Not reported	–
[147]	Not reported	Not reported	Not reported	Not reported	–
[148]	Fitness	Graph	Graph	Not reported	–
[149]	Not reported	Not reported	Not reported	Not reported	–
[150]	Not reported	Not reported	Not reported	Not reported	–
[151]	SD	0.53	0.73	Not reported	27.40
[152]	Not reported	Not reported	Not reported	Not reported	–
[156]	Squared errors	0	1.51	Matlab 2.4 GHz, Core i5, 16 GB RAM	**100**
[161]	Fitness	Graph	Graph	CPU with Oracle Java 1.8, Core i7, 2.4 GHz, 8 GB RAM	–
[164]	Not reported	Not reported	Not reported	Not reported	–
[167]	Fitness	Graph	Graph	Not reported	–

perform equally well when their performance is averaged across all possible problems [189]. Hence, NFL remains an advanced open research question that shows the varieties in solving many EA problems.

7. Future research directions

7.1. Encoding strategy

Indirect encoding can be envisioned as one of the possible approaches to address the drawbacks of direct encoding in dealing with highly complex optimisation with millions of parameters. In contrast to the direct encoding strategy, which performs one-to-one mapping, indirect encoding allows multiple phenotypic elements of problem domains to be compressed into a single genome element of DE solutions, hence reducing the search space size considerably. The idea of indirect encoding is currently utilised to solve large-scale neuro-evolution problems [190], achieving promising performance. A similar indirect encoding strategy may be applied by DE algorithms to solve other large-scale optimisation problems with competitive performance provided that their underlying properties are appropriately investigated. Some potential numbering systems that can be applied to implement an indirect encoding scheme include complex number, phase angle and IP address.

7.2. Fitness value and search performance

The ideas of novelty search [191] can be incorporated into DE to address the negative impacts of poorly designed fitness functions such as poor solution diversity, which can have detrimental impacts on the long-term evolution of the population. The main purpose of integrating the novelty-driven searching process into DE is to preserve more novel solutions that might exhibit temporary poor performance but be able to contribute to notable performance enhancement in the long term. The idea of quality–diversity optimisation [192] is another interesting future direction to improve the search performance of DE in dealing with optimisation problems with challenging fitness landscapes. Unlike novelty search, both fitness and diversity criteria are considered by quality–diversity optimisation during the solution search process. Proper balancing between the fitness and diversity criteria needs to be achieved to produce a diverse yet highly performing set of DE solutions.

7.3. Optimisation problems

The implicit parallelism of DE and its variants can be harnessed to solve the multiple and diverse problems of MFO simultaneously under a multitasking environment to accelerate the optimisation process of complex problems. Some key issues need to be addressed to develop efficient and robust multitasking optimisers by using DE algorithms. Depending on the relationship between different tasks in an MFO, different encoding and decoding schemes need to be designed to map the phenotypic elements of the problem domain and the genome element of DE in a unified search space. Inspired by the assortative mating and vertical cultural transmission, i.e. two essential mechanisms of a biological concept known as multifactorial inheritance, effective knowledge transfer strategies can be designed to evolve the DE population, thus facilitating an efficient search for the knowledge of multiple tasks.

7.4. Exploitation and exploration process

Parameter-free DE variants need to be developed to eliminate the sensitivity of DE's search performance towards the changes of algorithmic-specific parameters. In contrast to most existing DE variants, parameter-free DE does not require any human intervention to determine the optimal combination of algorithmic-specific parameters in solving any given optimisation problems. One of the ways to develop parameter-free DE is to design an intelligent mechanism such as reinforcement learning, which can fine-tune all parameters adaptively based on the feedback provided by search environments such as population diversity, overall solution quality and fitness landscape of optimisation problems.

7.5. No free lunch theorem

While it might appear as a counterintuitive concept, the NFL theorem can be addressed by increasing the number of adaptive and automated processes of the DE optimiser. The suitable design of parameter adaptation mechanisms is expected to achieve a proper balancing of the exploration and exploitation search strengths of these new DE variants, thus emerging as an interesting future research direction for these DE variants to solve optimisation problems with different fitness landscapes in a more robust manner.

8. Conclusions

This paper presents a recent review based on the enhancements in DE, referred to as EDE. The performance improvement percentages of EDE over DE are computed except for some research that has not reported numerical values. Comparing the performance of the enhancements of DE among different studies is unfair because EDE runs on different conditions and benchmarks. DE has been modified by researchers to enhance its effectiveness and efficiency in solving various optimisation problems starting from 2010. This trend could continue to exponentially increase in the future due to the global attention on AI and its proficiency in solving multiple engineering problems. This statement can be proved by our review, which showed that 158 out of 192 papers were published within 2016 to 2021. All this information can assist new researchers in looking for suitable modifications of the original DE, as well as expert researchers for developing further enhancements of DE.

Our findings show that most researchers used $NP = 100$, F and $CR = 0.5$ as their parameter settings for the implementation of EDE in various domains. Thus, we suggest these values as standard DE parameter values. The approach of balancing between exploitation and exploration of the DE algorithm is also important in preventing premature convergence, thereby enhancing the quality of final solutions. A big data sample is also crucial because it can validate the accuracy of the solutions. However, a fast processing scheme is important to address the slow computing issue of this approach. The engagement of a complex number and phase angle can improve the capability of DE to solve various complex optimisation and real engineering problems. The way to overcome the NFL theorem is also discussed. Last but not least, we believe that this review can have an important impact, especially for new researchers who want to explore how the DE algorithm can be enhanced according to various approaches applied by many researchers.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Ministry of Higher Education Malaysia for Fundamental Research Grant Scheme (FRGS) with Project Code FRGS/1/2018/TK04/USM/01/2 entitled 'Enhanced Differential Evolution Algorithm by Balancing the Exploitation and Exploration Search Behavior for Data Clustering'.

References

[1] A.N. Azmi, Y. Kamin, M.K. Noordin, A.N.M. Nasir, Towards industrial revolution 4.0: employers' expectations on fresh engineering graduates, Int. J. Eng. Technol. 7(4.28) (2018) 267-272. 10.14419/ijet.v7i4.28.22593.
[2] L. Damiani, M. Demartini, G. Guizzi, R. Revetria, F. Tonelli, Augmented and virtual reality applications in industrial systems: A qualitative review towards the industry 4.0 era, IFAC-PapersOnLine 51(11) (2018) 624–630. 10.1016/j.ifacol.2018.08.388.
[3] N. Oladejo, A. Abolarinwa, S. Salawu, A. Lukman, Optimization principle and its' application in optimizing landmark university bakery production using linear programming, Int. J. Civ. Eng. Technol. (IJCIET) 10 (2) (2019) 183–190.
[4] J. Kim, K.-K.-K. Kim, Dynamic programming for scalable just-in-time economic dispatch with non-convex constraints and anytime participation, Int. J. Electr. Power Energy Syst. 123 (2020), https://doi.org/10.1016/j.ijepes.2020.106217 106217.
[5] M. Turkyilmazoglu, A simple algorithm for high order Newton iteration formulae and some new variants, Hacettepe J. Math. Statist. 49(1) (2020) 425-438. 10.15672/hujms.459810.
[6] A. Jamal, M. Tauhidur Rahman, H.M. Al-Ahmadi, I. Ullah, M. Zahid, Intelligent intersection control for delay optimization: Using meta-heuristic search algorithms, Sustainability 12(5) (2020) 1896. 10.3390/su12051896.
[7] J.D. Schaffer, P.J. Roebber, C. Evans, Development and evaluation of an evolutionary programming-based tropical cyclone intensity model, Mon. Weather Rev. 148 (5) (2020) 1951–1970, https://doi.org/10.1175/MWR-D-19-0346.1.

