Engineering Applications of Artificial Intelligence 121 (2023) 106008

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Particle Swarm Optimization or Differential Evolution—A comparison

Adam P. Piotrowski [a,*], Jaroslaw J. Napiorkowski [a], Agnieszka E. Piotrowska [b]

[a] Institute of Geophysics, Polish Academy of Sciences, Ks. Janusza 64, 01-452 Warsaw, Poland
[b] Faculty of Polish Studies, University of Warsaw, Krakowskie Przedmiescie 26/28, 00-927 Warsaw, Poland

ARTICLE INFO

Keywords:
Particle Swarm Optimization
Differential Evolution
Evolutionary algorithms
Swarm intelligence
Metaheuristics

ABSTRACT

In the mid 1990s two landmark metaheuristics have been proposed: Particle Swarm Optimization and Differential Evolution. Their initial versions were very simple, but rapidly attracted wide attention. During the last quarter century hundreds of variants of both optimization algorithms have been proposed and applied in almost any field of science or engineering. However, no broader comparison of performance between both families of methods has been presented so far. In the present paper ten Particle Swarm Optimization and ten Differential Evolution variants, from historical ones from the 1990s up to the most recent ones from 2022, are compared on numerous single-objective numerical benchmarks and 22 real-world problems. On average Differential Evolution algorithms clearly outperform Particle Swarm Optimization ones. Such advantage of Differential Evolution over Particle Swarm Optimization is in contradiction with popularity: In the literature Particle Swarm Optimization algorithms are two–three times more frequently used than Differential Evolution ones. Problems for which Particle Swarm Optimization performs better than Differential Evolution do exist but are relatively few. Although this result may be an effect of the choice of specific variants, experimental settings or problems used for comparison, some re-consideration of algorithmic philosophy may be needed for Particle Swarm Optimization variants to make them more competitive.

1. Introduction

Evolutionary Algorithms and Swarm Intelligence methods compose a significant fraction of metaheuristics (Boussaid et al., 2013; Sorensen et al., 2018) and are commonly applied to solve optimization problems in virtually any field of science or industry, from physics (Lv et al., 2011; Larsen et al., 2017), astronomy (Jontof-Hutter et al., 2015; Grimm et al., 2018), biology (El-Hussieny et al., 2016; Fan et al., 2020), chemistry (Ani et al., 2023; Nowak-Sliwinska et al., 2016), energy production (Amaral and Castro, 2017; Biswas et al., 2018; Wang et al., 2022), neural networks training and design (Ilonen et al., 2003; Bouaziz et al., 2019; Xue et al., 2021, 2022; Zhou et al., 2023), and various fields engineering (Eiben and Smith, 2015; Houssein et al., 2022; Jia et al., 2022; Tang et al., 2023), to humanities (Neggaz and Benyettou, 2009; Abualigah and Khader, 2017) or even interactions between humanities and biology (Shir et al., 2022). In 1995 two families of methods were proposed that have somehow revolutionized the field, namely Differential Evolution (DE) (Storn and Price, 1995, 1997) and Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995; Shi and Eberhart, 1998). Both methods very quickly achieved wide-scale attention (Parouha and Verma, 2022), during 25 years sprinkled into hundreds of variants (Das et al., 2016; Bonyadi and Michalewicz, 2017a), and become extremely popular tools to solve versatile kinds of optimization problems (Darwish et al., 2020; Bilal et al., 2020).

Both DE and PSO algorithms are population-based optimization methods. Both algorithms in each iteration move the population of solutions across the search space. In the simplest application to single-objective, non-dynamic, numerical unconstrained optimization problems (see Boussaid et al., 2013), the goal is to find a single global optimum of the objective function within the search bounds. The search is either terminated when the computational budget (the number of times the objective function may be called) is exhausted (e.g. Awad et al., 2016), after which the quality of obtained results (performance) is analyzed, or when some pre-assumed value of the objective function is reached (e.g. Varelas et al., 2020), after which the number of function calls needed to reach that value is compared. In any case, DE and PSO algorithms move their individuals across the search space in a different manner. Both DE and PSO moves are to some extent randomized (e.g. Zelinka et al., 2013; Zamuda and Brest, 2015). However, each move of DE algorithms is mainly a function of the current location of DE solutions in the search space. Only if the new location is better than the previous one, the particular DE individual is moved to the new position, otherwise it rejects the new position and stays in the previous location. DE algorithms generally remember just the current locations and objective function values. In PSO, particles are moving to the new location irrespectively of the performance but remember the best location they have visited so far, as well as the size and direction of the

* Corresponding author.
E-mail address: adampp@igf.edu.pl (A.P. Piotrowski).

https://doi.org/10.1016/j.engappai.2023.106008
Received 6 October 2022; Received in revised form 20 January 2023; Accepted 14 February 2023
Available online 11 March 2023
0952-1976/

recent move. The move of PSO individuals is governed by their current location, the location of the best historical position, the size of the recent move, and the best locations visited so far by other individuals in the swarm. As a result, the functioning of both DE and PSO highly differs in practice. However, although historically DE is considered to be Evolutionary Algorithm, and PSO to be Swarm Intelligence method, some researchers point out that due to the one-to-one nature of the selection mechanism used, DE could also be considered a kind of Swarm Intelligence approach, such as PSO (Cheng et al., 2013; Neri et al., 2013; Neri and Tirronen, 2010). Both DE and PSO algorithms have at least three control parameters (in the simplest variants, much more in more complicated ones) that may be set by the user or adaptively modified during search.

According to the bibliometric indices PSO variants are two-three times more popular among users than DE algorithms. However, DE methods achieved more success in competitions aiming at comparison between Evolutionary Computation methods, in which they frequently win or reach competing positions (Skvorc et al., 2019; Price et al., 2019; Bujok and Kolenovsky, 2022). Nonetheless, despite so much interest in application, and effort in developing both kinds of methods, there are very few studies aiming at a direct comparison between DE and PSO variants on wider sets of problems. In 2004, a classical DE version was compared against two PSO variants on 36 mathematical functions (Vesterstrøm and Thomson, 2004). In 2013 again, the basic DE and PSO were compared, along with two other algorithms (Civicioglu and Besdok, 2013). Similar basic variants were also compared against each other, and against ten other algorithms much more recently (Ezugwu et al., 2020). All these studies showed a clear advantage of the basic DE algorithm over the basic PSO competitor. However, a similar comparison presented in Lim and Haron (2013) suggested the superiority of the basic PSO. Unfortunately, such studies were based solely on simple, initial variants of PSO or DE. A larger, wide-scale comparison between various 38 optimization algorithms on specific kind of problems has been presented in Price et al. (2019). However, this study tests only two PSO variants, against many DE-based ones, and does not comment on the relative performance of both kinds of methods. Another comparison presented in Liao et al. (2015) was based on numerous problems but included just seven algorithms, among them only a single PSO and a single DE variant. In Bujok et al. (2019) one finds another interesting comparison between almost 20 different optimization algorithms. However, not all algorithms were applied to all considered problems, and again only two PSO variants, including the basic version from 1995, were applied, hence the readers cannot compare the performance of DE and PSO from this study. In Piotrowski et al. (2017) a comparison among 32 algorithms, including many DE and five PSO variants, was presented for various problems and different computational budgets. Although PSO variants were again much fever than DE-based ones, from the paper it seemed that PSO methods are inferior to DE algorithms, with exception of the lowest computational budgets, where PSO prevails. However, no discussion on the relative performance of both kinds of algorithms was given. In Parouha and Verma (2022) a new hybrid adaptive DE-PSO algorithm is introduced, and a comparison between some PSO and DE algorithms is performed. The comparison is based on various artificial benchmark functions and three real-world problems. Unfortunately, different algorithms are compared on different problems, hence, apart from the classical finding that the new hybrid is better than its competitors, no direct conclusions on the superiority of DE or PSO may be inferred from this study.

Both PSO and DE variants have been numerous times compared on solving specific kind of practical problems, like economic dispatch to minimize total production costs (Pattanaik et al., 2019), optimization of wells localization in groundwater research (Redoloza and Li, 2021), planning the trajectory of space travel (Zuo et al., 2022), optimization of multilevel image thresholding (Hammouche et al., 2010), appropriate automatic test case generation (Su et al., 2022) and numerous other engineering applications (see review in Shukla et al., 2019). However, the results from single-problem oriented studies cannot be generalized. PSO and DE algorithms are also frequently used together as competitors in papers in which new optimization algorithms are introduced. However, such studies aim at showing the advantage of the new approach, hence they are inadequate for comparison of existing families of methods — as may be concluded e.g. from the Parouha and Verma (2022) paper discussed above. Moreover, in such papers often either majority of competing algorithms are from PSO, or from DE, or from other kind of optimization methods — depending on the nature of the newly proposed algorithm, what makes the direct comparison between DE and PSO families impossible.

In the present study we aim at a detailed comparison between DE and PSO variants that were proposed during the whole history of their research, from the beginning in the 1990s to 2022. Ten different variants of each family of methods are chosen for comparison. The selected variants were, in pairs, proposed at the same time of research, and in high-quality papers. Four different sets of problems are used to perform nine separate competitions between PSO and DE algorithms. One of the considered sets is composed solely of 22 real-world problems – in a sense in which this phrase is widely used in the Evolutionary Computation community, which should not be misunderstood as the applications that serve humanity in solving practical goals (see Michalewicz, 2012, for an intriguing discussion). The other three benchmarks are based on mathematical minimization functions. Only single-objective non-dynamic unconstrained numerical optimization problems are addressed in this study (Boussaid et al., 2013). We follow the approach when the number of allowed function calls is fixed, and the algorithms are compared according to the quality of solutions found during the available computational budget (Awad et al., 2016). From competing algorithms, we have excluded all variants that we have introduced ourselves, to sustain our neutrality in research. The only goal of the present study is to answer a simple question: in practice, to solve my optimization problem should I use rather Particle Swarm Optimization or Differential Evolution?

2. Brief literature review

DE and PSO algorithms have been developed simultaneously for over 25 years (Bilal et al., 2020; Camacho-Villalon et al., 2022). Both algorithms were initially proposed in the late XX'th century as a very simple heuristic optimization tools (Storn and Price, 1995; Kennedy and Eberhart, 1995) that can be coded in a dozen of lines and may be easily understood by new readers without much background in the field (Rana et al., 2011; Al-Dabbagh et al., 2018). Since then, both DE and PSO families of methods sprinkled into hundreds of variants (Bonyadi and Michalewicz, 2017a; Das et al., 2016), among which some remained relatively simple (to find examples, see Liu et al., 2019; Sun et al., 2020), but others become more and more advanced (e.g. Tanabe and Fukunaga, 2014; Qin et al., 2016; Meng and Yang, 2022; Stanonov et al., 2022; Xia et al., 2022). At some stage it turned out that particular DE or PSO variants may be unnecessarily complicated, such that their simplification may not only facilitate our understanding of the algorithm's behavior but also improve the performance (Piotrowski and Napiorkowski, 2018; Caraffini et al., 2019). Although theoretical studies on metaheuristics are more limited than empirical ones (Sorensen et al., 2018), some proofs of convergence of the specific PSO versions have already appeared in the literature (van der Berg and Engelbrecht, 2010; Leboucher et al., 2016; Huang et al., 2022). On the contrary, proving the convergence of DE variants turned out more challenging task (Locatelli and Vasile, 2015; Hu et al., 2016; Opara and Arabas, 2019).

