Differential Evolution with Individuals Redistribution for Real Parameter Single Objective Optimization

Chengjun Li and Yang Li
School of Computer Science, China University of Geosciences, Wuhan, China
{chengjun.li, li.yang.cs}@cug.edu.cn

Abstract
Differential Evolution (DE) is quite powerful for real parameter single objective optimization. However, the ability of extending or changing search area when falling into a local optimum is still required to be developed in DE for accommodating extremely complicated fitness landscapes with a huge number of local optima. We propose a new flow of DE, termed DE with individuals redistribution, in which a process of individuals redistribution will be called when progress on fitness is low for generations. In such a process, mutation and crossover are standardized, while trial vectors are all kept in selection. Once diversity exceeds a predetermined threshold, our opposition replacement is executed, then algorithm behavior returns to original mode. In our experiments based on two benchmark test suites, we apply individuals redistribution in ten DE algorithms. Versions of the ten DE algorithms based on individuals redistribution are compared with not only original version but also version based on complete restart, where individuals redistribution and complete restart are based on the same entry criterion. Experimental results indicate that, for most of the DE algorithms, version based on individuals redistribution performs better than both original version and version based on complete restart.

1 Introduction
Real parameter single objective optimization is one of the active research fields in artificial intelligence and the basis of more complex optimization such as multi-objective optimization, constrained optimization, etc. [Brest et al., 2017]. So far, a variety of population-based metaheuristics have been proposed for this type of optimization. In the series of Congress of Evolutionary Computation (CEC), at least five competitions among population-based metaheuristics on real parameter single objective optimization have been held since 2013. We briefly introduce the winner of each of five competitions as follows.

• NBIPOP-aCMA-ES [Lozhchilov, 2013] is a variant of Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [Hansen et al., 2003] with a restart scheme including two regimes;
• L-SHADE [Tanabe and Fukunaga, 2014] is a Differential Evolution (DE) [Storn and Price, 1997] algorithm extended from SHADE [Tanabe and Fukunaga, 2013] with linear population size reduction.
• L-SHADE-EpSin [Awad et al., 2016] is a variant of L-SHADE with adaptive parameter settings and a local search based on Gaussian walks.
• UMOEAs-II [Elayed et al., 2016] employs two subpopulations evolved by an ensemble DE and CMA-ES, respectively.
• EBOwithCMAR [Kumar et al., 2017] is a self-adaptive butterfly optimizer with success history based adaption, linear population size reduction, and covariance matrix adapted retreat phase; and
• HS-ES [Zhang and Shi, 2018] is CMA-ES with univariate sampling.

Note that L-SHADE-EpSin and UMOEAI are joint winners in 2016. Among the six winners, L-SHADE and L-SHADE-EpSin are based on DE, while NBIPOP-aCMA-ES and HS-ES are based on CMA-ES. Moreover, UMOEAs-II is based on both DE and CMA-ES. In [Skvorc et al., 2019], the six winners are compared. The comparison shows that L-SHADE-EpSin, UMOEA-II, and HS-ES are the three top performers.

In brief, both DE and CMA-ES perform powerfully for real parameter single objective optimization and deserve to be further studied. In this paper, we choose to scrutinize DE.

In DE, population, the set of tentative solutions, i.e. individuals or target vectors, is driven by the operators - mutation, crossover, and selection. In generation $g$, mutant vectors $\vec{u}_{i,g}(i = 1, 2, ..., NP)$, where $NP$ denotes population size, are produced based on target vectors $\vec{x}_{i,g}$ by mutation. After mutation, trial vectors $\vec{v}_{i,g}$ are generated based on both $\vec{x}_{i,g}$ and $\vec{v}_{i,g}$ by crossover. Thus, in DE, crossover and mutation together are specified as trial vector generation strategy. For selection, $\vec{x}_{i,g+1}$ is selected from $\vec{x}_{i,g}$ or $\vec{v}_{i,g}$ based on their fitness computed by function evaluation.