[8] A. Zahir, S.S.N. Alhady, A. Wahab, M. Ahmad, Objective functions modification of GA optimized PID controller for brushed DC motor, Int. J. Electrical Comput. Eng. 10(3) (2020) 2426. 10.11591/ijece.v10i3.pp2426-2433.
[9] E. Özcan, J.H. Drake, C. Altıntaş, S. Asta, A self-adaptive multimeme memetic algorithm co-evolving utility scores to control genetic operators and their parameter settings, Appl. Soft Comput. 49 (2016) 81–93, https://doi.org/10.1016/j.asoc.2016.07.032.
[10] A.D. Mehr, V. Nourani, E. Kahya, B. Hrnjica, A.M. Sattar, Z. M. Yaseen, Genetic programming in water resources engineering: A state-of-the-art review, J. Hydrol. 566 (2018) 643–667, https://doi.org/10.1016/j.jhydrol.2018.09.043.
[11] S. Li, Q. Gu, W. Gong, B. Ning, An enhanced adaptive differential evolution algorithm for parameter extraction of photovoltaic models, Energy Convers. Manage. 205 (2020), https://doi.org/10.1016/j.enconman.2019.112443 112443.
[12] O. Montiel, O. Castillo, P. Melin, A.R. Díaz, R. Sepúlveda, Human evolutionary model: A new approach to optimization, Inf. Sci. 177 (10) (2007) 2075–2098, https://doi.org/10.1016/j.ins.2006.09.012.
[13] I. Rahman, J. Mohamad-Saleh, Hybrid bio-Inspired computational intelligence techniques for solving power system optimization problems: A comprehensive survey, Appl. Soft Comput. 69 (2018) 72–130, https://doi.org/10.1016/j.asoc.2018.04.051.
[14] H. Yu, Y. Tan, J. Zeng, C. Sun, Y. Jin, Surrogate-assisted hierarchical particle swarm optimization, Inf. Sci. 454 (2018) 59–72, https://doi.org/10.1016/j.ins.2018.04.062.
[15] J. Uthayakumar, N. Metawa, K. Shankar, S. Lakshmanaprabu, Financial crisis prediction model using ant colony optimization, Int. J. Inf. Manage. 50 (2020) 538–556, https://doi.org/10.1016/j.ijinfomgt.2018.12.001.
[16] J. Chen, W. Yu, J. Tian, L. Chen, Z. Zhou, Image contrast enhancement using an artificial bee colony algorithm, Swarm Evol. Comput. 38 (2018) 287–294, https://doi.org/10.1016/j.swevo.2017.09.002.
[17] M.A. Al-Betar, M.A. Awadallah, Island bat algorithm for optimization, Expert Syst. Appl. 107 (2018) 126–145, https://doi.org/10.1016/j.eswa.2018.04.024.
[18] H. Chen, C. Yang, A.A. Heidari, X. Zhao, An efficient double adaptive random spare reinforced whale optimization algorithm, Expert Syst. Appl. 154 (2020), https://doi.org/10.1016/j.eswa.2019.113018 113018.
[19] H. Wang, W. Wang, X. Zhou, H. Sun, J. Zhao, X. Yu, Z. Cui, Firefly algorithm with neighborhood attraction, Inf. Sci. 382 (2017) 374–387, https://doi.org/10.1016/j.ins.2016.12.024.
[20] T.V. Luu, N.S. Nguyen, Parameters extraction of solar cells using modified JAYA algorithm, Optik 203 (2020), https://doi.org/10.1016/j.ijleo.2019.164034 164034.
[21] A.I. Wagan, M.M. Shaikh, A new metaheuristic optimization algorithm inspired by human dynasties with an application to the wind turbine micrositing problem, Appl. Soft Comput. 90 (2020), https://doi.org/10.1016/j.asoc.2020.106176 106176.
[22] A.K. Shukla, P. Singh, M. Vardhan, An adaptive inertia weight teaching-learning-based optimization algorithm and its applications, Appl. Math. Model. 77 (2020) 309–326, https://doi.org/10.1016/j.apm.2019.07.046.
[23] Q. Askari, I. Younas, M. Saeed, Political Optimizer: A novel socio-inspired meta-heuristic for global optimization, Knowl.-Based Syst. 195 (2020), https://doi.org/10.1016/j.knosys.2020.105709 105709.
[24] M. Li, H. Zhao, X. Weng, T. Han, Cognitive behavior optimization algorithm for solving optimization problems, Appl. Soft Comput. 39 (2016) 199–222, https://doi.org/10.1016/j.asoc.2015.11.015.
[25] Z.-K. Feng, S. Liu, W.-J. Niu, S.-S. Li, H.-J. Wu, J.-Y. Wang, Ecological operation of cascade hydropower reservoirs by elite-guide gravitational search algorithm with Lévy flight local search and mutation, J. Hydrol. 581 (2020), https://doi.org/10.1016/j.jhydrol.2019.124425 124425.
[26] B. Xing, W.-J. Gao, Electromagnetism-like Mechanism Algorithm, Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms, Springer 2014, pp. 347–354. 10.1007/978-3-319-03404-1_21.
[27] A.E. Taser, K. Guney, E. Kurt, Circular antenna array synthesis using multiverse optimizer, Int. J. Antennas Propagation 2020 (2020), https://doi.org/10.1155/2020/3149826.
[28] N. Neggaz, A.A. Ewees, M. Abd Elaziz, M. Mafarja, Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection, Expert Syst. with Appl. 145 (2020), https://doi.org/10.1016/j.eswa.2019.113103 113103.
[29] Y. Chung, I. Char, W. Neiswanger, K. Kandasamy, A.O. Nelson, M.D. Boyer, E. Kolemen, J. Schneider, Offline contextual bayesian optimization for nuclear fusion, arXiv preprint arXiv:2001.01793 (2020).
[30] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, J. Global Optim. 11 (4) (1997) 19, https://doi.org/10.1023/A:1008202821328.
[31] K. Price, R.M. Storn, J.A. Lampinen, Differential evolution: a practical approach to global optimization, Springer Science & Business Media, 2006.
[32] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2008) 398–417, https://doi.org/10.1109/TEVC.2008.927706.
[33] S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2010) 4–31, https://doi.org/10.1109/TEVC.2010.2059031.
[34] M. Pant, H. Zaheer, L. Garcia-Hernandez, A. Abraham, Differential Evolution: A review of more than two decades of research, Eng. Appl. Artif. Intell. 90 (2020), https://doi.org/10.1016/j.engappai.2020.103479 103479.
[35] F. Neri, V. Tirronen, Recent advances in differential evolution: a survey and experimental analysis, Artif. Intell. Rev. 33 (1) (2010) 61–106, https://doi.org/10.1007/s10462-009-9137-2.
[36] S. Das, S.S. Mullick, P.N. Suganthan, Recent advances in differential evolution–an updated survey, Swarm Evol. Comput. 27 (2016) 1–30, https://doi.org/10.1016/j.swevo.2016.01.004.
[37] L. Jebaraj, C. Venkatesan, I. Soubache, C.C.A. Rajan, Application of differential evolution algorithm in static and dynamic economic or emission dispatch problem: A review, Renew. Sustain. Energy Rev. 77 (2017) 1206–1220, https://doi.org/10.1016/j.rser.2017.03.097.
[38] K.R. Opara, J. Arabas, Differential Evolution: A survey of theoretical analyses, Swarm Evol. Comput. 44 (2019) 546–558, https://doi.org/10.1016/j.swevo.2018.06.010.
[39] N. Javaid, Differential evolution: An updated survey, Conference on Complex, Intelligent, and Software Intensive Systems, Springer, 2018, pp. 681-691.
[40] P. Xiao, D. Zou, Z. Xia, X. Shen, Multi-strategy different dimensional mutation differential evolution algorithm, in: AIP Conference Proceedings, AIP Publishing LLC, 2019, p. 020102.
[41] S. Rahnamayan, H.R. Tizhoosh, M.M. Salama, A novel population initialization method for accelerating evolutionary algorithms, Comput. Math. Appl. 53 (10) (2007) 1605–1614, https://doi.org/10.1016/j.camwa.2006.07.013.
[42] A.B. Ozer, CIDE: chaotically initialized differential evolution, Expert Syst. Appl. 37 (6) (2010) 4632–4641, https://doi.org/10.1016/j.eswa.2009.12.045.
[43] V.V. de Melo, A.C.B. Delbem, Investigating smart sampling as a population initialization method for differential evolution in