It is well known that swarm intelligence and evolutionary algorithms are sensitive to their control parameters. In numerous studies it has been analyzed to what extent DE algorithms are affected by the scaling factor (Weber et al., 2011; Segura et al., 2015; Zhu et al., 2020; Dhanalakshmy et al., 2022), crossover (Zaharie, 2009; Lin et al., 2011; Weber et al., 2013; Wang et al., 2014; Stanonov et al., 2021) or population size (Mallipeddi and Suganthan, 2008; Piotrowski, 2017). Also, the impact of problem's rotation and algorithm's structural bias on the performance of DE has been analyzed (Caraffini and Neri, 2019; Caraffini et al., 2019; Kononova et al., 2021). One may also find many important studies on the major PSO control parameters, including coefficients of movement (Clerc and Kennedy, 2002; Bonyadi and Michalewicz, 2017b; Cleghorn and Engelbrecht, 2018; Harrison et al., 2018; Cleghorn and Stapleberg, 2022), inertia weight (Eberhart and Shi, 2000; Chatterjee and Siarry, 2006; Cipriani et al., 2022) or population size (van der Berg and Engelbrecht, 2001; Piotrowski et al., 2020). Also, it is known that the choice of the specific topology may affect the performance of both PSO and DE algorithms (Kennedy, 1999; Dorronsoro and Bouvry, 2011; Lim and Isa, 2014; Lynn et al., 2018).

Development of novel DE or PSO variants could follow very different paths – for example compact DE (Mininno et al., 2010) or PSO (Neri et al., 2013) algorithms throw away the classical idea of populations and use its probabilistic representation instead. However, there are some classical ways of creating novel DE or PSO variants. Many DE and PSO variants use multiple search strategies together – famous historical examples include algorithms proposed by Qin et al. (2009), Iacca et al. (2014), Tanabe and Fukunaga (2014) or Lynn and Suganthan (2017). A general review of multi-strategy algorithms may be found in Wu et al. (2019), a similar survey on DE variants has been given by Mohamed et al. (2021). Another classical way for developing new DE or PSO versions is making their control parameters adaptive, which makes the algorithm more flexible and allows users to avoid the subjective choices of the control parameters that could highly affect the performance. Detailed, a few years old reviews of such adaptive DE and PSO algorithms may be found in Harrison et al. (2018) or Al-Dabbagh et al. (2018), some interesting recent examples include adaptive variants proposed by Viktorin et al. (2019), Zamani et al. (2021), Meng et al. (2021) and Flori et al. (2022). One may point here at an interesting observation: despite each year more papers on PSO are published than on DE, the number of adaptive DE variants proposed so far is much higher than the number of adaptive PSO methods. Another famous way of developing new versions of metaheuristics is their hybridization (Parouha and Verma, 2021) – both DE and PSO algorithms have been hybridized with various other algorithms, like gravitational search (Fuqing et al., 2018; Kar et al., 2020; Khan and Ling, 2021), cultural algorithm (Sun et al., 2012; Awad et al., 2017a), cuckoo search (Zhang et al., 2019b; Kumar et al., 2021), simulated annealing (Javidrad and Nazari, 2017; Potthuri et al., 2018) or genetic algorithms (Kao and Zahara, 2008; Trivedi et al., 2015); DE and PSO were also frequently hybridized with each other (Xin et al., 2012; Parouha and Verma, 2022). More details on the diversity of both DE and PSO algorithms, and on the development of ideas that lead to novel versions may be found in a large number of survey papers that covered both the development of DE (Neri and Tirronen, 2010; Das et al., 2016; Opara and Arabas, 2019; Bilal et al., 2020; Ahmad et al., 2022), PSO (Poli et al., 2007; Marini and Walczak, 2015; Bonyadi and Michalewicz, 2017a; Elbes et al., 2019; Sengupta et al., 2019; Shami et al., 2022), as well as their hybrids (Das et al., 2008; Xin et al., 2012; Parouha and Verma, 2022). Interested readers are referred to the mentioned papers for a more comprehensive discussion.

3. Methods

3.1. Algorithms tested

As given above, plenty of DE and PSO variants exist, hence the way of choosing competitive versions may be crucial for any study based on computational tests. For comparison between PSO and DE families of optimization methods we have selected ten DE and ten PSO variants, specified in Table 1. The choice of specific variants is, as always, to some extent subjective, but we tried to choose variants of similar popularity, published at a similar time, and in similarly reputable sources. To perform a fair test, variants were selected in pairs (one DE against one PSO variant) from the whole history of DE and PSO development, hence from the late 1990s up to 2022. Variants in each pair were published in a similar year: the first pair (algorithms 1st and 11th in Table 1) is composed of the basic DE from 1997 and the basic PSO with inertia weight from 1998, the last pair (algorithms 10th and 20th in Table 1) is composed of N-L-SHADE and PSO-sono that were proposed in 2022. All 20 variants are tested and compared together on every chosen benchmark set. Only variants introduced in the major computer science journals or proceedings of the main competitions between Evolutionary Algorithms are considered. The control parameters of tested algorithms are set to the values suggested in their source papers, with a few exceptions marked and justified in Table 1. Due to diversified ways of setting DE and PSO population size and initial PSO velocities proposed for various variants, for each variant we explicitly specify the population size and the approach to velocity initialization in Table 1.

3.2. Test problems

To perform a wide scale comparison, four different collections of benchmark problems are applied in this study, including three sets of mathematical functions (CEC 2014, Liang et al., 2013, composed of 30 functions, CEC 2017, Awad et al., 2016, composed of another 30 functions, and CEC 2020, Yue et al., 2019, composed of 10 functions) and a set of 22 real-world problems (CEC 2011, Das and Suganthan, 2010). Real-world tasks contain 1- to 216-dimensional problems, which are solved with up to 150 000 function calls (see Das and Suganthan, 2010). 10- and 50-dimensional versions of CEC 2014 and CEC 2017 functions are used (called CEC 2014_10, CEC2014_50, and CEC 2017_10, CEC2017_50, respectively), and the maximum number of allowed function calls per problem is set to $10\,000D$, where D is the problem dimensionality, following Liang et al. (2013) and Awad et al. (2016). CEC 2020 problems were proposed in 5-, 10-, 15- and 20-dimensional versions, and according to Yue et al. (2019) 50 000 function calls were allowed for the 5-dimensional version, 1 000 000 function calls for the 10-dimensional version, 3 000 000 function calls for the 15-dimensional version, and 10 000 000 function calls for the 20-dimensional version. In this study we use all four versions of CEC 2020 problems (called CEC 2020_5 to CEC 2020_20).

Considering various dimensionalities, DE and PSO variants are compared on nine different sets of problems (one CEC 2011; two CEC 2014; two CEC 2017; and four CEC 2020). This should allow readers to compare the performances of PSO and DE variants on problems with various dimensionalities, various sources, and very different computational budgets.

3.3. Comparison criteria

For every problem each algorithm has been run 51 times, as suggested in Liang et al. (2013) and Awad et al. (2016). For each run the solution with the lowest value of the objective function is remembered, hence we obtain a sample of 51 results per problem for each algorithm. Then, for each algorithm and problem we report the 51-runs averaged performance.

Twenty algorithms are ranked for each problem from the best one (rank 1) to the worst (rank 20), according to the averaged performance. If the difference in the performance of some algorithms on the specific problem is lower than 10^{-8}, these algorithms are given equal rank (Liang et al., 2013; Awad et al., 2016). Finally, ranks are averaged over all problems from the particular set. The algorithm with the lowest average rank is considered the best for the particular set of problems.

However, averaged ranks may focus attention on non-specialized methods. Some DE or PSO variants may perform very well for some specific problems but below the average on others. To find whether

Table 1
Algorithms compared. D – problem dimensionality. [MIN_i, MAX_i] – bounds for ith dimension. DE-based variants are numbered 1–10 and are given in blue, PSO-based ones are numbered 11–20 and are given in red (Awad et al., 2017b; Cui et al., 2016; Chen et al., 2013; Ghosh et al., 2022; Gong et al., 2016; Li et al., 2016; Liang et al., 2006; Lynn and Suganthan, 2015; Meng and Pan, 2019; Meng et al., 2022; Pan et al., 2020; Tang et al., 2015; Xia et al., 2020; Zhang et al., 2019a; Zhu et al., 2013).

Algorithm's number	Short name	Descriptive name	Reference	Year	Comments
1	DE	Differential Evolution	Storn and Price, 1997	1997	Population size = 5D (but within [10,500]), F = 0.8, CR = 0.5, DE/rand/1 mutation.
2	SADE	Self-Adaptive DE	Qin et al., 2009	2009	Population size = 50.
3	ATPS-DE	JADE with adaptive population tuning	Zhu et al., 2013	2013	Population size is adaptive during run, initialized with 5D (but within [50,200]).
4	IDE	DE with an individual independent mechanism	Tang et al., 2015	2015	Population size depends on the dimensionality. It is set to 50 for up to 10-dimensional problems, to 200 for over 50-dimensional problems, and is scaled linearly between 50 and 200 for problems with dimensionalities between 10 and 50.
5	MPADE	Adaptive DE with sub-populations	Cui et al., 2016	2016	Total population size = 200.
6	HMJCDE	Hybrid memetic CoDE and JADE	Li et al., 2016	2016	Population size = 100.
7	L-SHADE-cnEpSin	Ensemble sinusoidal parameter adaptation L-SHADE	Awad et al., 2017b	2017	Population size is linearly decreased from 18D at the beginning of the search to 4 at the end.
8	HARD-DE	Hierarchical archive-based DE	Meng and Pan 2019	2019	Population size is parabolically (quicker at the end of the search) decreased during run from $25\,ln(D)\sqrt{D}$ to 4.
9	CIJADE	Hybrid DE	Pan et al., 2020	2020	Population size = 100.
10	N-L-SHADE	Neighborhood-based L-SHADE	Ghosh et al., 2022	2022	Population size is linearly decreased from 18D at the beginning of the search to 4 at the end. Control parameters are adapted according to L-SHADE rules, but only among the $\sqrt{Np}$ closest solutions, where Np is the current population size.
11	PSO	Particle Swarm Optimization	Shi and Eberhart, 1998	1998	Population size = 40, $c_1 = c_2$ = 1.49, inertia weight decrease linearly with time within [0.9,0.4]. Velocities are initialized randomly within [0.2·MIN_i, 0.2· MAX_i] interval.
12	CLPSO	Comprehensive learning PSO	Liang et al., 2006	2006	Population size = 40. Velocities are initialized randomly within [0.2·MIN_i, 0.2· MAX_i] interval.
13	ALC-PSO	PSO with ageing leader	Chen et al., 2013	2013	Population size = 20. Velocities are initialized to 0.
14	HCLPSO	Heterogeneous comprehensive learning PSO	Lynn and Suganthan, 2015	2015	Population size = 40. Velocities are initialized randomly within [0.2·MIN_i, 0.2· MAX_i] interval.
15	IILPSO	Inter-swarm interactive learning PSO	Qin et al., 2016	2016	Population size = 50 (divided into two swarms). Velocities are initialized randomly within [0.125·MIN_i, 0.125· MAX_i] interval.
16	GLPSO	Genetic learning PSO	Gong et al., 2016	2016	Population size = 50. Velocities are initialized randomly within [0.2·MIN_i, 0.2· MAX_i] interval.
17	EPSO	Ensemble of PSO variants	Lynn and Suganthan, 2017	2017	Population size = 40, divided into two uneven swarms. Velocities are initialized randomly within [MIN_i,MAX_i] interval.
18	DEPSO	Dual environmental PSO	Zhang et al., 2019a	2019	Population size = 50. Velocities are initialized randomly within [MIN_i,MAX_i] interval.
19	TAPSO	Triple-archive PSO	Xia et al., 2020	2020	Population size = 60. Velocities are initialized randomly within [MIN_i,MAX_i] interval, but during run are effectively restricted to ±0.2·[MIN_i-MAX_i].
20	PSO-sono	PSO for single-objective problems	Meng et al., 2022	2022	Population size = 100. Ring topology is used. Velocities are initialized randomly within [-30,30] interval.

such specialized variants are among DE or PSO families of methods, we also count the number of wins obtained by the particular algorithm on each set of problems. The algorithm that achieves the lowest 51-run averaged function value for the specific problem is termed a winner. Ties (within 10^{-8} precision) are counted as a win for each tied variant, hence there may be more than one winner for a specific problem. As a result, the number of wins summed over all problems in the particular set is often higher than the number of problems.

