A Differential Evolution Framework with Ensemble of Parameters and Strategies and Pool of Local Search Algorithms

Giovanni Iacca[1], Ferrante Neri[2,3], Fabio Caraffini[2,3], and Ponnuthurai Nagaratnam Suganthan[4]

[1] INCAS[3]
Dr. Nassaulaan 9, 9401 HJ, Assen, The Netherlands
giovanniiacca@incas3.eu
[2] Centre for Computational Intelligence,
School of Computer Science and Informatics,
De Montfort University, The Gateway, Leicester LE1 9BH, England, UK
{fneri,fcaraffini}@dmu.ac.uk
[3] Department of Mathematical Information Technology,
P.O. Box 35 (Agora), 40014 University of Jyväskylä, Finland
{ferrante.neri,fabio.caraffini}@jyu.fi
[4] School of Electrical & Electronic Engineering, College of Engineering,
Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798
epnsugan@ntu.edu.sg

Abstract. The ensemble structure is a computational intelligence supervised strategy consisting of a pool of multiple operators that compete among each other for being selected, and an adaptation mechanism that tends to reward the most successful operators. In this paper we extend the idea of the ensemble to multiple local search logics. In a memetic fashion, the search structure of an ensemble framework cooperatively/competitively optimizes the problem jointly with a pool of diverse local search algorithms. In this way, the algorithm progressively adapts to a given problem and selects those search logics that appear to be the most appropriate to quickly detect high quality solutions. The resulting algorithm, namely Ensemble of Parameters and Strategies Differential Evolution empowered by Local Search (EPSDE-LS), is evaluated on multiple testbeds and dimensionality values. Numerical results show that the proposed EPSDE-LS robustly displays a very good performance in comparison with some of the state-of-the-art algorithms.

Keywords: Differential Evolution, Global Optimization, Ensemble, Parameter Adaptation, Mutation Strategy Adaptation.

1 Introduction

Differential Evolution (DE) [24] is a simple, fast and efficient stochastic algorithm with few parameters to tune [4, 21]. After the early DE implementations, important efforts have been made to improve the performance by introducing different

mutation and crossover strategies [5, 26, 36, 4, 21]. The choice of appropriate mutation and crossover strategies (as well as their related control parameters) is not easy due to the complex interaction between them [1]. An inappropriate choice of strategies or parameters may lead to an efficient behaviour of the algorithm. Thus, various empirical guidelines were suggested for choosing a mutation strategy and its associated control parameter settings, see e.g. [17] and [35]. Although these guidelines are rather useful for choosing the mutation parameters, the performance of DE is still sensitive to the combination of the mutation and crossover strategies, with their associated parameters. Furthermore, the best setting of mutation strategy, crossover strategy and control parameters can be different for different optimization problems. Based on these observations, different adaptation schemes have been proposed in the past years, see e.g [1, 25, 34] to overcome the time consuming trial-and-error procedure and let the algorithm self-adapt to the fitness landscape.

As an alternative to adaptation, in [16] a DE framework with an *ensemble* of mutation and crossover strategies and parameter values (known as EPSDE: Ensemble of Parameters and Strategies in DE) was proposed. EPSDE contains a pool of mutation and crossover strategies along with a pool of values corresponding to each associated parameter which compete to produce successful offspring. Due to its richness of search moves, EPSDE has proved so far extremely successful on many different optimization problems. In different contexts, a similar logic has been employed, see e.g. [32].

In this paper, we extend the concept of ensemble by combining the ensemble of strategies and parameters proposed in [16] with a pool of local search algorithms, as suggested in [30]. More specifically, three different local search algorithms, which co-exist and compete to produce better solutions, are embedded within the EPSDE framework in order to improve upon its performance. The proposed algorithm, referred to as EPSDE-LS, is evaluated on two different benchmarks in comparison with three state-of-the-art optimization algorithms.

The reminder of this paper is organized as follows. Section 2 presents the proposed EPSDE-LS algorithm. Section 3 presents the numerical results. Finally, Section 4 concludes the paper and suggests some possible future developments.