continuous problems, Inf. Sci. 193 (2012) 36–53, https://doi.org/10.1016/j.ins.2011.12.037.
[44] W. Zhu, Y. Tang, J.-A. Fang, W. Zhang, Adaptive population tuning scheme for differential evolution, Inf. Sci. 223 (2013) 164–191, https://doi.org/10.1016/j.ins.2012.09.019.
[45] I. Poikolainen, F. Neri, F. Caraffini, Cluster-based population initialization for differential evolution frameworks, Inf. Sci. 297 (2015) 216–235, https://doi.org/10.1016/j.ins.2014.11.026.
[46] D. Bajer, G. Martinović, J. Brest, A population initialization method for evolutionary algorithms based on clustering and Cauchy deviates, Expert Syst. Appl. 60 (2016) 294–310, https://doi.org/10.1016/j.eswa.2016.05.009.
[47] M.Z. Ali, N.H. Awad, P.N. Suganthan, R.G. Reynolds, An adaptive multipopulation differential evolution with dynamic population reduction, IEEE Trans. Cybern. 47 (9) (2016) 2768–2779, https://doi.org/10.1109/TCYB.2016.2617301.
[48] D. Mustafi, G. Sahoo, A hybrid approach using genetic algorithm and the differential evolution heuristic for enhanced initialization of the k-means algorithm with applications in text clustering, Soft. Comput. 23 (15) (2019) 6361–6378, https://doi.org/10.1007/s00500-018-3289-4.
[49] S.M. Islam, S. Das, S. Ghosh, S. Roy, P.N. Suganthan, An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (2) (2011) 482–500, https://doi.org/10.1109/TSMCB.2011.2167966.
[50] W. Gong, Z. Cai, Differential evolution with ranking-based mutation operators, IEEE Trans. Cybern. 43 (6) (2013) 2066–2081, https://doi.org/10.1109/TCYB.2013.2239988.
[51] Q. Fan, X. Yan, Self-adaptive differential evolution algorithm with discrete mutation control parameters, Expert Syst. Appl. 42 (3) (2015) 1551–1572, https://doi.org/10.1016/j.eswa.2014.09.046.
[52] M.Z. Ali, N.H. Awad, P.N. Suganthan, Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization, Appl. Soft Comput. 33 (2015) 304–327, https://doi.org/10.1016/j.asoc.2015.04.019.
[53] N.M. Hamza, D.L. Essam, R.A. Sarker, Constraint consensus mutation-based differential evolution for constrained optimization, IEEE Trans. Evol. Comput. 20 (3) (2015) 447–459, https://doi.org/10.1109/TEVC.2015.2477402.
[54] V. Ho-Huu, T. Vo-Duy, T. Luu-Van, L. Le-Anh, T. Nguyen-Thoi, Optimal design of truss structures with frequency constraints using improved differential evolution algorithm based on an adaptive mutation scheme, Autom. Constr. 68 (2016) 81–94, https://doi.org/10.1016/j.autcon.2016.05.004.
[55] K. Opara, J. Arabas, Comparison of mutation strategies in differential evolution–a probabilistic perspective, Swarm Evol. Comput. 39 (2018) 53–69, https://doi.org/10.1016/j.swevo.2017.12.007.
[56] A.W. Mohamed, P.N. Suganthan, Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation, Soft. Comput. 22 (10) (2018) 3215–3235, https://doi.org/10.1007/s00500-017-2777-2.
[57] X. Yu, X. Yu, Y. Lu, G.G. Yen, M. Cai, Differential evolution mutation operators for constrained multi-objective optimization, Appl. Soft Comput. 67 (2018) 452–466, https://doi.org/10.1016/j.asoc.2018.03.028.
[58] M. Ramadas, M. Pant, A. Abraham, S. Kumar, Segmentation of weather radar image based on hazard severity using RDE: reconstructed mutation strategy for differential evolution algorithm, Neural Comput. Appl. 31 (2) (2019) 1253–1261, https://doi.org/10.1007/s00521-017-3091-8.
[59] S. Wang, Y. Li, H. Yang, Self-adaptive mutation differential evolution algorithm based on particle swarm optimization, Appl. Soft Comput. 81 (2019), https://doi.org/10.1016/j.asoc.2019.105496 105496.
[60] S. Prabha, R. Yadav, Differential evolution with biological-based mutation operator, Eng. Sci. Technol., Int. J. 23 (2) (2020) 253–263, https://doi.org/10.1016/j.jestch.2019.05.012.
[61] A.A. Bidgoli, S. Mahdavi, S. Rahnamayan, H. Ebrahimpour-Komleh, Gde4: The generalized differential evolution with ordered mutation, International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2019, pp. 101–113.
[62] L. Deng, L. Zhang, H. Sun, L. Qiao, DSM-DE: a differential evolution with dynamic speciation-based mutation for single-objective optimization, Memetic Comput. 12 (1) (2020) 73–86, https://doi.org/10.1007/s12293-019-00279-0.
[63] G. Sun, Y. Lan, R. Zhao, Differential evolution with Gaussian mutation and dynamic parameter adjustment, Soft. Comput. 23 (5) (2019) 1615–1642, https://doi.org/10.1007/s00500-017-2885-z.
[64] M. Attia, M. Arafa, E. Sallam, M. Fahmy, An enhanced differential evolution algorithm with multi-mutation strategies and self-adapting control parameters, Int. J. Intell. Syst. Appl. 10 (4) (2019) 26, https://doi.org/10.5815/ijisa.2019.04.03.
[65] S.-M. Guo, C.-C. Yang, Enhancing differential evolution utilizing eigenvector-based crossover operator, IEEE Trans. Evol. Comput. 19 (1) (2014) 31–49, https://doi.org/10.1109/TEVC.2013.2297160.
[66] Y. Cai, J. Wang, Differential evolution with hybrid linkage crossover, Inf. Sci. 320 (2015) 244–287, https://doi.org/10.1016/j.ins.2015.05.026.
[67] S. Hui, P.N. Suganthan, Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization, IEEE Trans. Cybern. 46 (1) (2015) 64–74, https://doi.org/10.1109/TCYB.2015.2394466.
[68] Y. Xu, J.-A. Fang, W. Zhu, X. Wang, L. Zhao, Differential evolution using a superior–inferior crossover scheme, Comput. Optimiz. Appl. 61 (1) (2015) 243–274, https://doi.org/10.1007/s10589-014-9701-9.
[69] Q. Fan, Y. Zhang, Self-adaptive differential evolution algorithm with crossover strategies adaptation and its application in parameter estimation, Chemometr. Intell. Lab. Syst. 151 (2016) 164–171, https://doi.org/10.1016/j.chemolab.2015.12.020.
[70] I. Fister, A. Tepeh, I. Fister Jr, Epistatic arithmetic crossover based on Cartesian graph product in ensemble differential evolution, Appl. Math. Comput. 283 (2016) 181–194, https://doi.org/10.1016/j.amc.2016.02.034.
[71] X. Qiu, K.C. Tan, J.-X. Xu, Multiple exponential recombination for differential evolution, IEEE Trans. Cybern. 47 (4) (2016) 995–1006, https://doi.org/10.1109/TCYB.2016.2536167.
[72] L.-B. Deng, S. Wang, L.-Y. Qiao, B.-Q. Zhang, DE-RCO: rotating crossover operator with multiangle searching strategy for adaptive differential evolution, IEEE Access 6 (2017) 2970–2983, https://doi.org/10.1109/ACCESS.2017.2786347.
[73] A. Ghosh, S. Das, S.S. Mullick, R. Mallipeddi, A.K. Das, A switched parameter differential evolution with optional blending crossover for scalable numerical optimization, Appl. Soft Comput. 57 (2017) 329–352, https://doi.org/10.1016/j.asoc.2017.03.003.
[74] A.W. Mohamed, A.S. Almazyad, Differential evolution with novel mutation and adaptive crossover strategies for solving large scale global optimization problems, Appl. Computat. Intell. Soft Comput. 2017 (2017), https://doi.org/10.1155/2017/7974218.
[75] Y.-Z. Zhou, W.-C. Yi, L. Gao, X.-Y. Li, Adaptive differential evolution with sorting crossover rate for continuous optimization problems, IEEE Trans. Cybern. 47 (9) (2017) 2742–2753, https://doi.org/10.1109/TCYB.2017.2676882.
[76] M. Alswaitti, M. Albughdadi, N.A.M. Isa, Variance-based differential evolution algorithm with an optional crossover for