To compare DE and PSO families of methods, we use both approaches: averaged ranks and the number of wins for each of the nine competitions.

3.4. Statistical tests

In this study we perform a comparison between multiple algorithms on many problems. As a result, statistical tests designed for multiple comparisons between all tested algorithms are needed (Demsar, 2006; Garcia and Herrera, 2008; Derrac et al., 2011; Carrasco et al., 2020). We verified the statistical significance of the multiple comparisons among 20 algorithms on each set of problems by means of Friedman's test with the post-hoc Shaffer's static procedure at $\alpha = 0.05$ (Shaffer, 1986), which has been suggested as one among efficient methods for such task in the section 5 of Derrac et al. (2011). We have used the codes from Ulas et al. (2012). In the research we tried to follow the main guidelines on fair comparison presented in LaTorre et al. (2021).

4. Results and discussion

4.1. Mean rank-based comparison

In Fig. 1, separately for each of the nine competitions, all 20 algorithms are ordered from the best to the poorest one according to the average ranks. The information on the average rank is also given for each algorithm in Fig. 1. To facilitate reading, DE-based variants are given in blue, and PSO-based variants – in red. The detailed results are given in Supplementary Tables 1–9.

From Fig. 1 one immediately notes that for each set of problems, the first four places, out of 20, are always occupied by DE-based (blue) variants. Apart from CEC 2020_10 and CEC2020_15, the seven best variants are DE-based ones. The average ranks obtained by the best DE variants for different sets vary between 2.20 for CEC 2020_5 to 5.48 for real-world problems (CEC 2011), when the best PSO variants

order	avg_ranks	CEC_2011	avg_ranks	CEC_2014_10	avg_ranks	CEC_2014_50	avg_ranks	CEC_2017_10	avg_ranks	CEC_2017_50	avg_ranks	CEC_2020_5	avg_ranks	CEC_2020_10	avg_ranks	CEC_2020_15	avg_ranks	CEC_2020_20	order
1	5,48	MPADE	3,65	HARD_DE	3,50	HARD_DE	5,23	HARD_DE	2,20	L_SHADE_cnEpSin	4,20	HARD_DE	3,65	IDE	5,20	L_SHADE_cnEpSin	3,70	SADE	1
2	5,75	L_SHADE_cnEpSin	4,82	CIJADE	3,72	L_SHADE_cnEpSin	5,68	N_L_SHADE	2,47	HARD_DE	4,40	L_SHADE_cnEpSin	5,70	HARD_DE	5,60	HARD_DE	5,20	HARD_DE	2
3	5,93	HARD_DE	4,95	N_L_SHADE	6,08	ATPS_DE	6,17	CIJADE	5,75	N_L_SHADE	5,80	IDE	6,35	MPADE	6,10	IDE	5,30	L_SHADE_cnEpSin	3
4	5,98	CIJADE	5,18	L_SHADE_cnEpSin	6,52	CIJADE	6,23	L_SHADE_cnEpSin	6,07	MPADE	6,00	HMJCDE	6,75	SADE	7,20	N_L_SHADE	6,60	IDE	4
5	7,61	N_L_SHADE	6,10	IDE	7,02	HMJCDE	6,65	HMJCDE	6,28	IDE	6,65	CIJADE	6,90	CLPSO	7,80	CLPSO	7,30	N_L_SHADE	5
6	7,66	ATPS_DE	6,78	HMJCDE	7,08	N_L_SHADE	7,00	IDE	6,70	CIJADE	7,10	N_L_SHADE	7,75	N_L_SHADE	8,20	CIJADE	7,50	CIJADE	6
7	8,34	HMJCDE	8,02	SADE	7,33	IDE	7,40	SADE	7,37	ATPS_DE	7,50	SADE	7,90	HCLPSO	8,50	SADE	8,20	ATPS_DE	7
8	8,93	IDE	8,72	ATPS_DE	8,32	MPADE	7,43	MPADE	7,80	HMJCDE	8,15	CLPSO	8,10	L_SHADE_cnEpSin	8,55	HCLPSO	8,30	HCLPSO	8
9	9,11	SADE	9,80	MPADE	9,88	HCLPSO	8,72	ATPS_DE	9,67	DEPSO	8,35	HCLPSO	8,45	CIJADE	8,70	ATPS_DE	8,40	CLPSO	9
10	9,34	HCLPSO	10,10	HCLPSO	9,90	EPSO	8,72	HCLPSO	9,87	SADE	10,10	MPADE	8,65	HMJCDE	8,85	EPSO	8,90	EPSO	10
11	9,61	EPSO	11,10	EPSO	11,27	DEPSO	9,28	EPSO	10,70	EPSO	10,35	EPSO	8,70	EPSO	8,90	MPADE	9,00	HMJCDE	11
12	11,30	GLPSO	11,42	CLPSO	11,30	SADE	9,60	CLPSO	10,82	HCLPSO	11,15	DE	8,90	ATPS_DE	10,30	HMJCDE	10,10	MPADE	12
13	12,11	TAPSO	13,20	PSO_sono	11,58	CLPSO	12,65	DE	12,52	CLPSO	11,55	ATPS_DE	11,65	DE	11,90	DE	12,80	PSO_sono	13
14	12,20	CLPSO	13,30	DE	12,05	TAPSO	13,92	PSO_sono	13,47	GLPSO	12,15	IILPSO	12,95	PSO_sono	12,60	IILPSO	13,20	DE	14
15	14,48	PSO_sono	14,22	GLPSO	13,05	GLPSO	14,42	GLPSO	13,63	TAPSO	13,30	PSO_sono	14,30	IILPSO	13,65	PSO_sono	13,80	TAPSO	15
16	14,52	IILPSO	14,43	TAPSO	13,80	PSO_sono	14,55	IILPSO	14,47	PSO_sono	14,85	ALC_PSO	16,30	GLPSO	14,00	TAPSO	15,10	GLPSO	16
17	14,57	DEPSO	15,25	DEPSO	14,50	IILPSO	15,68	TAPSO	15,37	IILPSO	15,90	GLPSO	16,40	PSO	14,80	ALC_PSO	16,00	DEPSO	17
18	14,84	PSO	15,48	IILPSO	16,80	PSO	16,30	DEPSO	17,17	PSO	16,20	TAPSO	16,40	ALC_PSO	15,10	GLPSO	16,30	IILPSO	18
19	15,57	ALC_PSO	16,18	PSO	17,27	ALC_PSO	16,52	PSO	17,77	ALC_PSO	17,40	PSO	16,90	TAPSO	16,45	DEPSO	16,60	ALC_PSO	19
20	16,66	DE	17,30	ALC_PSO	19,03	DE	17,85	ALC_PSO	19,93	DE	18,90	DEPSO	17,30	DEPSO	17,60	PSO	17,70	PSO	20

Fig. 1. Averaged-ranks based ranking of PSO and DE algorithms on nine different competitions; each competition is composed of multiple problems. DE variants are given in blue, and PSO variants are given in red. Top algorithms are better, bottom algorithms are worse.

Table 2
Pair-wise comparison between algorithms – CEC 2011. 1 – differences are statistically significant at $\alpha = 0.05$; 0 – differences are not statistically significant.

	DE	SADE	ATPS-DE	IDE	MPA DE	HMJC DE	L-SHADE-cnEpSin	HARD -DE	CI-JADE	N-L-SHADE	PSO	CLPSO	ALC-PSO	HCL PSO	IILPSO	GLPSO	EPSO	DEPSO	TAPSO	PSO-sono
DE	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0	0	0
SADE	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
ATPS-DE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	1
IDE	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
MPADE	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	1	1	1
HMJCDE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
L-SHADE-cnEpSin	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
HARD-DE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	1
CIJADE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	1
N-L-SHADE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	1
PSO	0	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
CLPSO	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
ALC-PSO	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
HCLPSO	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IILPSO	0	0	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
GLPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EPSO	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DEPSO	0	0	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
TAPSO	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PSO-sono	0	0	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0

Table 3
Pair-wise comparison between algorithms – CEC 2014_10. 1 – differences are statistically significant at $\alpha = 0.05$; 0 – differences are not statistically significant.

	DE	SADE	ATPS-DE	IDE	MPADE	HMJCDE	L-SHADE-cnEpSin	HARD-DE	CI-JADE	N-L-SHADE	PSO	CLPSO	ALC-PSO	HCL PSO	IILPSO	GLPSO	EPSO	DEPSO	TAPSO	PSO-sono
DE	0	0	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
SADE	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	0
ATPS-DE	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	0
IDE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	1
MPADE	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0	1	0	0
HMJCDE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	1
L-SHADE-cnEpSin	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1	1
HARD-DE	1	0	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
CIJADE	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1	1
N-L-SHADE	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1	1
PSO	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0
CLPSO	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0
ALC-PSO	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0	1	0	0	0
HCLPSO	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0	0
IILPSO	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0
GLPSO	0	1	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
EPSO	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0
DEPSO	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
TAPSO	0	1	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
PSO-sono	0	0	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0

Table 4
Pair-wise comparison between algorithms – CEC 2014_50. 1 – differences are statistically significant at $\alpha = 0.05$; 0 – differences are not statistically significant.

	DE	SADE	ATPS-DE	IDE	MPADE	HMJCDE	L-SHADE-cnEpSin	HARD-DE	CI-JADE	N-L-SHADE	PSO	CLPSO	ALC-PSO	HCL PSO	IILPSO	GLPSO	EPSO	DEPSO	TAPSO	PSO-sono
DE	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	1	1	1	0
SADE	1	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0
ATPS-DE	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	1	1
IDE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	1
MPADE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1
HMJCDE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	1
L-SHADE-cnEpSin	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
HARD-DE	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
CIJADE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	1	1
N-L-SHADE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	1
PSO	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	0	0
CLPSO	1	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0
ALC-PSO	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	0	0
HCLPSO	1	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0
IILPSO	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
GLPSO	1	0	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
EPSO	1	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0
DEPSO	1	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0
TAPSO	1	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
PSO-sono	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0

Table 5
Pair-wise comparison between algorithms – CEC 2017_10. 1 – differences are statistically significant at $\alpha = 0.05$; 0 – differences are not statistically significant.