Motivation. Either benchmark test functions or real-world problems of real parameter single objective optimization often have very complex fitness landscapes. It is very common...
that there are a huge number of local optima in fitness landscapes. The above fact is a challenge to algorithms. During an execution of DE for real parameter single objective optimization, individuals tend to get closer and closer in distance under the control of the operators. That is, diversity of population goes lower and lower. Thus, search area decided by individuals distribution becomes smaller and smaller. Then, all or most individuals may enter the neighbourhood of an optimum. After that, although the found optimum may be just a local optimum, no solution better than the optimum can be found any more because the optimum is the best solution in the current search area. Therefore, extending and changing search area when execution has fallen into a local optimum is required to keep long-term search ability of DE for real parameter single objective optimization.

Six population-based metaheuristics from the CEC competitions including five winners are executed by us for much larger Maximum Number of Function Evaluations $MFES$ than the value used in existing competitions and papers. We find that these algorithms for real parameter single objective optimization can hardly make progress in the latter stage of execution. That is, the similar tendency of falling into local optimum widely exists in different types of population-based metaheuristics for real parameter single objective optimization. Considering the leading position of DE in the field of real parameter single objective optimization, our idea of improving the ability of DE will contribute to the whole field.

In this paper, we propose to equip DE with individuals redistribution. In the scheme, a process of individuals redistribution starts given that progress on fitness is continuously lower than a predetermined degree for a certain number of generations. In such a process, mutation is changed to $DE/rand/1$ with $F = 1.0$, while crossover is set to binomial crossover with $CR = 0.5$. Meanwhile, trial-vector-kept mutation strategy, adaptive evolution mechanism, and two benchmark test suites.

The state-of-the-art DE algorithms - SaDE, JADE, CoDE, EPSDE, CoBiDE, and IDE - are involved in our experiments directly or indirectly, and the up-to-date DE algorithms - MLCC-SI and NDE - as well. Further, two winners and a good performer in competitions - L-SHADE, L-SHADE-EpSin, and L-SHADE-RSP - are studied in this paper. In fact, L-SHADE-EpSin is also one of the top performers in all the competitions.

The CEC 2014 and CEC 2017 benchmark test suites are both used in our experiments. After large $MFES$, results of version based on our individuals redistribution are compared with not only original version but also version based on complete restart. Results show that version based on complete restart performs better than original version. More importantly, for most of the DE algorithms, version with individuals redistribution performs better than not only original version but also version with complete restart.

The major contribution of this paper is summarized as

- showing that, for real parameter single objective optimization, population-based metaheuristics including DE tend to fall into local optimum in many occasions;
- proposing a new flow of DE - DE with individuals redistribution - for extending and changing search area to jump out of local optimum; and
- validating general applicability and effectiveness of the new flow by experiments based on ten DE algorithms and two benchmark test suites.

Figure 1: Sketch of process of individuals redistribution in two-dimensional search space

(a) The initial state  (b) The first step  (c) The second step

- L-SHADE [Tanabe and Fukunaga, 2014] is extended from SHADE [Tanabe and Fukunaga, 2013] - a variant of JADE [Zhang and Sanderson, 2009].
- CoBiDE [Wang et al., 2014] is based on covariance matrix learning and parameters adaption.
- L-SHADE-EpSin [Awad et al., 2016] is extended from L-SHADE.
- SaDE/Mexp [Qiu et al., 2016] is an improved version of SaDE [Qin and Suganathan, 2005] with multiple exponential crossover.
- MPEDE [Wu et al., 2016] is based on parameters adaption and three subpopulations controlled by different mutation strategies.
- ETI-JADE [Du et al., 2017] is an improved version of JADE with Event-Triggered Impulsive (ETI) scheme.
- L-SHADE-RSP [Stanovov et al., 2018] ranks second in the CEC 2018 competition. Based on L-SHADE, the algorithm employs rank-based mutation scheme in which better target vector has larger possibility to be chosen in the secondary differential item.
- EDEV [Wu et al., 2018] is based on three competing subpopulations controlled by JADE, CoDE [Wang et al., 2011], and EPSDE [Mallipeddi et al., 2011].
- MLCC-SI [Zhang et al., 2019] is a multi-layer competitive-cooperative framework based on SHADE and IDE [Tang et al., 2015].
- NDE [Tian and Gao, 2019] is with neighborhood-based mutation strategy, adaptive evolution mechanism, and linear population size reduction.
Data Analysis