data clustering, Appl. Soft Comput. 80 (2019) 1–17, https://doi.org/10.1016/j.asoc.2019.03.013.
[77] K.M. Sallam, S.M. Elsayed, R.A. Sarker, D.L. Essam, Landscape-based adaptive operator selection mechanism for differential evolution, Inf. Sci. 418 (2017) 383–404, https://doi.org/10.1016/j.ins.2017.08.028.
[78] M. Tian, X. Gao, C. Dai, Differential evolution with improved individual-based parameter setting and selection strategy, Appl. Soft Comput. 56 (2017) 286–297, https://doi.org/10.1016/j.asoc.2017.03.010.
[79] J. Guo, Z. Li, S. Yang, Accelerating differential evolution based on a subset-to-subset survivor selection operator, Soft. Comput. 23 (12) (2019) 4113–4130, https://doi.org/10.1007/s00500-018-3060-x.
[80] B. Qu, J.J. Liang, Y. Zhu, P.N. Suganthan, Solving dynamic economic emission dispatch problem considering wind power by multi-objective differential evolution with ensemble of selection method, Nat. Comput. 18 (4) (2019) 695–703, https://doi.org/10.1007/s11047-016-9598-6.
[81] P. Rakshit, Improved differential evolution for noisy optimization, Swarm Evol. Comput. 52 (2020) 100628, https://doi.org/10.1016/j.swevo.2019.100628.
[82] K. Mason, J. Duggan, E. Howley, A multi-objective neural network trained with differential evolution for dynamic economic emission dispatch, Int. J. Electr. Power Energy Syst. 100 (2018) 201–221, https://doi.org/10.1016/j.ijepes.2018.02.021.
[83] J. Kumar, A.K. Singh, Workload prediction in cloud using artificial neural network and adaptive differential evolution, Future Generation Comput. Syst. 81 (2018) 41–52, https://doi.org/10.1016/j.future.2017.10.047.
[84] A. Dahou, M.A. Elaziz, J. Zhou, S. Xiong, Arabic sentiment classification using convolutional neural network and differential evolution algorithm, Computat. Intell. Neurosci. 2019 (2019), https://doi.org/10.1155/2019/2537689.
[85] R. Li, H. Zhang, Q. Zhuang, R. Li, Y. Chen, BP neural network and improved differential evolution for transient electromagnetic inversion, Comput. Geosci. 137 (2020), https://doi.org/10.1016/j.cageo.2020.104434 104434.
[86] R. Jiang, J. Zhang, Y. Tang, C. Wang, J. Feng, A Collective Intelligence Based Differential Evolution Algorithm for Optimizing the Structure and Parameters of a Neural Network, IEEE Access 8 (2020) 69601–69614, https://doi.org/10.1109/Access.628763910.1109/ACCESS.2020.2986398.
[87] B. Majhi, D. Naidu, Differential evolution based radial basis function neural network model for reference evapotranspiration estimation, SN Applied Sciences 3 (1) (2021) 1–19, https://doi.org/10.1007/s42452-020-04069-z.
[88] C.M. Saporetti, L. Goliatt, E. Pereira, Neural network boosted with differential evolution for lithology identification based on well logs information, Earth Sci. Inf. 14 (1) (2021) 133–140, https://doi.org/10.1007/s12145-020-00533-x.
[89] A. Moharam, M.A. El-Hosseini, H.A. Ali, Design of optimal PID controller using hybrid differential evolution and particle swarm optimization with an aging leader and challengers, Appl. Soft Comput. 38 (2016) 727–737, https://doi.org/10.1016/j.asoc.2015.10.041.
[90] J.-H. Zhang, Y. Zhang, Y. Zhou, Path planning of mobile robot based on hybrid multi-objective bare bones particle swarm optimization with differential evolution, IEEE Access 6 (2018) 44542–44555, https://doi.org/10.1109/ACCESS.2018.2864188.
[91] M. Song, K. Chen, J. Wang, Three-dimensional wind turbine positioning using Gaussian particle swarm optimization with differential evolution, J. Wind Eng. Ind. Aerodyn. 172 (2018) 317–324, https://doi.org/10.1016/j.jweia.2017.10.032.
[92] R. Boks, H. Wang, T. Bäck, A modular hybridization of particle swarm optimization and differential evolution, in: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, 2020, pp. 1418–1425.
[93] J. Dash, B. Dam, R. Swain, Design and implementation of sharp edge FIR filters using hybrid differential evolution particle swarm optimization, AEU-Int. J. Electron. Commun. 114 (2020), https://doi.org/10.1016/j.aeue.2019.153019 153019.
[94] Z.C. Choi, K.M. Ang, W.H. Lim, S.S. Tiang, C.K. Ang, M.I. Solihin, M.R.M. Juhari, C.E. Chow, Hybridized Metaheuristic Search Algorithm with Modified Initialization Scheme for Global Optimization, Advances in Robotics, Automation and Data Analytics: Selected Papers from ICITES 2020 1350 (2021) 172. 10.1007/978-3-030-70917-4_17.
[95] I. Ebtehaj, H. Bonakdari, Design of a fuzzy differential evolution algorithm to predict non-deposition sediment transport, Appl. Water Sci. 7 (8) (2017) 4287–4299, https://doi.org/10.1007/s13201-017-0562-0.
[96] D.K. Sahoo, R.K. Sahu, G.C. Sekhar, S. Panda, 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. 5 (3) (2018) 944–963, https://doi.org/10.1016/j.jesit.2016.12.003.
[97] A. Dixit, A. Mani, R. Bansal, Image Segmentation using Hybrid Differential evolution algorithm with Fuzzy Entropy, 2018 4th International Conference on Computing Communication and Automation (ICCCA), IEEE, 2018, pp. 1-6.
[98] R. Sharma, V. Vashisht, U. Singh, EEFCM-DE: energy-efficient clustering based on fuzzy C means and differential evolution algorithm in WSNs, IET Commun. 13 (8) (2019) 996–1007, https://doi.org/10.1049/iet-com.2018.5546.
[99] A. Jamali, R. Mallipeddi, M. Salehpour, A. Bagheri, Multi-objective differential evolution algorithm with fuzzy inference-based adaptive mutation factor for Pareto optimum design of suspension system, Swarm Evol. Comput. 54 (2020), https://doi.org/10.1016/j.swevo.2020.100666 100666.
[100] M. Babanezhad, S. Zabihi, A. Taghvaie Nakhjiri, A. Marjani, I. Behroyan, S. Shirazian, Prediction of nanofluid characteristics and flow pattern on artificial differential evolution learning nodes and fuzzy framework, ACS omega 5 (35) (2020) 22091-22098. 10.1021/acsomega.0c02121.
[101] A. Karimi, T.J. Gandomani, Software development effort estimation modeling using a combination of fuzzy-neural network and differential evolution algorithm, Int. J. Electr. Comput. Eng. (2088-8708) 11(1) (2021). 10.11591/ijece.v11i1.pp707-715.
[102] U. Mlakar, B. Potočnik, J. Brest, A hybrid differential evolution for optimal multilevel image thresholding, Expert Syst. Appl. 65 (2016) 221–232, https://doi.org/10.1016/j.eswa.2016.08.046.
[103] H. Lin, S.W. Siu, A hybrid cuckoo search and differential evolution approach to protein–ligand docking, Int. J. Mol. Sci. 19 (10) (2018) 3181, https://doi.org/10.3390/ijms19103181.
[104] Z. Zhang, S. Ding, W. Jia, A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems, Eng. Appl. Artif. Intell. 85 (2019) 254–268, https://doi.org/10.1016/j.engappai.2019.06.017.
[105] R. Chi, Y. Su, Z. Qu, X. Chi, A hybridization of cuckoo search and differential evolution for the logistics distribution center location problem, Math. Probl. Eng. 2019 (2019), https://doi.org/10.1155/2019/7051248.
[106] J. Xia, L. Zhengb, A hybrid algorithm based on cuckoo search and differential evolution for numerical optimization, Computing 4 (2020) 1–8, 10.23977/cpcs.20.
[107] W.-F. Gao, L.-L. Huang, J. Wang, S.-Y. Liu, C.-D. Qin, Enhanced artificial bee colony algorithm through differential