	DE	SADE	ATPS-DE	IDE	MPADE	HMJCDE	L-SHADE-cnEpSin	HARD-DE	CI-JADE	N-L-SHADE	PSO	CLPSO	ALC-PSO	HCL PSO	IILPSO	GLPSO	EPSO	DEPSO	TAPSO	PSO-sono
DE	0	0	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
SADE	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	1
ATPS-DE	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	0
IDE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	1
MPADE	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	1
HMJCDE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	1
L-SHADE-cnEpSin	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	1
HARD-DE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	1
CIJADE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	1
N-L-SHADE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	1
PSO	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0	1	0	0	0
CLPSO	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	0
ALC-PSO	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0	1	0	0	0
HCLPSO	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	0
IILPSO	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0
GLPSO	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0
EPSO	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	0
DEPSO	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0	1	0	0	0
TAPSO	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0	1	0	0	0
PSO-sono	0	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0

achieve average ranks between 6.90 (for CEC 2020_10) to 10.10 (for CEC 2014_10). The advantage of DE-based variants over PSO-based ones is massive for any set of problems, including real-world ones.

It is more difficult to point at the best DE-based variant. Among nine comparisons, four out of ten DE-based variants may reach the first place according to the average ranks. HARD-DE from 2019 performs best on four out of the nine competitions and is ranked the second best on four other comparisons. Only for real-world problems it is ranked 3rd best method (see Fig. 1). However, on real-world problems (CEC 2011) the difference in average ranks achieved by the four best DE-based variants is very low (between 5.48 and 5.98, the difference is not statistically significant), hence the results obtained by HARD-DE are among the best in each out of the nine competitions. L-SHADE-cnEpSin seems to be the second-best DE variant, with two wins and three second places.

According to Friedman's statistical test, the results obtained by various algorithms for each set of problems are significantly different. However, multiple comparisons between some algorithms are statistically significant, and between others are not, as is shown in Tables 2–10. The differences between DE and PSO algorithms are mainly statistically significant for real-world problems (CEC 2011), and benchmarks CEC 2014 and CEC 2017. The differences between algorithms observed for CEC 2020 problems are in most cases not statistically significant. If one focuses on real-world problems, it may be found that the differences between six DE variants: ATPS-DE, MPADE, L-SHADE-cnEpSin, HARD-DE, CIJADE and N-L-SHADE, and about 50% of PSO variants are statistically significant. On CEC 2014 and CEC 2017 benchmarks, the differences between eight DE variants (except DE and SADE) and at least 50% of PSO variants are statistically significant. Moreover, on 10- and 50-dimensional CEC 2014 and 50-dimensional CEC 2017 problems, the differences between the DE variant called HARD-DE and all ten PSO-based variants are statistically significant. The differences between HARD-DE and at least 50% of PSO variants are also statistically significant for 5-, 10-, and 20-dimensional versions of CEC 2020 problems. The statistical significance of differences between the L-SHADE-cnEpSin

Table 6
Pair-wise comparison between algorithms – CEC 2017_50. 1 – differences are statistically significant at $\alpha = 0.05$; 0 – differences are not statistically significant.

	DE	SADE	ATPS-DE	IDE	MPADE	HMJCDE	L-SHADE-cnEpSin	HARD-DE	CI-JADE	N-L-SHADE	PSO	CLPSO	ALC-PSO	HCL PSO	IILPSO	GLPSO	EPSO	DEPSO	TAPSO	PSO-sono
DE	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	1	1	1	1
SADE	1	0	0	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0
ATPS-DE	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	1	1
IDE	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	1	1
MPADE	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	1	1
HMJCDE	1	0	0	0	0	0	1	1	0	0	1	0	1	0	1	1	0	0	1	1
L-SHADE-cnEpSin	1	1	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1
HARD-DE	1	1	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1
CIJADE	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	1	1
N-L-SHADE	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	1	1
PSO	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	0	0
CLPSO	1	0	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
ALC-PSO	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	0	0
HCLPSO	1	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0
IILPSO	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0
GLPSO	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
EPSO	1	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0
DEPSO	1	0	0	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0
TAPSO	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
PSO-sono	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0

Table 7
Pair-wise comparison between algorithms – CEC 2020_5. 1 – differences are statistically significant at $\alpha = 0.05$; 0 – differences are not statistically significant.

	DE	SADE	ATPS-DE	IDE	MPADE	HMJCDE	L-SHADE-cnEpSin	HARD-DE	CI-JADE	N-L-SHADE	PSO	CLPSO	ALC-PSO	HCL PSO	IILPSO	GLPSO	EPSO	DEPSO	TAPSO	PSO-sono
DE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SADE	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
ATPS-DE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IDE	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0
MPADE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HMJCDE	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0
L-SHADE-cnEpSin	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0
HARD-DE	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0
CIJADE	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0
N-L-SHADE	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
PSO	0	1	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
CLPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
ALC-PSO	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
HCLPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
IILPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GLPSO	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
EPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DEPSO	0	1	0	1	0	1	1	1	1	1	0	1	0	1	0	0	0	0	0	0
TAPSO	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
PSO-sono	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

algorithm and PSO variants is almost as frequently confirmed as in the case of the HARD-DE variant.

None PSO-based variants tested are competitive with DE algorithms. However, among ten PSO variants HCLPSO from 2015 performs better than all the others in five out of the nine competitions, and is always among the first three PSO methods. One may also note that three PSO-based variants (HCLPSO, CLPSO, EPSO) are somehow competitive to some weaker DE-based variants on small sets of low-dimensional problems (i.e. CEC 2020_5, CEC 2020_10 and CEC 2020_15).

Interestingly, the only DE variant that consistently performs poorly is the initial DE version from 1997, which turns out the worst among 20 algorithms on CEC 2014_50 and CEC 2017_50 benchmarks, as well as on real-world problems (CEC 2011). This may be the result of inappropriate control parameter settings (that were based on the original choices, proposed for simple problems), which may be especially critical for higher dimensional problems. However, even this original DE variant performs on average better than the original PSO from 1998 in six out of ten competitions — which is in full agreement with older literature on comparison between the basic DE and PSO variants (Vesterstrøm and Thomson, 2004; Civicioglu and Besdok, 2013; Ezugwu et al., 2020). The similar, relatively poor performance of the two initial variants may currently be especially annoying as both initial versions are widely used in the literature to solve various practical problems. However, the lack of clear superiority of one initial version over another could initially benefit to the rapid development of both families of methods. Surprisingly, the advantage of more advanced DE-based variants over modern PSO-ones seems to be overlooked by both experts and practitioners.

4.2. Searching for the specialized algorithms

Some algorithms may perform especially well on selected problems, even though on average their performance may not be outstanding. Pointing at such methods may be useful for practitioners that are looking for an algorithm to solve their specific problem of interest. Hence, to present a different kind of comparison, we count the number

Table 8
Pair-wise comparison between algorithms – CEC 2020_10. 1 – differences are statistically significant at $\alpha = 0.05$; 0 – differences are not statistically significant.

	DE	SADE	ATPS-DE	IDE	MPADE	HMJCDE	L-SHADE-cnEpSin	HARD-DE	CI-JADE	N-L-SHADE	PSO	CLPSO	ALC-PSO	HCL PSO	IILPSO	GLPSO	EPSO	DEPSO	TAPSO	PSO-sono
DE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SADE	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0
ATPS-DE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IDE	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	0
MPADE	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0
HMJCDE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L-SHADE-cnEpSin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HARD-DE	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0
CIJADE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N-L-SHADE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
PSO	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
CLPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
ALC-PSO	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
HCLPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IILPSO	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GLPSO	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
EPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DEPSO	0	1	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0
TAPSO	0	1	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
PSO-sono	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 9
Pair-wise comparison between algorithms – CEC 2020_15. 1 – differences are statistically significant at $\alpha = 0.05$; 0 – differences are not statistically significant.

	DE	SADE	ATPS-DE	IDE	MPADE	HMJCDE	L-SHADE-cnEpSin	HARD-DE	CI-JADE	N-L-SHADE	PSO	CLPSO	ALC-PSO	HCL PSO	IILPSO	GLPSO	EPSO	DEPSO	TAPSO	PSO-sono
DE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SADE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ATPS-DE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IDE	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
MPADE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HMJCDE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L-SHADE-cnEpSin	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	0
HARD-DE	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
CIJADE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N-L-SHADE	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
PSO	0	0	0	1	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0
CLPSO	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
ALC-PSO	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
HCLPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IILPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GLPSO	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
EPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DEPSO	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
TAPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PSO-sono	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

of wins achieved by each algorithm on each set of problems (including ties; algorithms are tied winners if the difference between their 51-runs averaged performance on the specific problem is lower than 10^{-8}). Due to ties, the summarized number of wins is often larger than the number of problems. The number of wins achieved by DE and PSO-based variants on each of the nine competitions is illustrated in Fig. 2, and the detailed values are given in Table 11. In Table 11 we also count the total number of wins obtained by DE and by PSO-based variants across all benchmarks, and the total number of wins obtained by each specific algorithm on all considered problems in the nine competitions.

A brief look at Fig. 2 confirms that DE-based variants win much more problems than PSO-based ones. For any of the nine competitions DE-based variants note more than twice more wins than PSO-based variants (see Table 11).

The relative difference between the number of wins obtained by DE and PSO variants is the lowest for real-world problems (39 to 18, see Table 11). However, this is only due to the fact that almost all algorithms tie on real-world problems 3 and 8, which gives PSO-based variants the vast majority of their tied-wins. The only real-world problem on which the PSO-based variant outperformed all DE variants (no ties were noted) is problem 2, on which TAPSO turned out the best method. This is clearly seen in Fig. 3, in which the 51-runs averaged performances obtained by every algorithm are shown for all 22 real-world problems. In Fig. 3 the performance of DE-based variants is shown in the left sub-figures in blue, and the performance of PSO-based variants is given in the right sub-figures in red. As one may see, for the majority of problems either almost all (often apart from the basic DE) or some of the DE-based algorithms outperform PSO variants. There is also a group of problems (e.g. no. 5, 11.2, 11.10) on which the differences are small and hard to spot visually. However, PSO-based variants marginally dominate over DE-based ones only on a single problem 2. The visual inspection of the results given in Fig. 3 clearly confirms the superiority of DE algorithms over PSO ones, not only in rankings based on averaged statistics but also in practical differences easily observed for various real-world problems.