Here we run six metaheuristics selected from the competitions - L-SHADE, L-SHADE-EpSin, UMOEAs-II, EBOwithCMAR, jSO [Brest et al., 2017], and HS-ES. Except that jSO ranks second in the CEC 2017 competition, each of the other five algorithms is the winner in different competitions. These algorithms are executed for the CEC 2017 benchmark test suite. In the suite, hybrid functions, F11-F20, and composition functions, F21-F30, have much more local optima in fitness landscapes than unimodal functions, F1-F3, and simple multimodal functions, F4-F10. MFES is set $3.0E + 06$ for all the algorithms. The value is 10 times as much as the value widely used in competitions and papers. Besides, all the other parameters are taken the original value in literature.

According to results, global optimum of 21 functions - F5, F11-F30 is never found. That is, it is hard for the six algorithms to obtain global optimum of F11-F30, the functions with a huge number of local optima in fitness landscapes. For F11-F30, we plot the averages of the best error after the first generation and at 100 intervals of all the DE algorithms. We find that, for the functions, all the algorithms can hardly make progress in the latter stage of execution. In brief, even winner or good performer in the competition among population-based metaheuristics for real parameter single objective optimization do fall into local optimum in many occasions.

The aforementioned data analysis shows that the tendency of falling into local optimum widely exists in different types of metaheuristics for real parameter single objective optimization including DE. Now that DE performs well with possibility of falling into local optimum, a new flow of DE for extending and changing search area to leave local optimum is meaningful in the field of real parameter single objective optimization.

Related Work

Firstly, recent methods which can be widely used in DE algorithms for improving solutions are briefly summarized here. In multiple exponential crossover [Qiu et al., 2016], multiple segments of individuals are exchanged. SaDE/Mexp is based on multiple exponential crossover. Under the control of ETI scheme [Du et al., 2017], stabilizing impulses and destabilizing impulses are executed based on event-triggered mechanism. In fact, destabilizing impulses serve as a partial restart strategy. ETI-JADE is based on ETI scheme. Adaptive social learning strategy [Cai et al., 2018] can extract the neighborhood relationship information of individuals. Multitopology-based DE [Sun et al., 2018] includes multiple population topologies, individual-dependent adaptive topology selection scheme, and topology-dependent mutation strategy.

Moreover, ensemble DE algorithms - DE algorithms with more than one combinations of operators - are also related to our work because, to apply our scheme, a combination of the changed operators need be added in DE algorithms. In MPEDE, mutation is set differently in subpopulations. In LSAOS-DE [Sallam et al., 2017], each combination controls a subpopulation in a part of generations, while the combination showing the best performance seizes the whole population in the other generations. According to [Cui et al., 2018], two trial vector generation strategies both control the whole population in a part of generations, while population is controlled by only one of the two strategies in other generations. EDEV employs three state-of-the-art DE algorithms to control three subpopulations, respectively. MLCC framework [Zhang et al., 2019] implements a parallel structure with the entire population simultaneously monitored by multiple DEs assigned to different layers. Thus, individuals are processed in different layers. MLCC-SI is based on MLCC framework.

Among the DE algorithms listed in related work, SaDE/Mexp, MPEDE, ETI-JADE, EDEV, and MLCC-SI are involved in our experiments. Thus, our scheme can be validated even based on related work.

Methodology

We propose DE with individuals redistribution - a new flow of DE - to improve DE at the aspect of jumping out of local optimum for keeping search ability in long run. As the name implies, individuals redistribution makes the distribution of individuals significantly change. Details of our scheme are given as below.