evolution, Appl. Soft Comput. 48 (2016) 137–150, https://doi.org/10.1016/j.asoc.2015.10.070.

[108] J. Zhou, X. Yao, DE-caABC: differential evolution enhanced context-aware artificial bee colony algorithm for service composition and optimal selection in cloud manufacturing, Int. J. Adv. Manuf. Technol. 90 (1–4) (2017) 1085–1103, https://doi.org/10.1007/s00170-016-9455-x.

[109] C. Haohao, Q.i. Xiaohui, Y. Sen, Obstacle Avoidance and Path Planning for Quadrotor based on Differential Evolution-Artificial Bee Colony Algorithm, J. Phys.: Conference Series, IOP Publishing 1087 (2018) 022030, https://doi.org/10.1088/1742-6596/1087/2/022030.

[110] S. Najari, G. Gróf, S. Saeidi, F. Gallucci, Modeling and optimization of hydrogenation of CO2: Estimation of kinetic parameters via Artificial Bee Colony (ABC) and Differential Evolution (DE) algorithms, Int. J. Hydrogen Energy 44 (10) (2019) 4630–4649, https://doi.org/10.1016/j.ijhydene.2019.01.020.

[111] E. Zorarpacı, S.A. Özel, Differentially private 1R classification algorithm using artificial bee colony and differential evolution, Eng. Appl. Artif. Intell. 94 (2020), https://doi.org/10.1016/j.engappai.2020.103813 103813.

[112] G. Xiong, J. Zhang, X. Yuan, D. Shi, Y. He, G. Yao, Parameter extraction of solar photovoltaic models by means of a hybrid differential evolution with whale optimization algorithm, Sol. Energy 176 (2018) 742–761, https://doi.org/10.1016/j.solener.2018.10.050.

[113] S. Dhabal, D.K. Saha, Image enhancement using differential evolution based whale optimization algorithm, Emerging Technology in Modelling and Graphics, Springer, 2020, pp. 619–628. 10.1007/978-981-13-7403-6_54.