2 Ensemble of Parameters and Strategies Differential Evolution empowered by Local Search

This paper proposes an extension of the concept of EPSDE in a memetic fashion. More specifically, the proposed algorithm, namely Ensemble of Parameters and Strategies Differential Evolution empowered by Local Search (EPSDE-LS), integrates within a EPSDE framework also a pool of local search algorithms (see Alg. 1). The main motivation behind the proposed design, besides the benefits of DE memetic schemes, see e.g. [12], [19] and [20], is that an a priori design of an algorithm should take into account the features of the optimization problems, such as ill-conditioning, separability, multimodality, see [2] and [11]. Thus, multiple search logics are here blended within the same framework. The algo-

rithm should progressively "learn" how each component can successfully tackle the features of the optimization problem and progressively adapt to the problem and solve it efficiently.

Let us describe the proposed EPSDE-LS more in detail, considering the EPSDE framework first and then the pool of local search algorithms. As described in [16], EPSDE makes use of the following pools of strategies:

- Pool of mutation strategies: $P_{mut} = \{\text{cur-to-pbest/1}, \text{cur-to-rand/1}\}$
- Pool of crossover strategies: $P_{cross} = \{\text{bin}, \text{exp}\}$

In addition to that, two pools of mutation an crossover parameters are defined, namely P_F and P_{CR}. The mutation strategies are defined as:

- DE/cur-to-pbest/1: $x'_{off} = x_i + F\,(x^p_{best} - x_i) + F\,(x_s - x_t)$
- DE/cur-to-rand/1: $x_{off} = x_i + K\,(x_r - x_i) + F\,(x_s - x_t)$.

Regarding DE/cur-to-rand/1, it should be observed that the crossover operation is not applied, as this mutation strategy contains an implicit arithmetic crossover, see [25]. On the contrary, when DE/cur-to-pbest/1 strategy is selected, a crossover completes the offspring generation, see [36]. Moreover, x^p_{best} is an individual randomly selected amongst the best $100 \cdot p\%$ where p is a dynamic value that varies between 0 and 1 according to the following rule:

$$p = \lfloor 0.005 \cdot (1 - n_{eval}/N_{eval}) \rfloor \tag{1}$$

where n_{eval} and N_{eval} indicate, respectively, the current and maximum number of fitness evaluation. In this way, at the beginning of the optimization ($n_{eval} = 0$) the mutation uses a random solution among the top 50% individuals in the current population: this guarantees a higher chance of taking a suboptimal solution, thus increasing the exploration pressure. Later on ($n_{eval} \to N_{eval}$), the percentage of top individuals will progressively decrease, thus making the mutation strategy more exploitative.

The selection of these strategies is motivated by the consideration that they offer diverse and complementary search moves. As a general observation, as highlighted in [21], DE is characterized by a limited amount of search moves. Hence, the employment of multiple mutations and crossover compensates the lack of DE search moves. However, it is important to note that in order to have an effective ensemble, the candidate pools of mutation/crossover strategies and parameters must be chosen so to avoid the unfavorable influences of less effective mutation strategies and parameters [27]. In other words, the strategies and parameters present in the pools should have diverse characteristics, so that they can exhibit distinct performance during different stages of the evolution, as well as when dealing with different problems, see [21].

In the EPSDE framework, the mutation and crossover strategies have been chosen as they correspond to two diverse search logics. More specifically, the DE/cur-to-rand/1 mutation strategy attempts to enhance upon the performance of each population individual by adding to it two randomized vectors:

$$x_{off} = x_i + K\,(x_r - x_i) + F\,(x_s - x_t). \tag{2}$$

This operation is rather exploratory as it can potentially reach every point of the decision space and each offspring is loosely related to the generating parent (e.g. no sequences of design variables are copied from the parent to the offspring). Conversely, DE/cur-to-pbest/1 is a fairly exploitative mutation strategy as it makes use of a fitness based criterion to increase the selection pressure. Since part of the mutation takes into account only those solutions that display the best fitness values, the mutation excludes some search moves and exploit only those search directions that appear the most promising. In addition, the crossover application makes the offspring more similar to the parent that has generated it, thus further increasing the exploitation. The two crossover strategies allow different degrees of exploration/exploitation balance. The DE/cur-to-pbest/1 mutation followed by exponential crossover is the most exploitative option while the offspring generation by DE/cur-to-pbest/1 mutation and binomial crossover contains a higher exploratory potential, see [31]. The exponential crossover leads to a copy of contiguous design variables while the binomial tends to copy scattered variables. This fact has an impact especially in non-separable problems where the inter-variable interaction can be strong.