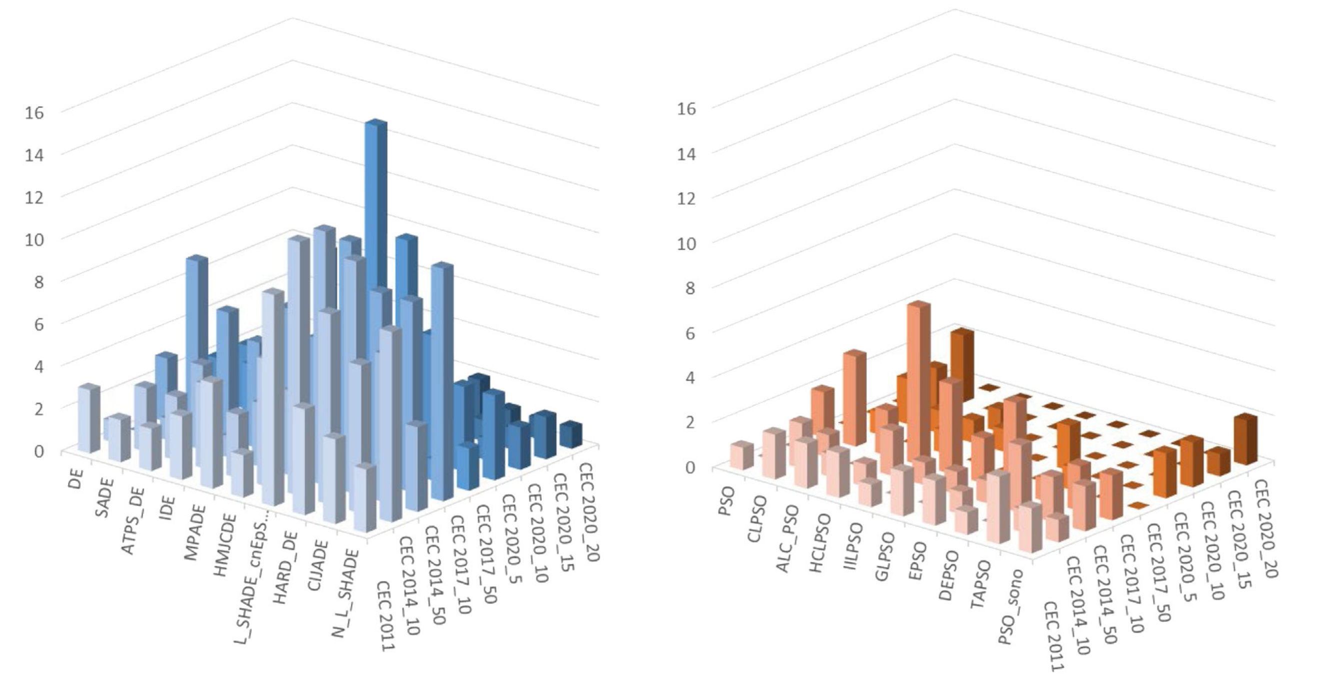

Fig. 2. The number of wins (best performances among all competing algorithms, including ties within the 10^{-8} threshold) achieved by ten DE and ten PSO-based variants in nine competitions. The number of wins is given in the vertical axis. DE-based variants are illustrated in blue, and PSO-based ones are in red. The scale is identical on both DE and PSO-based sub-figure.

Table 10
Pair-wise comparison between algorithms – CEC 2020_20. 1 – differences are statistically significant at $\alpha = 0.05$; 0 – differences are not statistically significant.

	DE	SADE	ATPS-DE	IDE	MPADE	HMJCDE	L-SHADE-cnEpSin	HARD-DE	CI-JADE	N-L-SHADE	PSO	CLPSO	ALC-PSO	HCL PSO	IILPSO	GLPSO	EPSO	DEPSO	TAPSO	PSO-sono
DE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SADE	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	0
ATPS-DE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IDE	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0
MPADE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HMJCDE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L-SHADE-cnEpSin	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	0	0
HARD-DE	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	0	0
CIJADE	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
N-L-SHADE	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
PSO	0	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
CLPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ALC-PSO	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
HCLPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IILPSO	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
GLPSO	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
EPSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DEPSO	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
TAPSO	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PSO-sono	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

On benchmarks based on artificially-created functions the difference between the number of wins achieved by DE and PSO-based variants is even larger than on real-world problems. DE wins on an order of magnitude more problems than PSO in the case of 50-dimensional CEC 2017 benchmarks (DE note 40 wins, PSO just 3) and on 20-dimensional CEC 2020 problems (18 to 2). Over all sets of problems, DE-based variants noted 339 wins, against 91 wins achieved by PSO-based algorithms. Although one may still find problems on which PSO-based variants outperform DE-based ones, as expected from No Free Lunch theorems for optimization (Wolpert and Macready, 1997), the overall advantage of DE-based algorithms is very clear also when one focuses on the number of wins, instead of the average ranks.

Considering each algorithm separately, from the last column of Table 11 one may note that L-SHADE-cnEpSin collects overall the largest number of 71 wins, followed by HARD-DE with 54 wins and N-L-SHADE with 38 wins. Readers could be surprised, as according to the average ranks HARD-DE seems to be a better method than L-SHADE-cnEpSin. This result points out that L-SHADE-cnEpSin is more specialized in some kinds of problems, and the HARD-DE algorithm is more flexible. The flexibility of HARD-DE may be also confirmed by the very uniform ranking in all nine competitions based on averaged ranks (which was ranked always between 1st and 3rd place, Fig. 1); variability of L-SHADE-cnEpSin ranking was much higher (ranging from 1st to 8th).

Table 11
The number of wins achieved by each algorithm on every set of problems.

	CEC 2011	CEC 2014_10	CEC 2014_50	CEC 2017_10	CEC 2017_50	CEC 2020_5	CEC 2020_10	CEC 2020_15	CEC 2020_20	**Summarized**
DE	3	1	0	3	0	2	1	1	1	**12**
SADE	2	3	1	8	0	3	1	2	6	**26**
ATPS-DE	2	3	4	6	3	2	1	1	1	**23**
IDE	3	4	1	5	2	3	4	3	3	**28**
MPADE	5	3	3	7	2	1	1	1	1	**24**
HMJCDE	2	6	4	6	3	4	1	1	1	**28**
L-SHADE-cnEpSin	10	12	12	11	16	5	1	2	2	**71**
HARD-DE	5	9	11	9	11	6	1	1	1	**54**
CIJADE	4	7	7	9	1	4	1	1	1	**35**
N-L-SHADE	3	9	4	11	2	4	2	2	1	**38**
PSO	1	0	0	2	0	0	0	0	0	**3**
CLPSO	2	2	1	4	1	2	2	3	0	**17**
ALC-PSO	2	1	0	2	0	1	0	0	0	**6**
HCLPSO	2	1	2	7	1	1	1	0	0	**15**
IILPSO	1	0	1	4	0	1	0	0	0	**7**
GLPSO	2	0	1	2	0	0	0	0	0	**5**
EPSO	2	1	1	4	0	2	0	0	0	**10**
DEPSO	1	0	3	1	1	0	0	0	0	**6**
TAPSO	3	1	2	2	0	0	0	0	0	**8**
PSO-sono	2	1	2	2	0	2	2	1	2	**14**
Summarized DE	**39**	**57**	**47**	**75**	**40**	**34**	**14**	**15**	**18**	**339**
Summarized PSO	**18**	**7**	**13**	**30**	**3**	**9**	**5**	**4**	**2**	**91**

The best PSO-based variants achieved much fever wins (CLPSO – 17, HCLPSO – 15, PSO-sono – 14, see Table 11). These numbers are much lower than the numbers of wins achieved by the poorest DE-based variants apart of the basic DE: the 9th best DE-based variant, ATP-DE, noted 23 wins. This again shows a clear superiority of DE-based algorithms over PSO-based ones.

5. Conclusions

Particle Swarm Optimization and Differential Evolution families of optimization algorithms were proposed in the mid-1990's, sprinkled into plentiful variants during the last quarter century, and found applications in almost every field of science or engineering. However, very few wider-scale comparisons between both families of methods were presented so far. In this paper we compare ten Differential Evolution variants against ten Particle Swarm Optimization ones in nine different competitions, including one composed of 22 real-world problems. The competing algorithms were selected from the whole history of PSO and DE development, with a larger focus on more recent variants. Tested algorithms were proposed in the major scientific journals or in proceedings of the main competitions among optimization algorithms. The comparison was based on the averaged ranking, and on the number of wins (problems on which a particular algorithm achieved the best performance) obtained for problems from the particular competition.

Irrespective of the way the algorithms were compared, DE-based variants clearly outperform PSO-based methods. According to the average ranks, in each of the nine competitions, the first four places were always occupied by DE-based variants. In some competitions, including the one composed of 22 real-world problems, nine DE-based variants (apart from the initial DE version from 1997) performed better than all ten PSO-based algorithms. The differences between the best DE variants and a large part of PSO algorithms are statistically significant.

In each of the nine competitions the number of wins achieved by DE-based variants was more than twice higher than the number of wins achieved by PSO-based variants. In two competitions DE-based variants noted about ten times more wins than PSO-based algorithms. Problems on which PSO-based algorithms perform better than DE-based ones do exist but are relatively rare.

The discussed findings are obviously restricted by the choice of the benchmark problems, competing algorithms and their control parameters, and the general limitations of empirical comparison exemplified by the No Free Lunch theorems (Wolpert and Macready, 1997). Nonetheless, the wide-scale superiority of DE variants over PSO ones on widely used benchmarks and real-world problems is highly appealing.

In the scientific literature focused on applications of Swarm Intelligence or Evolutionary Algorithms, PSO methods are 2–3 times more popular than DE-based ones. According to the experimental comparison presented in this study, we recommend that practitioners pay more attention to DE-based algorithms when seeking the optimizer for their problem of interest. We also advise the Swarm Intelligence community to search for such modifications that would make PSO variants more competitive to DE-based ones. It seems that the major paths could be related to control parameters adaptation, population size dynamic, the topology of the swarm, and communication systems between particles. The search for new paths of PSO development seems necessary, as the current study experimentally shows a clear advantage of DE-based algorithms over PSO-based ones.

CRediT authorship contribution statement

Adam P. Piotrowski: Designed the research, Performed research, Analyzed data and results, Wrote the paper. **Jaroslaw J. Napiorkowski:** Analyzed data and results, Wrote the paper. **Agnieszka E. Piotrowska:** Designed the research, Wrote the paper.

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.

Data availability

The authors do not have permission to share data.

Fig. 3. (2 pages long). The 51-runs average performances obtained by every algorithm on real-world CEC 2011 problems. Each algorithm is marked by the number associated with it in Table 1. Algorithms numbered 1–10 are DE-based ones (given in blue), and algorithms numbered 11–20 are PSO-based ones (given in red). On the vertical axis the value of the objective function is given (the lower – the better). The scale is identical for DE and PSO-based algorithms on the specific problem, but vary between problems.

Acknowledgment

This work was supported within statutory activities No 3841/E-41/S/2022 of the Ministry of Science and Higher Education of Poland.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.engappai.2023.106008.