[114] S. Sarbazfard, A. Jafarian, A hybrid algorithm based on firefly algorithm and differential evolution for global optimization, Int. J. Adv. Comput. Sci. Appl. 7(6) (2016) 95–106. 10.14569/ijacsa.2016.070612.

[115] T.K. Ghosh, S. Das, A novel hybrid algorithm based on firefly algorithm and differential evolution for job scheduling in computational grid, Int. J. Distributed Syst. Technol. (IJDST) 9 (2) (2018) 1–15, https://doi.org/10.4018/IJDST.2018040101.

[116] M. Anuradha, V. Ganesan, S. Oliver, T. Jayasankar, R. Gopi, Hybrid firefly with differential evolution algorithm for multi agent system using clustering based personalization, J. Ambient Intell. Hum. Comput. 12 (6) (2021) 5797–5806, https://doi.org/10.1007/s12652-020-02120-w.

[117] M.B. Rosić, M.I. Simić, P.V. Pejović, An improved adaptive hybrid firefly differential evolution algorithm for passive target localization, Soft. Comput. 25 (7) (2021) 5559–5585, https://doi.org/10.1007/s00500-020-05554-8.

[118] N. Rahmat, N. Aziz, M. Mansor, I. Musirin, Optimizing economic load dispatch with renewable energy sources via differential evolution immunized ant colony optimization technique, Int. J. Adv. Sci. Eng. Inf. Technol 7(6) (2017) 2012. 10.18517/ijaseit.7.6.2328.

[119] X. Zhang, Y. Xue, X. Lu, S. Jia, Differential-evolution-based coevolution ant colony optimization algorithm for Bayesian network structure learning, Algorithms 11 (11) (2018) 188, https://doi.org/10.3390/a11110188.

[120] X. Xie, K. Xu, X. Wang, Cloud computing resource scheduling based on improved differential evolution ant colony algorithm, in: Proceedings of the 2019 International Conference on Data Mining and Machine Learning, 2019, pp. 171–177.

[121] A. Trivedi, D. Srinivasan, S. Biswas, T. Reindl, A genetic algorithm–differential evolution based hybrid framework: case study on unit commitment scheduling problem, Inf. Sci. 354 (2016) 275–300, https://doi.org/10.1016/j.ins.2016.03.023.

[122] J. Thakshaayene, D. Kavitha, Unit commitment using hybrid genetic algorithm with differential evolution, 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), IEEE, 2017, pp. 1-6.

[123] Y. Li, S. Wang, X. Hong, Y. Li, Multi-objective task scheduling optimization in cloud computing based on genetic algorithm and differential evolution algorithm, 2018 37th Chinese Control Conference (CCC), IEEE, 2018, pp. 4489-4494.

[124] A. Onan, S. Korukoğlu, H. Bulut, A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification, Expert Syst. Appl. 62 (2016) 1–16, https://doi.org/10.1016/j.eswa.2016.06.005.

[125] J. Hu, M. Wu, X. Chen, S. Du, P. Zhang, W. Cao, J. She, A multilevel prediction model of carbon efficiency based on the differential evolution algorithm for the iron ore sintering process, IEEE Trans. Ind. Electron. 65 (11) (2018) 8778–8787, https://doi.org/10.1109/TIE.2018.2811371.

[126] L. Peng, S. Liu, R. Liu, L. Wang, Effective long short-term memory with differential evolution algorithm for electricity price prediction, Energy 162 (2018) 1301–1314, https://doi.org/10.1016/j.energy.2018.05.052.

[127] Zainab Abdulelah Al-Sudani, Sinan Q. Salih, Ahmad sharafati, Zaher Mundher Yaseen, Development of multivariate adaptive regression spline integrated with differential evolution model for streamflow simulation, J. Hydrol. 573 (2019) 1–12, https://doi.org/10.1016/j.jhydrol.2019.03.004.

[128] X. Li, M. Yin, Application of differential evolution algorithm on self-potential data, PLoS ONE 7 (12) (2012), https://doi.org/10.1371/journal.pone.0051199 e51199.

[129] T. Marčič, B. Štumberger, G. Štumberger, Differential-evolution-based parameter identification of a line-start IPM synchronous motor, IEEE Trans. Ind. Electron. 61 (11) (2014) 5921–5929, https://doi.org/10.1109/TIE.2014.2308160.

[130] K.M.A. Kadhar, S. Baskar, S.M.J. Amali, Diversity Controlled Self Adaptive Differential Evolution based design of non-fragile multivariable PI controller, Eng. Appl. Artif. Intell. 46 (2015) 209–222, https://doi.org/10.1016/j.engappai.2015.09.015.

[131] V. Santucci, M. Baioletti, A. Milani, Algebraic differential evolution algorithm for the permutation flowshop scheduling problem with total flowtime criterion, IEEE Trans. Evol. Comput. 20 (5) (2015) 682–694, https://doi.org/10.1109/TEVC.2015.2507785.

[132] Y. Wang, M. Filippini, G. Bacco, N. Bianchi, Parametric design and optimization of magnetic gears with differential evolution method, IEEE Trans. Ind. Appl. 55 (4) (2019) 3445–3452, https://doi.org/10.1109/ICELMACH.2018.8507160.

[133] M.H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, H. Faris, MTDE: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems, Appl. Soft Comput. 97 (2020), https://doi.org/10.1016/j.asoc.2020.106761 106761.

[134] D. Teijeiro, X.C. Pardo, D.R. Penas, P. González, J.R. Banga, R. Doallo, A cloud-based enhanced differential evolution algorithm for parameter estimation problems in computational systems biology, Cluster Computing 20 (3) (2017) 1937–1950, https://doi.org/10.1007/s10586-017-0860-1.

[135] A. LaTorre, M.T. Kwong, J.A. García-Grajales, R. Shi, A. Jérusalem, J.-M. Pena, Model calibration using a parallel differential evolution algorithm in computational neuroscience: Simulation of stretch induced nerve deficit, J. Computat. Sci. 39 (2020), https://doi.org/10.1016/j.jocs.2019.101053 101053.

[136] E.H. Houssein, M.A. Mahdy, M.G. Eldin, D. Shebl, W.M. Mohamed, M. Abdel-Aty, Optimizing quantum cloning circuit parameters based on adaptive guided differential evolution algorithm, J. Adv. Res. 29 (2021) 147–157, https://doi.org/10.1016/j.jare.2020.10.001.