Regarding the ensemble coordination and adaptation, EPSDE operates as follows. At the beginning of the optimization, each member in the initial population is randomly assigned a mutation/crossover strategy and the associated parameter values taken from the respective pools. Then, during each generation, the population members (target vectors) produce offspring (trial vectors) using the assigned mutation/crossover strategies and parameter values. If the trial vector is better than the target vector, in the next generation the mutation/crossover strategies—and the corresponding parameter values—are retained, while the trial vector replaces its parent (target vector). Otherwise, the target vector is retained and randomly associated, with equal probability, to new mutation/crossover strategies and associated parameter values from the respective pools. Thus, this mechanism relies on the selection properties of evolution to increase, while the optimization process goes on, the probability of producing offspring by the best combinations of strategies and parameters. In summary, the ensemble is a simple and straightforward self-adaptation where the fittest strategies survive along with the solutions that have generated.

Let us consider now the pool of local search algorithms. The proposed EPSDE-LS employs a pool P_{LS} containing three algorithms, namely Nelder-Mead simplex [18], Powell's conjugate direction method [22], and Rosenbrock's algorithm [28]. As in the case of offspring generation within EPSDE, the pool has been selected in order to empower the algorithm with multiple and diverse search operators.

As an initial note, the three local search algorithms can be divided into two groups. We should remark indeed, that while Powell's and Rosenbrock's algorithms require only an initial point to start the search, the simplex algorithm requires $n + 1$ points. In our case, for the first two algorithms we initialize the initial point to the current best solution in the EPSDE population. In Nelder-Mead algorithm, we instead initialize the first point of polytope to the current

best solution, while the remaining n points are initialized randomly. Moreover, while Rosenbrock's and Powell's algorithms are purely deterministic local search operators, Nelder-Mead algorithm contains some randomization features due to the random initialization of n points that generate the polytope. Since these n points can be sampled apart from each other within the decision space, Nelder-Mead algorithm contains some global search features and thus has the potential of jumping outside a basin of attraction and detect new promising search directions. On the contrary, Rosenbrock's and Powell's algorithms tend to exploit the starting solution and detect the closest optimum.

Furthermore, although both Rosenbrock's and Powell's algorithms belong to the same local search category, they present different features in terms of search logic. While Rosenbrock's algorithm follows the local gradient by means of a rotation matrix that changes the coordinate system, Powell's algorithm makes use of the conjugate search directions. This fact causes that Rosenbrock's algorithm performs a single diagonal move while Powell's algorithm performs a diagonal move as the result of n conjugate steps where the fitness has separately been optimized along each direction.

As for the coordination of the local search within the EPSDE framework, we adopted the following scheme. Every F_{LS} generations of the EPSDE framework (being F_{LS} a prefixed parameter, namely the local search activation frequency), the algorithm selects randomly one of the three locals search methods from the pool P_{LS}. The local search is then applied to the individual of the EPSDE population displaying the best performance, with a fixed computational budget (number of fitness evaluations) B_{LS}.

We should note that the employed coordination of local search has been chosen in consideration of the Ockham's Razor in Memetic Computing, i.e. an algorithmic design should be performed avoiding unnecessary components and attempting at first to achieve the desired performance in the simplest way. In this light, the local search activation by random selection of the meme/operator is likely one of the simplest way to perform the design of a hybrid algorithm.

Moreover, a straightforward extension to local search of the ensemble logic cannot be efficiently performed. In EPSDE, the trial of a strategy is based on a single fitness evaluation. On the contrary, in order to observe an improvement with the local search, a certain budget allocation must be considered. Thus, if a reward is given to the most successful local search algorithm, there is a high risk that only one algorithm is used while the remaining two are disregarded. This action would inhibit the logic of multiple and diverse search logics thus resulting in a biased search. In other words, the selection pressure over the successful local search strategy (that is applied in EPSDE with the most successful mutation/crossover strategies) is implicit in this case, since the same operator is anyway applied iteratively until budget exhaustion.