Only if improvement on the best fitness is lower than a given threshold $T_{H0}$ for a given number of contiguous generations $G_N$, a process of individuals redistribution is called. In this way, individuals redistribution is always executed when making progress is difficult, or, in other words, when the current run falls into a local optimum. As soon as that the current best fitness becomes the best fitness in the current run, $G_N$ is doubled based on the given value to provide more generations for obtaining an ever-best solution.

In DE algorithms, operators are designed for convergence. However, for individuals redistribution, original operators are replaced by the changed operators to diversify population. For any DE algorithm, the changed operators are set same as below. Firstly, original mutation strategy is changed to DE/rand/1 with scaling factor $F = 1.0$ as shown in Equation (1)

$$u_{i,g} = \bar{x}_{i,g} + 1.0 \cdot (\bar{x}_{r1, g} - \bar{x}_{r2, g})$$

where $r1 \in \{1, 2, ..., NP\}$, $r2 \in \{1, 2, ..., NP\}$, and $r1 \neq r2 \neq i$. Then, crossover is set to binomial crossover with $CR = 0.5$. Let $t_{i,g} = (u_{1_i,g}, u_{2_i,g}, ..., u_{N_i,g})$, where $n$ denotes the number of dimensions. Our crossover method can be expressed by Equation (2).

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & \text{if rand}(0, 1) \leq 0.5 \\ \bar{x}_{j,i,g}, & \text{otherwise}. \end{cases}$$

Further, trial-vector-kept, which means keeping all trial vectors, is used as selection in this stage. In short, for individuals redistribution, two aspects are changed in algorithm. On one hand, trial vector generation strategy is standardized. More importantly, fitness is not considered in selection to remove selective pressure. Thus, diversity may show an up trend. The changed operators are executed in contiguous generations until diversity reaches a given value, $T_{DIV}$.

For real parameter single objective optimization, diversity can be denoted as the average of distance from individual to
Definition 1 (Population center). Let \( P_g = \{ \hat{x}_{1,g}, \hat{x}_{2,g}, ..., \hat{x}_{NP,g} \} \) be the \( g \)th generation of population, where \( \hat{x}_{i,g} = (x_{1,i,g}, x_{2,i,g}, ..., x_{n,i,g}) (i = 1, 2, ..., NP) \). Let \( m_j \) be the median of \( x_{j,1,g}, x_{j,2,g}, ..., x_{j,NP,g} \) where \( j = 1, 2, ..., n \). The center of \( P_g \) is \( c_g = (m_1, m_2, ..., m_n) \).

Here, based on normalized Manhattan distance, diversity is calculated according to Equation (3),

\[
div_g = \frac{\sum_{i=1}^{NP} \sum_{j=1}^{n} |x_{j,i,g} - m_j|}{NP} \tag{3}
\]

where \( up_j \) and \( low_j \) are upper limit and lower limit of the \( j \)th dimension, respectively.

After diversity reaches \( T_{DIV} \), a portion of target vectors, which are randomly chosen, are replaced by their opposite vector in search space. This step is called opposition replacement by us. Then, evolution goes back to original mode. The definition of opposite vector in search space is given below.

Definition 2 (Opposite vector in search space). Let \( \tilde{x}_{i,g} = (x_{1,i,g}, x_{2,i,g}, ..., x_{n,i,g}) \) be a target vector. The opposite vector of \( \tilde{x}_{i,g} \) is completely defined by its components

\[
x_{j,i,g}^o = up_j + low_j - x_{j,i,g} \tag{4}
\]

where \( j \in \{1, 2, ..., n\} \) and \( x_{j,i,g} \in [up_j, low_j] \).

According to our scheme, it is difficult to make progress in fitness any more, population is diversified gradually and slowly until diversity rises to \( T_{DIV} \). After opposition replacement is done, a portion of target vectors evolve far from the gathering position before individuals redistribution. Meanwhile, the other target vectors still locate in the vicinity of the position. After that, target vectors delimit a new search area. The new search area is larger than the search area before individuals redistribution and has very limited overlap ratio with the old one. It need be emphasized that our scheme preserves no elite.