[137] M. Azad, A. Kalam, E.M.d.G. Fernandes, A modified differential evolution based solution technique for economic dispatch problems, (2012). 10.3934/jimo.2012.1017.
[138] W.S. Sakr, R.A. El-Sehiemy, A.M. Azmy, Adaptive differential evolution algorithm for efficient reactive power management, Appl. Soft Comput. 53 (2017) 336–351, https://doi.org/10.1016/j.asoc.2017.01.004.
[139] A. Majed, Z. Salam, A.M. Amjad, Harmonics elimination PWM based direct control for 23-level multilevel distribution STATCOM using differential evolution algorithm, Electr. Power Syst. Res. 152 (2017) 48–60, https://doi.org/10.1016/j.epsr.2017.06.022.
[140] P.P. Biswas, P.N. Suganthan, G. Wu, G.A. Amaratunga, Parameter estimation of solar cells using datasheet information with the application of an adaptive differential evolution algorithm, Renew. Energy 132 (2019) 425–438, https://doi.org/10.1016/j.renene.2018.07.152.
[141] S. Özyön, Optimal short-term operation of pumped-storage power plants with differential evolution algorithm, Energy 194 (2020), https://doi.org/10.1016/j.energy.2019.116866 116866.
[142] A. Ghosh, A. Datta, S. Ghosh, Self-adaptive differential evolution for feature selection in hyperspectral image data, Appl. Soft Comput. 13 (4) (2013) 1969–1977, https://doi.org/10.1016/j.asoc.2012.11.042.
[143] U. Mlakar, I. Fister, J. Brest, B. Potočnik, Multi-objective differential evolution for feature selection in facial expression recognition systems, Expert Syst. Appl. 89 (2017) 129–137, https://doi.org/10.1016/j.eswa.2017.07.037.
[144] T. Vivekanandan, N.C.S.N. Iyengar, Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease, Comput. Biol. Med. 90 (2017) 125–136, https://doi.org/10.1016/j.compbiomed.2017.09.011.
[145] M. Gutoski, M. Ribeiro, N.M.R. Aquino, L.T. Hattori, A.E. Lazzaretti, H.S. Lopes, Feature selection using differential evolution for unsupervised image clustering, International conference on artificial intelligence and soft computing, Springer, 2018, pp. 376–385.
[146] S.K. Nayak, P.K. Rout, A.K. Jagadev, T. Swarnkar, Elitism-based multi-objective differential evolution with extreme learning machine for feature selection: A novel searching technique, Connection Science 30 (4) (2018) 362–387, https://doi.org/10.1080/09540091.2018.1487384.
[147] L. Yao, Z. Ge, Variable selection for nonlinear soft sensor development with enhanced binary differential evolution algorithm, Control Eng. Pract. 72 (2018) 68–82, https://doi.org/10.1016/j.conengprac.2017.11.007.
[148] E. Hancer, Fuzzy kernel feature selection with multi-objective differential evolution algorithm, Connection Science 31 (4) (2019) 323–341, https://doi.org/10.1080/09540091.2019.1639624.
[149] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, X.-Y. Sun, Binary differential evolution with self-learning for multi-objective feature selection, Inf. Sci. 507 (2020) 67–85, https://doi.org/10.1016/j.ins.2019.08.040.
[150] R. Rivera-López, E. Mezura-Montes, J. Canul-Reich, M.A. Cruz-Chávez, A permutational-based Differential Evolution algorithm for feature subset selection, Pattern Recogn. Lett. 133 (2020) 86–93, https://doi.org/10.1016/j.patrec.2020.02.021.
[151] S. Sarkar, S. Das, Multilevel image thresholding based on 2D histogram and maximum Tsallis entropy—a differential evolution approach, IEEE Trans. Image Process. 22 (12) (2013) 4788–4797, https://doi.org/10.1109/TIP.2013.2277832.
[152] Y.-J. Gong, Y. Zhou, Differential evolutionary superpixel segmentation, IEEE Trans. Image Process. 27 (3) (2017) 1390–1404, https://doi.org/10.1109/TIP.2017.2778569.
[153] A.K. Bhandari, A novel beta differential evolution algorithm-based fast multilevel thresholding for color image segmentation, Neural Comput. Appl. 32 (9) (2020) 4583–4613, https://doi.org/10.1007/s00521-018-3771-z.
[154] M. Kaur, V. Kumar, L. Li, Color image encryption approach based on memetic differential evolution, Neural Comput. Appl. 31 (11) (2019) 7975–7987, https://doi.org/10.1007/s00521-018-3642-7.
[155] X. Sui, S.-C. Chu, J.-S. Pan, H. Luo, Parallel compact differential evolution for optimization applied to image segmentation, Appl. Sci. 10 (6) (2020) 2195, https://doi.org/10.3390/app10062195.
[156] H.-H. Tam, S.-C. Ng, A.K. Lui, M.-F. Leung, Improved activation schema on automatic clustering using differential evolution algorithm, 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2017, pp. 1749-1756.
[157] N. Saini, S. Saha, P. Bhattacharyya, Automatic scientific document clustering using self-organized multi-objective differential evolution, Cognit. Comput. 11 (2) (2019) 271–293, https://doi.org/10.1007/s12559-018-9611-8.
[158] T. Nguyen-Trang, T. Nguyen-Thoi, T. Truong-Khac, A. Pham-Chau, H. Ao, An efficient hybrid optimization approach using adaptive elitist differential evolution and spherical quadratic steepest descent and its application for clustering, Sci. Program. 2019 (2019), https://doi.org/10.1155/2019/7151574.
[159] H.M. Mustafa, M. Ayob, D. Albashish, S. Abu-Taleb, Solving text clustering problem using a memetic differential evolution algorithm, PLoS ONE 15 (6) (2020), https://doi.org/10.1371/journal.pone.0232816 e0232816.
[160] J. Wu, L. Bie, N. Jin, L. Guo, J. Zhang, J. Tao, V. Snášel, Dual-frequency output of wireless power transfer system with single inverter using improved differential evolution algorithm, Energies 13 (9) (2020) 2209, https://doi.org/10.3390/en13092209.
[161] T. Hamdi, J.B. Ali, V. Di Costanzo, F. Fnaiech, E. Moreau, J.-M. Ginoux, Accurate prediction of continuous blood glucose based on support vector regression and differential evolution algorithm, Biocybernet. Biomed. Eng. 38 (2) (2018) 362–372, https://doi.org/10.1016/j.bbe.2018.02.005.
[162] S. Wang, H.-W. Shen, H. Chai, Y. Liang, Complex harmonic regularization with differential evolution in a memetic framework for biomarker selection, PLoS ONE 14 (2) (2019), https://doi.org/10.1371/journal.pone.0210786 e0210786.
[163] M. Kaur, H.K. Gianey, D. Singh, M. Sabharwal, Multi-objective differential evolution based random forest for e-health applications, Mod. Phys. Lett. B 33 (05) (2019) 1950022, https://doi.org/10.1142/S0217984919500222.
[164] K.Y. Kok, P. Rajendran, Differential-evolution control parameter optimization for unmanned aerial vehicle path planning, PLoS ONE 11 (3) (2016), https://doi.org/10.1371/journal.pone.0150558 e0150558.
[165] A. Zamuda, J.D.H. Sosa, Success history applied to expert system for underwater glider path planning using differential evolution, Expert Syst. Appl. 119 (2019) 155–170, https://doi.org/10.1016/j.eswa.2018.10.048.
[166] C. Chellaswamy, M. Krishnasamy, L. Balaji, A. Dhanalakshmi, R. Ramesh, Optimized railway track health monitoring system based on dynamic differential evolution algorithm, Measurement 152 (2020), https://doi.org/10.1016/j.measurement.2019.107332 107332.
[167] M. Zuo, G. Dai, L. Peng, M. Wang, Z. Liu, C. Chen, A case learning-based differential evolution algorithm for global optimization of interplanetary trajectory design, Appl. Soft Comput. 94 (2020), https://doi.org/10.1016/j.asoc.2020.106451 106451.
[168] S. Jain, V.K. Sharma, S. Kumar, Robot path planning using differential evolution, Advances in computing and intelligent systems, Springer, 2020, pp. 531–537. 10.1007/978-981-15-0222-4_50.