As a final remark, we should note that the choice of performing the local search over the best individual of the population is due to the DE nature/structure of the EPSDE scheme. As shown in [10], DE frameworks appear to work successfully when one solution displays a much better fitness than the

average population performance. The best solution, namely super-fit, guides the search and allows quick progression of the population. Thus, if the local search is always applied to the best solution there is the highest likelihood to generate a super-fit individual.

Algorithm 1 EPSDE-LS pseudo-code

```
initialize a pool of mutation strategies P_mut and crossover strategies P_cross
initialize a pool of scale factors P_F and crossover probabilities P_CR
generate N_p individuals of the initial population pseudo-randomly
for i = 1 : N_p do
   assign to x_i random strategies/parameters from {P_mut, P_cross, P_F, P_CR}
   compute f(x_i)
end for
g = 1
while budget condition do
   for i = 1 : N_p do
      generate x'_off through mutation strategy/parameter associated to x_i
      generate x_off through crossover strategy/parameter associated to x_i
      if f(x_off) ⩽ f(x_i) then
         save index i for replacing x_i = x_off (including mutation and crossover strate-
         gies as well as parameters) in the next generation
      else
         assign to x_i new random strategies/parameters from the pools
      end if
   end for
   perform replacements
   g = g + 1
   if (g mod F_LS) = 0 then
      select a local search algorithm from the pool P_LS
      apply it to x_best, until the budget condition B_LS
   end if
end while
```

3 Numerical Results

In order to assess the performance of EPSDE-LS on a broad set of real-parameter optimization problems, we evaluated the minimization results obtained by the proposed algorithm on two different benchmarks, namely:

- the benchmark used at the CEC 2013 [15], composed of 28 test functions;
- the large-scale optimization benchmark used at CEC 2010 [29], composed of 20 test functions.

Furthermore, we studied the scalability properties of the proposed algorithm testing the CEC 2013 benchmark in 10, 30 and 50 dimensions, and the CEC 2010 benchmark in 1000 dimensions. We compared EPSDE-LS performance (i.e., the quality of the final solutions) with that of the following algorithms:

– Modified Differential Evolution + pBX crossover (MDE-pBX) [13], with population size equal to 100 individuals and group size q equal to 15% of the population size;
– Cooperatively Coevolving Particle Swarms Optimizer (CCPSO2) [14], with population size equal to 30 individuals, Cauchy/Gaussian sampling selection probability $p = 0.5$ and set of potential group sizes $S = \{2, 5\}$, $S = \{2, 5, 10\}$, $S = \{2, 5, 10, 25\}$, for experiments in 10, 30 and 50 dimensions, respectively;
– Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [8], with the default parameter setting of the original implementation [7], namely $\lambda = \lfloor 4 + 3\ln(\mathrm{D}) \rfloor$, $\mu = \lfloor \lambda/2 \rfloor$, and initial step-size $\sigma = 0.2$.

As for EPSDE-LS, we set $N_p = 50$, $F_{LS} = 200$, and $B_{LS} = 1000$, while the parameter pools were chosen as $P_{CR} = \{0.1, 0.5, 0.9\}$ and $P_F = \{0.5, 0.9\}$. As said before we selected the following pools of strategies: $P_{cross} = \{\mathrm{bin}, \mathrm{exp}\}$ and $P_{mut} = \{\text{cur-to-pbest/1}, \text{cur-to-rand/1}\}$. The local search methods were configured as follows:

– Powell, with 100 fitness evaluations per each bi-directional line search. The Brent's line search algorithm was implemented and configured as in [23].
– Nelder-Mead: reflection coefficient $\alpha = 1$, contraction coefficient $\beta = 0.5$, expansion coefficient $\gamma = 2$ and shrinkage coefficient $\delta = 0.5$.
– Rosenbrock: positive perturbation factor $\alpha = 2$, negative perturbation factor $\beta = -0.5$, and threshold for coordinate system rotation $\epsilon = 10^{-5}$.

For each algorithm, we executed 100 independent runs, with a computational budget of $10000 \times \mathrm{D}$ fitness evaluations (where D is the problem dimension), as suggested by the CEC 2013 competition rules. As an additional note, a toroidal handling of the bounds was used for all the algorithms in this study. This means that, given an interval $[a, b]$, if $x_i = b + \zeta$, i.e. the i-th design variable exceeds the upper bound by a quantity ζ, its value is replaced with $a + \zeta$. A similar mechanism was applied for the lower bound.