In Algorithm 1, the pseudo-code of DE with individuals redistribution is given.

- \( b = 0 \) means original mode, while \( b = 1 \) means generations for individuals redistribution. It can be seen from Algorithm 1 that, if diversity is already higher than \( T_{DIV} \) before the start of individuals redistribution, just one generation with the changed operators is executed. Moreover, beside the main criterion - diversity higher than \( T_{DIV} \), another criterion - the number of contiguous generations for the changed operators exceeding our predetermined limit - is also used for switching from the changed operators to original ones. We set the secondary criterion because, although it has a very high probability that diversity is increased beyond \( T_{DIV} \) under the control of the changed operators after generations, very few counter examples can be found.

Neither the changed operators nor opposition replacement is not based on fitness. Thus, during a process of individuals redistribution, function evaluations are required just after opposition replacement for the following old mode of evolution. That is, only \( NP \) times of function evaluations are needed in a process of individuals redistribution. Nevertheless, the higher \( T_{DIV} \) is set, the more generations without function evaluations for the changed operators are required.

For DE algorithms with linear population size reduction, further consideration is required. In these algorithms, population size decreases linearly. When diversity is improved by individuals redistribution, population size needs increase again. Detailed steps are given below. The value of population size is recorded when diversity is for the first time lower than \( T_{DIV} \) in an execution. When the course for diversifying begin, not only trial vectors but also target vectors are
selected to increase population size until population size recovers to the recorded value.

5 Experiments

The ten DE algorithms are involved in experiments. Their parameters excepting $MFES$ are from literature. Three versions of the algorithms, original version (OV), version based on complete restart (CRV), and version based on our individuals redistribution (IRV), are compared. Entry criterion is set the same for CRV and IRV. The reason why studying CRV is as below. Complete restart in execution gives a new chance of evolution from beginning again. Now that population of DE is easy to fall into a local optimum, provided that the entry criterion of restart is set well, outcome of an execution of CRV setting a large value in $MFES$ corresponds to the best outcome of multiple executions of OV with standard $MFES$ setting. In brief, although complete restart does not belong to a contribution in theory of DE, it may lead to improvement of solution in long execution. If results based on CRV are better than results of OV, we can judge that parameters have been set well for IRV because IRV and CRV are based on the same entry criterion. Furthermore, comparing results based on IRV with results based on CRV can further validate our finding. Before formal experiments, we run OV of the ten DE algorithms for the CEC 2017 benchmark test suite. $MFES$ is set $3.0E + 06$, while dimensionality is set 30. We do Friedman rank sum test at a 0.05 significance level for results and find that, at least for the CEC 2017 functions with 30 in dimensionality, DE algorithms selected from the competitions - L-SHADE-RSP, L-SHADE-EpSin, and L-SHADE - perform best among the ten algorithms.

In the first experiment, the three versions of the ten DE algorithms are compared based on the CEC 2017 benchmark test suite. Functions in the suite are set 30 in dimensionality. Parameters for the three versions are listed in Table 1. In the table, $MFES$ is required for all the three versions. Moreover, $G_N$, $T_{IR}$, and $R$ are for both CRV and IRV, while $T_{DIV}$ and $T_{GEN}$ is for IRV only. We double $MFES$ for L-SHADE-RSP because, under $MFES = 3.0E + 06$, restart can be hardly observed in CRV of the DE algorithm. We employ seven values to threshold of diversity for returning to original mode - $T_{DIV}$ - because the suitable value of $T_{DIV}$ depends on not only algorithm but also fitness landscapes of function. Table 2 gives a summary of results.

In addition, the ratio of times of showing the best result under different $T_{DIV}$ is given in Table 3.