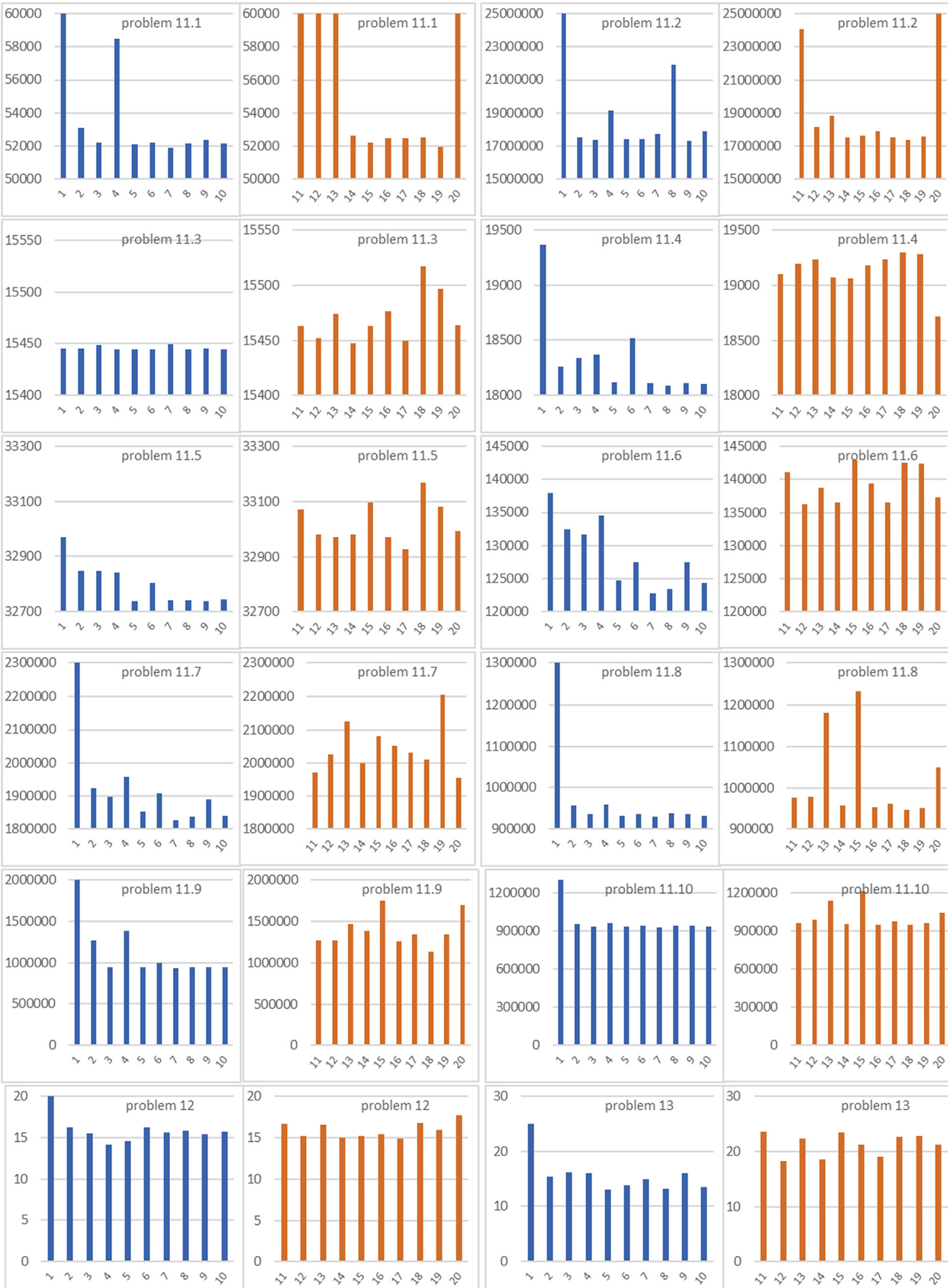

Fig. 3. (*continued*).

References

Abualigah, L.M., Khader, A.T., 2017. Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. J. Supercomput. 73 (11), 4773–4795.

Ahmad, M.F., Isa, N.A.M., Lim, W.H., Ang, K.M., 2022. Differential evolution: A recent review based on state-of-the-art works. Alex. Eng. J. 61 (5), 3831–3872.

Al-Dabbagh, R.D., Neri, F., Idris, D., Baba, M.S., 2018. Algorithmic design issues in adaptive differential evolution schemes: Review and taxonomy. Swarm Evol. Comput. 43, 284–311.

Amaral, L., Castro, R., 2017. Offshore wind farm layout optimization regarding wake effects and electrical losses. Eng. Appl. Artif. Intell. 60, 26–34.

Ani, Z.A., Gujarathi, A.M., Al-Muhtaseb, A.H., 2023. A state of art review on applications of multi-objective evolutionary algorithms in chemicals production reactors. Artif. Intell. Rev. 56, 2435–2496.

Awad, N.H., Ali, M.Z., Liang, J.J., Qu, B.Y., Suganthan, P.N., 2016. Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. Technical Report, Nan-yang Technological University, Singapore.

Awad, N.H., Ali, M.Z., Suganthan, P.N., 2017b. Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving CEC2017 benchmark problems. In: 2017 IEEE Congress on Evolutionary Computation. CEC, http://dx.doi.org/10.1109/CEC.2017.7969336.

Awad, N.H., Ali, M.Z., Suganthan, P.N., Reynolds, R.G., 2017a. CADE: A hybridization of cultural algorithm and differential evolution for numerical optimization. Inform. Sci. 378, 215–241.

Bilal, Pant, M., Zaheer, H., Garcia-Hernandez, L., Abraham, A., 2020. Differential evolution: A review of more than two decades of research. Eng. Appl. Artif. Intell. 90, 103479.

Biswas, P.P., Suganthan, P.N., Mallipeddi, R., Amaratunga, G.A.J., 2018. Optimal power flow solutions using differential evolution algorithm integrated with effective constraint handling techniques. Eng. Appl. Artif. Intell. 68, 81–100.

Bonyadi, M.R., Michalewicz, Z., 2017a. Particle swarm optimization for single objective continuous space problems: A review. Evol. Comput. 25 (1), 1–54.

Bonyadi, M.R., Michalewicz, Z., 2017b. Impacts of coefficients on movement patterns in the particle swarm optimization algorithm. IEEE Trans. Evol. Comput. 21 (3), 378–390.

Bouaziz, F., Oulhadj, H., Boutana, D., Siarry, P., 2019. Automatic ECG arrhythmias classification scheme based on the conjoint use of the multi-layer perceptron neural network and a new improved metaheuristic approach. IET Signal Process. 13 (8), 726–735.

Boussaid, I., Lepagnot, J., Siarry, P., 2013. A survey on optimization metaheuristics. Inform. Sci. 237, 82–117.

Bujok, P., Kolenovsky, P., 2022. Eigen crossover in cooperative model of evolutionary algorithms applied to CEC 2022 single objective numerical optimisation. In: Proceedings of the 2022 IEEE Congress on Evolutionary Computation. CEC, 18-23 2022, Padua, Italy, http://dx.doi.org/10.1109/CEC55065.2022.9870433.

Bujok, P., Tvrdik, J., Polakova, R., 2019. Comparison of nature-inspired population-based algorithms on continuous optimiza-tion problems. Swarm Evol. Comput. 50, 100490.

Camacho-Villalon, C.L., Dorigo, M., Stutzle, T., 2022. PSO-X : A component-based framework for the automatic design of particle swarm optimization algorithms. IEEE Trans. Evol. Comput. 26 (3), 402–416.

Caraffini, F., Kononova, A.V., Corne, D., 2019. Infeasibility and structural bias in differential evolution. Inform. Sci. 496, 161–179.

Caraffini, F., Neri, F., 2019. A study on rotation invariance in differential evolution. Swarm Evol. Comput. 50, 100436.

Carrasco, J., García, S., Rueda, M.M., Das, S., Herrera, F., 2020. Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review. Swarm Evol. Comput. 54, 100665.

Chatterjee, A., Siarry, P., 2006. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. Comput. Oper. Res. 33 (3), 859–871.

Chen, W.N., Zhang, J., Lin, Y., Chen, N., Zhan, Z.H., Chung, H.S.H., Li, Y., Shi, Y.H., 2013. Particle swarm optimization with an aging leader and challengers. IEEE Trans. Evol. Comput. 17 (2), 241–258.

Cheng, J., Zhang, G., Neri, F., 2013. Enhancing distributed differential evolution with multicultural migration for global numerical optimization. Inform. Sci. 247, 72–93.

Cipriani, C., Huang, H., Qiu, J., 2022. Zero-inertia limit: From particle swarm optimization to consensus-based optimization. SIAM J. Math. Anal. 54 (3), http://dx.doi.org/10.1137/21M1412323.

Civicioglu, P., Besdok, E., 2013. A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. Artif. Intell. Rev. 39, 315–346.

Cleghorn, C.W., Engelbrecht, A.P., 2018. Particle swarm stability: A theoretical extension using the non-stagnate distribution assumption. Swarm Intell. 12, 1–22.

Cleghorn, C.W., Stapleberg, B., 2022. Particle swarm optimization: Stability analysis using informers under arbitrary coefficient distributions. Swarm Evol. Comput. 71, 101060.

Clerc, M., Kennedy, J., 2002. The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. 6 (1), 58–73.

Cui, L.Z., Li, G.H., Lin, Q.Z., Chen, J.Y., Lu, N., 2016. Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations. Comput. Oper. Res. 67, 155–173.

Darwish, A., Hassanien, A.E., Das, S., 2020. A survey of swarm and evolutionary computing approaches for deep learning. Artif. Intell. Rev. 53, 1767–1812.

Das, S., Abraham, A., Konar, A., 2008. Particle swarm optimization and differential evolution algorithms: Technical analysis, applications and hybridization perspectives. In: Advances of Computational Intelligence in Industrial Systems. Studies in Computational Intelligence, Vol. 116. SCI, pp. 1–38.

Das, S., Mullick, S.S., Suganthan, P.N., 2016. Recent advances in differential evolution – An updated survey. Swarm Evol. Comput. 27, 1–30.

Das, S., Suganthan, P.N., 2010. Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolution-Ary Algorithms on Real World Optimization Problems. Jadavpur Univ. Nanyang Technol. Univ., Kolkata, India.

Demsar, J., 2006. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30.

Derrac, J., Garcia, S., Molina, D., Herrera, F., 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol. Comput. 1, 3–18.

Dhanalakshmy, D.M., Jeyakumar, G., Velayutham, C.S., 2022. Analytical study on the role of scale factor parameter of differential evolution algorithm on its convergence nature. In: Differential Evolution: From Theory to Practice. Springer, Germany, pp. 77–135.

Dorronsoro, B., Bouvry, P., 2011. Improving classical and decentralized differential evolution with new mutation operator and population topologies. IEEE Trans. Evol. Comput. 15 (1), 67–98.

Eberhart, R.C., Shi, Y., 2000. Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00, la Jolla, CA, USA, http://dx.doi.org/10.1109/CEC.2000.870279.

Eiben, A.E., Smith, J., 2015. From evolutionary computation to the evolution of things. Nature 521, 476–482.

El-Hussieny, H., Abouelsoud, A.A., Assal, S.F.M., Megahed, S.M., 2016. Adaptive learning of human motor behaviors: An evolving inverse optimal control approach. Eng. Appl. Artif. Intell. 50, 115–124.

Elbes, M., Alzubi, S., Kanan, T., Al.-Fuqaha, A., Hawashin, B., 2019. A survey on particle swarm optimization with emphasis on engineering and network applications. Evol. Intell. 12, 113–129.

Ezugwu, A.E., Adeleke, O.J., Akinyelu, A.A., Viriri, S., 2020. A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems. Neural Comput. Appl. 32, 6207–6251.

Fan, J.X., Shen, S.Z., Erwin, D.H., Sadler, P.M., MacLeod, N., Cheng, Q.M., Hou, X.D., Yang, J., Wang, X.D., Wang, Y., Zhang, H., Chen, X., Li, G.X., Zhang, Y.C., Shi, Y.K., Yuan, D.X., Chen, Q., Zhang, L.N., Li, C., Zhao, Y.Y., 2020. A high-resolution summary of cambrian to early triassic marine invertebrate biodiversity. Science 367, 272–277.

Flori, A., Oulhadj, H., Siarry, P., 2022. Quantum particle swarm optimization: An auto-adaptive PSO for local and global optimization. Comput. Optim. Appl. 82, 525–559.

Fuqing, Z., Feilong, X., Yi, Z., Weimin, M., Chuck, Z., Houbin, S., 2018. A hybrid algorithm based on self-adaptive gravitational search algorithm and differential evolution. Expert Syst. Appl. 113, 515–530.

Garcia, S., Herrera, F., 2008. An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. J. Mach. Learn. Res. 9, 2677–2694.