The entire experimental setup (fitness functions and algorithms) was coded in Java and executed on a hybrid network composed of Linux and Mac computers, using the distributed optimization platform Kimeme [3]. Numerical results, reporting for each test function the average of the fitness error (with respect to the global optimum) obtained by each algorithm at the end of the allotted budget, with its standard deviation, are shown in Tables 1, 2, 3, and 4. Next to the average error, we report the outcome of the Wilcoxon Rank-Sum test [33] applied, with confidence level 0.95, to each pair-wise comparison between the final fitness errors shown by EPSDE-LS (taken as reference) and those shown by the algorithm in the corresponding column name. To simplify the interpretation of this test, we indicate with "=" an acceptance of the null-hypothesis (that the two algorithms under comparison are statistically equivalent from an optimization point of view), and with "+" ("-") a superior (worse) performance of EPSDE-LS with respect to the algorithm in the column label. Finally, the bold face indicates the algorithms showing the best average fitness error.

From the numerical results, it can be seen that the proposed EPSDE-LS outperforms, on a regular basis, the competing algorithms at all dimensionalities. In particular, EPSDE-LS seems particularly competitive against CMA-ES and CCPSO2, while in low-mid dimensionalities (10-50) MDE-pBX shows in many cases the lowest average error, although with a larger standard deviation and thus lower robustness. Similarly, in 1000 dimensions CMA-ES obtains in most cases the lowest error, but statistically proves equivalent to EPSDE-LS. All in all, among the selected algorithms, EPSDE-LS shows the best characteristics in terms of robustness and scalability.

Table 1. Average Error ± Standard Deviation and Wilcoxon Rank-Sum Test (reference =EPSDE-LS) on CEC2013 [15] in 10 dimensions.