It can be seen from Table 2 that, for all the ten DE algorithms, CRV performs better than OV. Nevertheless, in the comparison for L-SHADE-EpSin and L-SHADE-RSP, the advantage is just slight. In fact, it can be observed from detailed outcomes that, within the current value of $MFES$, execution of OV of the two algorithms still do not completely fall into a local optimum in many cases. CRV cannot show its best in this situation. In short, according to Table 2, complete restart can improve results of DE algorithms as long as parameters for complete restart are set appropriately.

It can be seen from Table 2 that IRV outperforms OV. Further, comparison between IRV and OV show a larger ratio than comparison between CRV and OV for L-SHADE-EpSin, MPEDE, ETI-JADE, L-SHADE-RSP, EDEV, and NDE. Meanwhile, the ratio is same in the comparison for CoBiDE, SaDE/Mexp. Only in the comparison for L-SHADE and MLCC-SI, the ratio decreases. More importantly, in the comparison for all the DE algorithms excepting MLCC-SI, the times of IRV winning CRV is larger than the times of losing. However, for EDEV, the advantage of IRV is not distinct. In short, according to Table 2, IRV shows better performance than CRV for most of the algorithms.

According to Table 3, the best value of $T_{DIV}$ among the seven values is often large. In detail, the best value for L-SHADE, CoBiDE, L-SHADE-EpSin, SaDE/Mexp, MPEDE, ETI-JADE, L-SHADE-RSP is 1.0E-01. Meanwhile, 5.0E-02 is the best choice for CoBiDE, L-SHADE-EpSin, ETI-JADE, NDE. It can be seen that, both 1.0E-01 and 5.0E-02 are the best choices for CoBiDE, L-SHADE-EpSin, and ETI-JADE. Just EDEV and MLCC-SI are not fit for large value in $T_{DIV}$.

In the second experiment, the three versions are compared based on the CEC 2014 benchmark test suite. Here, a part of the ten DE algorithms are selected to be executed. Above all, L-SHADE-EpSin, one of the top performers in the CEC competitions and NDE, one of the latest DE algorithms, are the reasonable choices. Besides, ETI-JADE, which is based on partial restart, need be further studied. Finally, the three en-
ensemble DE algorithms - MPEDE, EDEV, and MLCC-SI - are involved in this experiment because the previous experiment shows that our scheme performs not so well for EDEV and is not appropriate to MLCC-SI at all. Parameters for the three versions different with the previous experiment are listed in Table 4. In fact, when $MFES$ and $G_N$ are set $3.0E + 06$ and $500$, respectively, for L-SHADE-EpSin, we find that complete restart leads to worse solutions in most cases. It means that restart is called when execution still does not fall into a local optimum. Therefore, we double $G_N$. Meanwhile, we double $MFES$. Table 5 summarizes results.

Then, ratio of times of showing the best result under different $T_{DIV}$ is given in Table 6. It can be seen from Table 5 that, for MPEDE, ETI-JADE, EDEV, MLCC-SI, and NDE, CRV outperforms OV. However, for L-SHADE-EpSin, although the parameters are doubled, complete restart still cannot lead to improvement in any case but bring deterioration in cases. According to the table, IRV always outperforms OV. The advantage for MLCC-SI is very slight. Meanwhile, IRV wins CRV for the algorithms excepting MLCC-SI. In addition, the advantage for EDEV and L-SHADE-EpSin is less than for MPEDE, ETI-JADE, and NDE. It needs be emphasized that, for L-SHADE-EpSin, CRV cannot win OV. That is, the parameters are still not suitable for both complete restart and individuals redistribution. However, IRV shows advantage on both OV and CRV.

According to Table 6, the best value of $T_{div}$ becomes smaller for L-SHADE-EpSin, MPEDE, ETI-JADE, and NDE than in the previous experiment. Meanwhile, EDEV and MLCC-SI are still fit for the smallest value in $T_{div}$.

6 Conclusion and Future Work

We find that the tendency of falling into local optimum widely exists in metaheuristics for real parameter single objective optimization. In this paper, we proposed DE with individuals redistribution for keeping search ability in long run. Our experiments based on ten DE algorithms and two benchmark test suites show that individuals redistribution is better at improving solutions for real parameter single objective optimization than complete restart. Considering the leading position of DE in the field of real parameter single objective optimization, our finding contributes to the whole field.