Ghosh, A., Das, S., Das, A.K., Senkerik, R., Viktorin, A., Zelinka, I., Masegosa, A.D., 2022. Using spatial neighborhoods for parameter adaptation: An improved success history based differential evolution. Swarm Evol. Comput. 71, 101057.

Gong, Y.J., Li, J.J., Zhou, Y., Li, Y., Chung, H.S.H., Shi, Y.H., Zhang, J., 2016. Genetic learning particle swarm optimization. IEEE Trans. Cybern. 46 (10), 2277–2290.

Grimm, S.L., Demory, B.O., Gillon, M., Dorn, C., Agol, E., Burdanov, A., Delrez, L., Sestovic, M., Triaud, A.H.M.J., Turbet, M., Bolmont, E., Caldas, A., Wit, J.de., Jehin, E., Leconte, J., Raymond, S.N., Grootel, V.van., Burgasser, A.J., Carey, S., Fabrycky, D., Heng, K., Hernandez, D.M., Ingalls, J.G., Lederer, S., Selsis, F., Queloz, D., 2018. The nature of the TRAPPIST-1 exoplanets. Astron. Astrophys. 613 (A68).

Hammouche, G., Diaf, M., Siarry, P., 2010. A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem. Eng. Appl. Artif. Intell. 23 (5), 676–688.

Harrison, K.R., Engelbrecht, A.P., Ombuki-Berman, B.M., 2018. Self-adaptive particle swarm optimization: A review and analysis of convergence. Swarm Intell. 12, 187–226.

Houssein, E.H., Rezk, H., Fathy, A., Mahdy, M.A., Nassef, A.M., 2022. A modified adaptive guided differential evolution algorithm applied to engineering applications. Eng. Appl. Artif. Intell. 113, 104920.

Hu, Z., Su, Q., Yang, X., Xiong, Z., 2016. Not guaranteeing convergence of differential evolution on a class of multimodal functions. Appl. Soft Comput. 41, 479–487.

Huang, H., Qiu, J., Riedl, K., 2022. On the global convergence of particle swarm optimization methods. arXiv: arXiv:2201.12460v2.

Iacca, G., Caraffini, F., Neri, F., 2014. Multi-strategy coevolving aging particle optimization. Int. J. Neural Syst. 24 (1), 1450008.

Ilonen, J., Kamarainen, J.K., Lampinen, J., 2003. Differential evolution training algorithm for feed-forward neural networks. Neural Process. Lett. 17, 93–105.
Javidrad, F., Nazari, M., 2017. A new hybrid particle swarm and simulated annealing stochastic optimization method. Appl. Soft Comput. 60, 634–654.
Jia, Y., Qu, L., Li, X., 2022. A double-layer coding model with a rotation-based particle swarm algorithm for unmanned combat aerial vehicle path planning. Eng. Appl. Artif. Intell. 116, 105410.
Jontof-Hutter, D., Rowe, J.F., Lissauer, J.J., Fabrycky, D.C., Ford, E.B., 2015. The mass of the Mars-sized exoplanet Kepler-138 b from transit timing. Nature 522, 321–323.
Kao, Y.T., Zahara, E., 2008. A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Appl. Soft Comput. 8 (2), 849–857.
Kar, D., Ghosh, M., Guha, R., Sarkar, R., Garcia-Hernandez, L., Abraham, A., 2020. Fuzzy mutation embedded hybrids of gravitational search and particle swarm optimization methods for engineering design problems. Eng. Appl. Artif. Intell. 95, 103847.
Kennedy, J., 1999. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99. Washington, DC, USA, http://dx.doi.org/10.1109/CEC.1999.785509.
Kennedy, J., Eberhart, R.C., 1995. Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks. Perth, Australia., IEEE, Piscataway, NJ, USA, pp. IV 1942–1948, 1995.
Khan, T.A., Ling, S.H., 2021. A novel hybrid gravitational search particle swarm optimization algorithm. Eng. Appl. Artif. Intell. 102, 104263.
Kononova, A.V., Caraffini, F., Back, T., 2021. Differential evolution outside the box. Inform. Sci. 581, 587–604.
Kumar, N., Shaikh, A.A., Mahato, S.K., Bhunia, A.K., 2021. Applications of new hybrid algorithm based on advanced Cuckoo search and adaptive Gaussian quantum behaved particle swarm optimization in solving ordinary differential equations. Expert Syst. Appl. 172, 114646.
Larsen, A.H., Mortensen, J.J., Blomqvist, J., Castelli, I.E., Christensen, R., Dulak, M., Friis, J., Groves, M.N., Hammer, B., Hargus, C., Hermes, E.D., Jennings, P.C., Jensen, P.B., Kermode, J., Kitchin, J.R., Kolsbjerg, E.L., Kubal, J., Lysgaard, S., Maronsson, J.B., Maxson, T., Olsen, T., Pastewka, L., Peterson, A., Rostgaard, C., Schiøtz, J., Schutt, O., Strange, M., Thygesen, K., Vegge, T., Vilhelmsen, L., Walter, M., Zeng, Z., Jacobsen, K.W., 2017. The atomic simulation environment — A python library for working with atoms. J. Physics – Condens. Matter 29 (27), 273002.
LaTorre, A., Molina, D., Osaba, E., Poyatos, J., DelSer, J., Herrera, F., 2021. A prescription of methodological guidelines for com-paring bio-inspired optimization algorithms. Swarm Evol. Comput. 67, 100973.
Leboucher, C., Shin, H.S., Siarry, P., Menec, S.Le., Chelouah, R., Tsourdos, A., 2016. Convergence proof of an enhanced particle swarm optimisation method integrated with evolutionary game theory. Inf. Sci. 346–347, 389–411.
Li, G., Lin, Q., Cui, L., Du, Z., Liang, Z., Chen, J., Lu, N., Ming, Z., 2016. A novel hybrid differential evolution algorithm with modified CoDE and JADE. Appl. Soft Comput. 47, 577–599.
Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S., 2006. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans. Evol. Comput. 10 (3), 281–295.
Liang, J.J., Qu, B.Y., Suganthan, P.N., 2013. Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China, and Nanyang Technological University, Singapore.
Liao, T., Molina, D., Stutzle, T., 2015. Performance evaluation of automatically tuned continuous optimizers on different benchmark sets. Appl. Soft Comput. 27, 490–503.
Lim, S.P., Haron, H., 2013. Performance comparison of genetic algorithm, differential evolution and particle swarm optimi-zation towards benchmark functions. In: Proc. of the IEEE Conference on Open Systems. pp. 41–46.
Lim, W.H., Isa, N.A.M., 2014. Particle swarm optimization with increasing topology connectivity. Eng. Appl. Artif. Intell. 27, 80–102.
Lin, C., Qing, A., Quanyuan, F., 2011. A comparative study of crossover in differential evolution. J. Heuristics 17, 675–703.
Liu, H.R., Cui, J.C., Lu, Z.D., Liu, D.Y., Deng, Y.J., 2019. A hierarchical simple particle swarm optimization with mean dimensional information. Appl. Soft Comput. 76, 712–725.
Locatelli, M., Vasile, M., 2015. (Non) convergence results for the differential evolution method. Optim. Lett. 9, 413–425.
Lv, J.A., Wang, Y.C., Zhu, L., Ma, Y.M., 2011. Predicted novel high-pressure phases of lithium. Phys. Rev. Lett. 106 (1), 015503.
Lynn, N., Ali, M.Z., Suganthan, P.N., 2018. Population topologies for particle swarm optimization and differential evolution. Swarm Evol. Comput. 39, 24–35.
Lynn, N., Suganthan, P.N., 2015. Heterogeneous comprehensive learning particle swarm optimization with enhanced explora-tion and exploitation. Swarm Evol. Comput. 24, 11–24.
Lynn, N., Suganthan, P.N., 2017. Ensemble particle swarm optimizer. Appl. Soft Comput. 55, 533–548.
Mallipeddi, R., Suganthan, P.N., 2008. Empirical study on the effect of population size on differential evolution algorithm. In: Proceedings of IEEE Congress on Evolutionary Computation. HongKong, 2008.
Marini, F., Walczak, B., 2015. Particle swarm optimization (PSO). A tutorial. Chemometr. Intell. Lab. Syst. 149 (Part B), 153–165.
Meng, Z., Pan, J.S., 2019. HARD-DE: Hierarchical archive based mutation strategy with depth information of evolution for the enhancement of differential evolution on numerical optimization. IEEE Access 7, 12832–12854.
Meng, Z., Yang, C., 2022. Two-stage differential evolution with novel parameter control. Inform. Sci. 596, 321–342.
Meng, Z., Zhong, Y., Mao, G., Liang, Y., 2022. PSO-Sono: A novel PSO variant for single-objective numerical optimization. Inform. Sci. 586, 176–191.
Meng, Z., Zhong, Y., Yang, C., 2021. CS-DE: Cooperative strategy based differential evolution with population diversity enhancement. Inform. Sci. 577, 663–696.
Michalewicz, Z., 2012. Quo vadis, evolutionary computation?. In: Liu, J., Alippi, C., Bouchon-Meunier, B., Greenwood, G.W., Abbass, H.A. (Eds.), Advances in Computational Intelligence: IEEE World Congress on Computational Intelligence. WCCI 2012, Brisbane, Australia, Springer, Berlin Heidelberg, Berlin, Heidelberg, pp. 98–121, 2012.
Mininno, E., Neri, F., Cupertino, F., Naso, D., 2010. Compact differential evolution. IEEE Trans. Evol. Comput. 15 (1), 32–54.
Mohamed, A.W., Hadi, A.A., Mohamed, A.K., 2021. Differential evolution mutations: Taxonomy, comparison and convergence analysis. IEEE Access 9, 68629–68662.
Neggaz, N., Benyettou, A., 2009. Hybrid models based on biological approaches for speech recognition. Artif. Intell. Rev. 32 (1–4), 45–57.
Neri, F., Mininno, E., Iacca, G., 2013. Compact particle swarm optimization. Inform. Sci. 239, 96–121.
Neri, F., Tirronen, V., 2010. Recent advances in differential evolution: A survey and experimental analysis. Artif. Intell. Rev. 33, 61–106.
Nowak-Sliwinska, P., Weiss, A., Ding, X., Dyson, P.J., van Den Bergh, H., Griffioen, A.W., Ho, C.M., 2016. Optimization of drug combinations using feedback system control. Nat. Protoc. 11 (2), 302–315.
Opara, K.R., Arabas, J., 2019. Differential evolution: A survey of theoretical analyses. Swarm Evol. Comput. 44, 546–558.
Pan, J.S., Liu, N., Chu, S.C., 2020. A hybrid differential evolution algorithm and its application in unmanned combat aerial vehicle path planning. IEEE Access 8, 17691–17712.
Parouha, R.P., Verma, P., 2021. Design and applications of an advanced hybrid meta-heuristic algorithm for optimization problems. Artif. Intell. Rev. 54, 5931–6010.
Parouha, R.P., Verma, P., 2022. A systematic overview of developments in differential evolution and particle swarm optimization with their advanced suggestion. Appl. Intell. 52, 10448–10492.
Pattanaik, J.K., Basu, M., Dash, D.P., 2019. Dynamic economic dispatch: A comparative study for differential evolution, particle swarm optimization, evolutionary programming, genetic algorithm, and simulated annealing. J. Electr. Syst. Inf. Technol. 6, 1–18.
Piotrowski, A.P., 2017. Review of differential evolution population size. Swarm Evol. Comput. 32, 1–24.
Piotrowski, A.P., Napiorkowski, J.J., 2018. Some metaheuristics should be simplified. Inform. Sci. 427, 32–62.
Piotrowski, A.P., Napiorkowski, M.J., Napiorkowski, J.J., Rowinski, R.M., 2017. Swarm intelligence and evolutionary algorithms: Performance versus speed. Inform. Sci. 384, 34–85.
Piotrowski, A.P., Napiorkowski, J.J., Piotrowska, A.E., 2020. Population size in particle swarm optimization. Swarm Evol. Comput. 58, 100718.
Poli, R., Kennedy, J., Blackwell, T., 2007. Particle swarm optimization. Swarm Intell. 1, 33–57.
Potthuri, S., Shankar, T., Rajesh, A., 2018. Lifetime improvement in wireless sensor networks using hybrid differential evolution and simulated annealing (DESA). Ain Shams Eng. J. 9 (4), 655–663.
Price, K.V., Awad, N.H., Ali, M.Z., Suganthan, P.N., 2019. The 2019 100-Digit Challenge on Real-Parameter, Single-Objective Optimization: Analysis of Results. Tech. Rep., Nanyang Technological University, Singapore, http://www.ntu.edu.sg/home/epnsugan.
Qin, Q., Cheng, S., Zhang, Q.Y., Li, L., Shi, Y.H., 2016. Particle swarm optimization with interswarm interactive learning strategy. IEEE Trans. Cybern. 46 (10), 2238–2251.
Qin, A.K., Huang, V.L., Suganthan, P.N., 2009. Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. Evol. Comput. 13 (2), 398–417.
Rana, S., Jasola, S., Kumar, R., 2011. A review on particle swarm optimization algorithms and their applications to data clustering. Artif. Intell. Rev. 35, 211–222.
Redoloza, F., Li, L., 2021. A comparison of extremal optimization, differential evolution and particle swarm optimization methods for well placement design in groundwater management. Math. Geosci. 53, 711–735.
Segura, C., Coello Coello, C.A., Segredo, E., Leon, C., 2015. On the adaptation of the mutation scale factor in differential evolution. Optim. Lett. 9, 189–198.
Sengupta, S., Basak, S., Peters, II, .R.A., 2019. Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives. Mach. Learn. Knowl. Extract. 1 (1), 157–191.