	EPSDE-LS	CMAES		MDE-pBX		CCPSO2	
f_1	$\mathbf{0.00e+00 \pm 0.00e+00}$	$\mathbf{0.00e+00 \pm 0.00e+00}$	=	$0.00e+00 \pm 2.27e-14$	=	$3.08e-03 \pm 1.05e-02$	+
f_2	$1.09e+03 \pm 1.15e+03$	$\mathbf{0.00e+00 \pm 0.00e+00}$	-	$2.54e+03 \pm 5.07e+03$	+	$1.80e+06 \pm 1.21e+06$	+
f_3	$6.86e+02 \pm 2.61e+03$	$\mathbf{7.69e-02 \pm 6.40e-01}$	-	$1.41e+05 \pm 1.23e+06$	+	$7.41e+07 \pm 1.12e+08$	+
f_4	$1.64e+01 \pm 1.46e+01$	$\mathbf{0.00e+00 \pm 0.00e+00}$	-	$3.82e+00 \pm 3.15e+01$	-	$1.05e+04 \pm 2.69e+03$	+
f_5	$\mathbf{0.00e+00 \pm 0.00e+00}$	$\mathbf{0.00e+00 \pm 0.00e+00}$	=	$0.00e+00 \pm 7.01e-14$	=	$2.20e-02 \pm 6.13e-02$	+
f_6	$8.05e+00 \pm 3.77e+00$	$6.95e+00 \pm 8.44e+00$	-	$5.70e+00 \pm 4.83e+00$	-	$\mathbf{4.67e+00 \pm 7.85e+00}$	=
f_7	$\mathbf{1.16e+00 \pm 6.98e-01}$	$6.36e+13 \pm 6.32e+14$	+	$7.37e+00 \pm 1.02e+01$	+	$3.99e+01 \pm 1.26e+01$	+
f_8	$\mathbf{2.04e+01 \pm 1.02e-01}$	$2.04e+01 \pm 1.16e-01$	=	$2.05e+01 \pm 9.69e-02$	+	$2.04e+01 \pm 7.48e-02$	+
f_9	$6.00e+00 \pm 1.02e+00$	$1.51e+01 \pm 4.02e+00$	+	$\mathbf{2.16e+00 \pm 1.39e+00}$	-	$5.48e+00 \pm 8.99e-01$	-
f_{10}	$1.44e-01 \pm 8.94e-02$	$\mathbf{1.60e-02 \pm 1.36e-02}$	-	$1.06e-01 \pm 8.03e-02$	-	$1.93e+00 \pm 9.27e-01$	+
f_{11}	$\mathbf{1.31e-10 \pm 5.64e-10}$	$2.56e+02 \pm 2.89e+02$	+	$2.89e+00 \pm 1.72e+00$	+	$2.76e+00 \pm 1.85e+00$	+
f_{12}	$1.13e+01 \pm 4.34e+00$	$3.30e+02 \pm 3.15e+02$	+	$\mathbf{1.02e+01 \pm 4.53e+00}$	-	$3.39e+01 \pm 1.02e+01$	+
f_{13}	$\mathbf{1.55e+01 \pm 6.17e+00}$	$2.29e+02 \pm 2.76e+02$	+	$1.94e+01 \pm 8.85e+00$	+	$4.22e+01 \pm 8.88e+00$	+
f_{14}	$\mathbf{3.90e+01 \pm 3.62e+01}$	$1.78e+03 \pm 4.21e+02$	+	$1.08e+02 \pm 9.77e+01$	+	$8.67e+01 \pm 6.15e+01$	+
f_{15}	$9.43e+02 \pm 2.74e+02$	$1.78e+03 \pm 4.00e+02$	+	$\mathbf{7.56e+02 \pm 2.63e+02}$	-	$1.03e+03 \pm 2.70e+02$	+
f_{16}	$7.49e-01 \pm 2.83e-01$	$\mathbf{3.90e-01 \pm 3.24e-01}$	-	$5.74e-01 \pm 4.62e-01$	-	$1.31e+00 \pm 2.35e-01$	+
f_{17}	$\mathbf{1.03e+01 \pm 1.08e-01}$	$9.74e+02 \pm 3.03e+02$	+	$1.32e+01 \pm 1.92e+00$	+	$1.79e+01 \pm 2.64e+00$	+
f_{18}	$2.32e+01 \pm 5.90e+00$	$1.03e+03 \pm 3.15e+02$	+	$\mathbf{2.02e+01 \pm 5.18e+00}$	-	$5.82e+01 \pm 6.30e+00$	+
f_{19}	$\mathbf{5.43e-01 \pm 1.52e-01}$	$1.18e+00 \pm 4.76e-01$	+	$6.57e-01 \pm 2.22e-01$	+	$1.00e+00 \pm 3.69e-01$	+
f_{20}	$2.99e+00 \pm 3.46e-01$	$4.79e+00 \pm 2.72e-01$	+	$\mathbf{2.73e+00 \pm 6.04e-01}$	-	$3.59e+00 \pm 2.16e-01$	+
f_{21}	$3.80e+02 \pm 6.01e+01$	$3.87e+02 \pm 5.04e+01$	=	$3.98e+02 \pm 1.99e+01$	+	$\mathbf{3.68e+02 \pm 6.68e+01}$	-
f_{22}	$1.73e+02 \pm 5.65e+01$	$2.32e+03 \pm 4.07e+02$	+	$1.77e+02 \pm 1.37e+02$	=	$\mathbf{1.23e+02 \pm 6.60e+01}$	-
f_{23}	$1.08e+03 \pm 2.89e+02$	$2.24e+03 \pm 4.28e+02$	+	$\mathbf{8.43e+02 \pm 3.48e+02}$	-	$1.37e+03 \pm 2.82e+02$	+
f_{24}	$2.11e+02 \pm 1.24e+01$	$3.73e+02 \pm 1.36e+02$	+	$\mathbf{2.05e+02 \pm 5.21e+00}$	-	$2.11e+02 \pm 1.80e+01$	+
f_{25}	$2.12e+02 \pm 5.16e+00$	$2.61e+02 \pm 5.29e+01$	+	$\mathbf{2.01e+02 \pm 8.24e+00}$	-	$2.12e+02 \pm 1.46e+01$	-
f_{26}	$1.83e+02 \pm 3.10e+01$	$2.57e+02 \pm 1.09e+02$	+	$\mathbf{1.40e+02 \pm 4.16e+01}$	-	$1.71e+02 \pm 2.37e+01$	-
f_{27}	$4.87e+02 \pm 5.37e+01$	$4.01e+02 \pm 9.94e+01$	-	$\mathbf{3.04e+02 \pm 1.72e+01}$	-	$4.33e+02 \pm 5.71e+01$	-
f_{28}	$\mathbf{2.96e+02 \pm 2.80e+01}$	$1.22e+03 \pm 1.13e+03$	+	$3.04e+02 \pm 5.53e+01$	+	$4.01e+02 \pm 1.63e+02$	+