Besides special scheme for calling individuals redistribution in ensemble DE algorithm, we want to obtain an adaptive method for setting $T_{DIV}$ to simplify application of individuals redistribution. Moreover, after adaption, individuals redistribution may be realized in metaheuristics for parameter single objective optimization other than DE, such as HS-ES and UMOEA-II. The above all will be accumulated as our future work.

References

[Awad et al., 2016] Noor H. Awad, Mostafa Z. Ali, Ponnuthurai N. Suganthan, and Robert G. Reynolds. An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems. In Proc. of CEC, pages 2958–2965. IEEE, 2016.

[Brest et al., 2017] Janez Brest, Mirjam Sepesv Mauj, and Borko Bośković. Single objective real-parameter optimization: Algorithm jSO. In Proc. of CEC, pages 1311–1318. IEEE, 2017.

[Cai et al., 2018] Yiqiao Cai, Jingliang Liao, Tian Wang, Yonghong Chen, and Hui Tian. Social learning differential evolution. Information Sciences, 433:464–509, 2018.

[Cui et al., 2018] Laizhong Cui, Genghui Li, Zexuan Zhu, Quzhen Lin, Ka-Chun Wong, Jianyong Chen, Nan Lu,
and Jian Lu. Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism. *Information Sciences*, 422:122–143, 2018.

[Du et al., 2017] Wei Du, Sunney Yung Sun Leung, Yang Tang, and Athanasios V Vasiliakos. Differential evolution with event-triggered impulsive control. *IEEE Trans. on Cybernetics*, 47(1):244–257, 2017.

[Elsayed et al., 2016] Saber Elsayed, Noha Hamza, and Ruhul Sarker. Testing united multi-operator evolutionary algorithms-II on single objective optimization problems. In *Proc of CEC*, pages 2966–2973. IEEE, 2016.

[Hansen et al., 2003] Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation*, 11(1):1–18, 2003.

[Kumar et al., 2017] Abhishek Kumar, Rakesh Kumar Misra, and Devender Singh. Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase. In *Proc of CEC*, pages 1835–1842. IEEE, 2017.

[Loshchilov, 2013] Ilya Loshchilov. CMA-ES with restarts for solving CEC 2013 benchmark problems. In *Proc of CEC*, pages 369–376. IEEE, 2013.

[Mallipeddi et al., 2011] Rammohan Mallipeddi, Ponnuthurai N Suganthan, Quan-Ke Pan, and Mehmet Fatih Tasgewire. Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing*, 11(2):1679–1696, 2011.

[Qin and Suganthan, 2005] A Kai Qin and Ponnuthurai N Suganthan. Self-adaptive differential evolution algorithm for numerical optimization. In *Proc of CEC*, volume 2, pages 1785–1791. IEEE, 2005.

[Qiu et al., 2016] Xin Qiu, Kay Chen Tan, and Jian-Xin Xu. Multiple exponential recombination for differential evolution. *IEEE Trans. on Cybernetics*, 47(4):995–1006, 2016.

[Sallam et al., 2017] Karam M Sallam, Saber M Elsayed, Ruhul A Sarker, and Daryl L Essam. Landscape-based adaptive operator selection mechanism for differential evolution. *Information Sciences*, 418:383–404, 2017.

[Škvorc et al., 2019] Urban Škvorc, Tome Eftimov, and Peter Korošec. CEC real-parameter optimization competitions: Progress from 2013 to 2018. In *Proc of CEC*, pages 3126–3133. IEEE, 2019.

[Stanovov et al., 2018] Vladimir Stanovov, Shakhnaz Akhmedova, and Eugene Semenkin. LSHADE algorithm with rank-based selective pressure strategy for solving CEC 2017 benchmark problems. In *Proc of CEC*, pages 1–8. IEEE, 2018.

[Storn and Price, 1997] Rainer Storn and Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.