Shaffer, J., 1986. Modified sequentially rejective multiple test procedures. J. Amer. Statist. Assoc. 81, 826–831.
Shami, T.M., El-Saleh, A.A., Alswaiti, M., Al-Tashi, Q., Summakieh, M.A., Mirjalili, S., 2022. Particle swarm optimization: A comprehensive survey. IEEE Access 10, 10031–10061.
Shi, Y., Eberhart, R.C., 1998. A modified particle swarm optimizer. In: Proceeding in IEEE Congress on Evolutionary Computation. CEC, pp. 69–73.
Shir, O.M., Israeli, A., Caftory, A., Zepko, G., Bloch, I., 2022. Algorithmically-guided discovery of viral epitopes via linguistic parsing: Problem formulation and solving by soft computing. Appl. Soft Comput. 129, 109509.
Shukla, A.K., Janmaijaya, M., Abraham, A., Muhuri, P.K., 2019. Engineering applications of artificial intelligence: A bibliometric analysis of 30 years (1988–2018). Eng. Appl. Artif. Intell. 85, 517–532.
Skvorc, U., Eftimov, T., Korsec, P., 2019. CEC real-parameter optimization competitions: Progress from 2013 to 2018. In: IEEE Congress on Evolutionary Computation. CEC, Wellington, New Zealand, http://dx.doi.org/10.1109/CEC.2019.8790158.
Sorensen, K., Sevaux, M., Glover, F., 2018. A history of metaheuristics. In: Martí, R., Panos, P., Resende, M. (Eds.), Handbook of Heuristics. Springer, Cham, Switzerland, pp. 1–18.
Stanonov, V., Akhmedova, S., Semenkin, E., 2021. Visualizing crossover rate influence in differential evolution with expected fitness improvement. In: META 2021: Metaheuristics and Nature Inspired Computing. Springer, pp. 106–123.
Stanonov, V., Akhmedova, S., Semenkin, E., 2022. The automatic design of parameter adaptation techniques for differential evolution with genetic programming. Knowl.-Based Syst. 239, 108070.
Storn, R., Price, K.V., 1995. Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical Report TR-95-012, International Computer Sciences Institute, Berkeley, California, USA.
Storn, R., Price, K.V., 1997. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11 (4), 341–359.
Su, Q., Cai, G., Hu, Z., Yang, X., 2022. Test case generation using improved differential evolution algorithms with novel hypercube-based learning strategies. Eng. Appl. Artif. Intell. 112, 104840.
Sun, G., Xu, G., Jiang, N., 2020. A simple differential evolution with time-varying strategy for continuous optimization. Soft Comput. 24, 2727–2747.
Sun, Y., Zhang, L., Gu, X., 2012. A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems. Neurocomputing 98, 76–89.
Tanabe, R., Fukunaga, A., 2014. Improving the search performance of SHADE using linear population size reduction. In: Proc. IEEE Congress on Evolutionary Computation. Bejing, China, pp. 1658–1665.
Tang, L., Dong, Y., Liu, J., 2015. Differential evolution with an individual-dependent mechanism. IEEE Trans. Evol. Comput. 19 (4), 560–574.
Tang, J., Duan, H., Lao, S., 2023. Swarm intelligence algorithms for multiple unmanned aerial vehicles collaboration: A comprehensive review. Artif. Intell. Rev. http://dx.doi.org/10.1007/s10462-022-10281-7, (in press).
Trivedi, A., Srinivasan, D., Biswas, S., Reindl, T., 2015. Hybridizing genetic algorithm with differential evolution for solving the unit commitment scheduling problem. Swarm Evol. Comput. 23, 50–64.
Ulas, A., Yildiz, O.T., Alpaydin, E., 2012. Cost-conscious comparison of supervised learning algorithms over multiple data sets. Pattern Recognit. 45, 1772–1781.
van der Berg, F., Engelbrecht, A.P., 2001. Effects of swarm size on cooperative particle swarm optimisers. In: GECCO'01: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. pp. 892–899.
van der Berg, F., Engelbrecht, A.P., 2010. A convergence proof for the particle swarm optimiser. Fund. Inform. 105 (4), 341–374.
Varelas, K., El Hara, O.A., Brockhoff, D., Hansen, N., Nguyen, D.M., Tusar, T., Auger, A., 2020. Benchmarking large-scale continu-ous optimizers: The BBOB-largescale testbed, a COCO software guide and beyond. Appl. Soft Comput. 97, 106737.
Vesterstrøm, J., Thomson, R.A., 2004. Comparative study of differential evolution, particle swarm optimization, and evolution-ary algorithms on numerical benchmark problems. In: Proc. IEEE Congr. Evol. Comput.. pp. 1980–1987.
Viktorin, A., Senkerik, R., Pluhacek, M., Kadavy, T., Zamuda, A., 2019. Distance based parameter adaptation for success-history based differential evolution. Swarm Evol. Comput. 50, 100462.
Wang, Y., Li, H.X., Huang, T., Li, L., 2014. Differential evolution based on covariance matrix learning and bimodal distribution parameter setting. Appl. Soft Comput. 18, 232–247.
Wang, Y., Wang, L., Ye, L., Ma, X., Wu, W., Yang, Z., He, X., Zhang, L., Zhang, Y., Zhou, Y., Luo, Y., 2022. A novel self-adaptive fractional multivariable grey model and its application in forecasting energy production and conversion of China. Eng. Appl. Artif. Intell. 115, 105319.
Weber, M., Neri, F., Tirronen, V., 2011. A study on scale factor in distributed differential evolution. Inform. Sci. 181 (12), 2488–2511.
Weber, M., Neri, F., Tirronen, V., 2013. A study on scale factor/crossover interaction in distributed differential evolution. Artif. Intell. Rev. 39, 195–224.
Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1 (1), 67–82.
Wu, G., Mallipeddi, R., Suganthan, P.N., 2019. Ensemble strategies for population-based optimization algorithms – A survey. Swarm Evol. Comput. 44, 695–711.
Xia, X., Gui, L., Yu, F., Wu, H., Wei, B., Zhang, Y.L., Zhan, Z.H., 2020. Triple archives particle swarm optimization. IEEE Trans. Cybern. 50 (12), 4862–4875.
Xia, Y., Yi, W., Zhang, D., 2022. Coupled extreme learning machine and particle swarm optimization variant for projectile aerodynamic identification. Eng. Appl. Artif. Intell. 114, 105100.
Xin, B., Chen, J., Zhang, J., Fang, H., Peng, Z.H., 2012. Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and Taxonomy. IEEE Trans. Syst. Man Cybern. C: Appl. Rev. 42 (5), 744–767.
Xue, Y., Tong, Y., Neri, F., 2022. An ensemble of differential evolution and adam for training feed-forward neural networks. Inform. Sci. 608, 453–471.
Xue, Y., Zhang, Q., Neri, F., 2021. Self-adaptive particle swarm optimization-based echo state network for time series prediction. Int. J. Neural Syst. 31 (12), 2150057.
Yue, C.T., Price, K.V., Suganthan, P.N., Liang, J.J., Ali, M.Z., Qu, B.Y., Awad, N.H., Biswas, P.P., 2019. Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization. Technical Report, Zhengzhou University, China and Nanyang Technological University, Singapore.
Zaharie, D., 2009. Influence of crossover on the behavior of differential evolution algorithms. Appl. Soft Comput. 9 (3), 1126–1138.
Zamani, H., Nadimi-Shahraki, M.H., Gandomi, A.H., 2021. QANA: Quantum-based avian navigation optimizer algorithm. Eng. Appl. Artif. Intell. 104, 104314.
Zamuda, A., Brest, J., 2015. Self-adaptive control parameters' randomization frequency and propagations in differential evolution. Swarm Evol. Comput. 25, 72–99.
Zelinka, I., Senkerik, R., Pluhacek, M., 2013. 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico. http://dx.doi.org/10.1109/CEC.2013.6557841.
Zhang, Z., Ding, S., Jia, W., 2019b. A hybrid optimization algorithm based on Cuckoo search and differential evolution for solving constrained engineering problems. Eng. Appl. Artif. Intell. 85, 254–268.
Zhang, J.Q., Zhu, X.X., Wang, Y.H., Zhou, M.C., 2019a. Dual-environmental particle swarm optimizer in noisy and noise-free environments. IEEE Trans. Cybern. 49 (6), 2011–2021.
Zhou, K., Oh, S.K., Pedrycz, W., Qiu, J., 2023. Data preprocessing strategy in constructing convolutional neural network classifier based on constrained particle swarm optimization with fuzzy penalty function. Eng. Appl. Artif. Intell. 117, 105580.
Zhu, W., Tang, Y., Fang, J.A., Zhang, W.B., 2013. Adaptive population tuning scheme for differential evolution. Inform. Sci. 223, 164–191.
Zhu, X., Zhang, H., Gao, Y., 2020. Correlations between the scaling factor and fitness values in differential evolution. IEEE Access 8, 32100–32120.
Zuo, M., Dai, G., Peng, L., Tang, Z., Gong, D., Wang, Q., 2022. A differential evolution algorithm with the guided movement for population and its application to interplanetary transfer trajectory design. Eng. Appl. Artif. Intell. 110, 104727.