[169] J.-S. Pan, N. Liu, S.-C. Chu, A hybrid differential evolution algorithm and its application in unmanned combat aerial vehicle path planning, IEEE Access 8 (2020) 17691–17712, https://doi.org/10.1109/ACCESS.2020.2968119.

[170] A. Céspedes-Mota, G. Castañón, A.F. Martínez-Herrera, L.E. Cárdenas-Barrón, A.M. Sarmiento, Differential evolution algorithm applied to wireless sensor distribution on different geometric shapes with area and energy optimization, J. Network Comput. Appl. 119 (2018) 14–23, https://doi.org/10.1016/j.jnca.2018.06.006.

[171] L. Cui, C. Xu, G. Li, Z. Ming, Y. Feng, N. Lu, A high accurate localization algorithm with DV-Hop and differential evolution for wireless sensor network, Appl. Soft Comput. 68 (2018) 39–52, https://doi.org/10.1016/j.asoc.2018.03.036.

[172] N.-N. Qin, J.-L. Chen, An area coverage algorithm for wireless sensor networks based on differential evolution 1550147718796734, Int. J. Distrib. Sens. Netw. 14 (8) (2018), https://doi.org/10.1177/1550147718796734.

[173] J. Wu, W. Gong, L. Wang, A clustering-based differential evolution with different crowding factors for nonlinear equations system, Appl. Soft Comput. 98 (2021), https://doi.org/10.1016/j.asoc.2020.106733 106733.

[174] W.-D. Chang, Parameter identification of Chen and Lü systems: a differential evolution approach, Chaos, Solitons Fractals 32 (4) (2007) 1469–1476, https://doi.org/10.1016/j.chaos.2005.11.067.

[175] C. Zhan, W. Situ, L.F. Yeung, P.W.-M. Tsang, G. Yang, A parameter estimation method for biological systems modelled by ode/dde models using spline approximation and differential evolution algorithm, IEEE/ACM Trans. Comput. Biol. Bioinf. 11 (6) (2014) 1066–1076, https://doi.org/10.1109/TCBB.2014.2322360.

[176] N. Panagant, S. Bureerat, Solving partial differential equations using a new differential evolution algorithm, Math. Probl. Eng. 2014 (2014), https://doi.org/10.1155/2014/747490.

[177] M.F. Fateh, A. Zameer, S.M. Mirza, N.M. Mirza, M.S. Aslam, M.A.Z. Raja, Differential evolution based computation intelligence solver for elliptic partial differential equations, Front. Inform. Technol. Electronic Eng. 20 (10) (2019) 1445–1456, https://doi.org/10.1631/FITEE.1900221.

[178] Z. Zhang, Y. Cai, D. Zhang, Solving ordinary differential equations with adaptive differential evolution, IEEE Access 8 (2020) 128908–128922, https://doi.org/10.1109/ACCESS.2020.3008823.

[179] M.A. Alqudah, C. Ravichandran, T. Abdeljawad, N. Valliammal, New results on Caputo fractional-order neutral differential inclusions without compactness, Adv. Difference Equations 2019 (1) (2019) 1–14, https://doi.org/10.1186/s13662-019-2455-z.

[180] C. Ravichandran, K. Logeswari, F. Jarad, New results on existence in the framework of Atangana-Baleanu derivative for fractional integro-differential equations, Chaos, Solitons Fractals 125 (2019) 194–200, https://doi.org/10.1016/j.chaos.2019.05.014.

[181] S.K. Panda, C. Ravichandran, B. Hazarika, Results on system of Atangana-Baleanu fractional order Willis aneurysm and nonlinear singularly perturbed boundary value problems, Chaos, Solitons Fractals 142 (2021), https://doi.org/10.1016/j.chaos.2020.110390 110390.

[182] S. Chen, J. Lü, Parameters identification and synchronization of chaotic systems based upon adaptive control, Phys. Lett. A 299 (4) (2002) 353–358, https://doi.org/10.1016/S0375-9601(02)00522-4.

[183] C.A. Fletcher, Weighted Residual Methods, Computational Techniques for Fluid Dynamics 1, Springer, 1998, pp. 98-162. 10.1007/978-3-642-58229-5_5.

[184] X. Liu, Analyzing the impact of user-generated content on B2B Firms' stock performance: Big data analysis with machine learning methods, Ind. Mark. Manage. 86 (2020) 30–39, https://doi.org/10.1016/j.indmarman.2019.02.021.

[185] D. Ding, B. del Pozo Cruz, M.A. Green, A.E. Bauman, Is the COVID-19 lockdown nudging people to be more active: a big data analysis, BMJ Publishing Group Ltd and British Association of Sport and Exercise Medicine, 2020.

[186] A. Abdollahi, M.A. Moghaddam, S.A.H. Monfared, M. Rashki, Y. Li, Subset simulation method including fitness-based seed selection for reliability analysis, Eng. Comput. (2020) 1–17, https://doi.org/10.1007/s00366-020-00961-9.

[187] S. Yao, Z. Dong, X. Wang, L. Ren, A multiobjective multifactorial optimization algorithm based on decomposition and dynamic resource allocation strategy, Inf. Sci. 511 (2020) 18–35, https://doi.org/10.1016/j.ins.2019.09.058.

[188] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82, https://doi.org/10.1109/4235.585893.

[189] T. Joyce, J.M. Herrmann, A review of no free lunch theorems, and their implications for metaheuristic optimisation, Nat.-Inspired Algorithms Appl. Optimiz. (2018) 27–51, https://doi.org/10.1007/978-3-319-67669-2_2.

[190] K.O. Stanley, J. Clune, J. Lehman, R. Miikkulainen, Designing neural networks through neuroevolution, Nat. Mach. Intell. 1 (1) (2019) 24–35, https://doi.org/10.1038/s42256-018-0006-z.

[191] I. Fister, A. Iglesias, A. Galvez, J. Del Ser, E. Osaba, I. Fister Jr, M. Perc, M. Slavinec, Novelty search for global optimization, Appl. Math. Comput. 347 (2019) 865–881, https://doi.org/10.1016/j.amc.2018.11.052.

[192] J.K. Pugh, L.B. Soros, K.O. Stanley, Quality diversity: A new frontier for evolutionary computation, Front. Robot. AI 3 (2016) 40, https://doi.org/10.3389/frobt.2016.00040.