In order to give a further insight into the results presented above, we ranked the algorithms under study by means of the Holm-Bonferroni procedure [9], as described in [6], with level of confidence set to 0.05. For the sake of completeness, in this analysis we included also the original EPSDE algorithm [16] (whose results are not reported in Tables 1-4 due to space limitations), executed with the same setting of EPSDE-LS but without pool of local search. Table 5 displays the ranks, z_j values, p_j values, and corresponding δ/j obtained in this way. The rank of EPSDE-LS is shown in parentheses in the table caption. Moreover, we indicate whether the null-hypothesis (that the two algorithms have indistinguishable performances) is "Rejected", i.e. EPSDE-LS statistically outperforms the algorithm under consideration, or "Accepted" if the distribution of values can be considered the same (there is no statistic out-performance). It can be seen that the proposed EPSDE-LS has the highest rank amongst all the algorithms considered in this study. It can be observed also that the null-hypothesis is rejected in all the cases, i.e. the global performance of EPSDE-LS over the two benchmarks is superior to the global performance of all the other algorithms considered in this study.

Table 2. Average Error ± Standard Deviation and Wilcoxon Rank-Sum Test (reference =EPSDE-LS) on CEC2013 [15] in 30 dimensions.

	EPSDE-LS	CMAES		MDE-pBX		CCPSO2	
f_1	$\mathbf{0.00e+00 \pm 0.00e+00}$	$0.00e+00 \pm 1.18e-13$	=	$2.27e-13 \pm 4.86e-13$	+	$1.36e-12 \pm 6.01e-12$	+
f_2	$1.68e+06 \pm 8.26e+05$	$\mathbf{0.00e+00 \pm 1.54e-13}$	-	$2.70e+05 \pm 2.62e+05$	-	$2.14e+06 \pm 1.04e+06$	+
f_3	$1.03e+06 \pm 2.13e+06$	$\mathbf{9.24e+01 \pm 4.00e+02}$	-	$5.19e+07 \pm 1.18e+08$	+	$1.13e+09 \pm 1.18e+09$	+
f_4	$2.17e+04 \pm 5.55e+03$	$\mathbf{0.00e+00 \pm 1.29e-13}$	-	$3.49e+02 \pm 3.18e+02$	-	$5.64e+04 \pm 2.09e+04$	+
f_5	$\mathbf{0.00e+00 \pm 3.01e-14}$	$9.09e-13 \pm 2.46e-12$	+	$1.09e-10 \pm 1.00e-09$	+	$3.04e-07 \pm 8.74e-07$	+
f_6	$1.05e+01 \pm 5.75e+00$	$\mathbf{4.83e+00 \pm 1.28e+01}$	-	$3.41e+01 \pm 2.77e+01$	+	$3.44e+01 \pm 2.78e+01$	+
f_7	$\mathbf{2.78e+01 \pm 9.88e+00}$	$3.51e+08 \pm 3.49e+09$	+	$5.61e+01 \pm 1.90e+01$	+	$1.19e+02 \pm 2.33e+01$	+
f_8	$\mathbf{2.10e+01 \pm 6.30e-02}$	$2.10e+01 \pm 5.49e-02$	+	$2.10e+01 \pm 5.93e-02$	+	$2.10e+01 \pm 5.44e-02$	+
f_9	$3.14e+01 \pm 1.65e+00$	$4.42e+01 \pm 7.09e+00$	+	$\mathbf{2.16e+01 \pm 4.36e+00}$	-	$3.02e+01 \pm 2.20e+00$	-
f_{10}	$2.60e-02 \pm 1.60e-02$	$\mathbf{2.01e-02 \pm 1.71e-02}$	-	$1.81e-01 \pm 1.10e-01$	+	$2.00e-01 \pm 9.45e-02$	+
f_{11}	$\mathbf{2.18e-03 \pm 3.82e-03}$	$1.05e+02 \pm 2.55e+02$	+	$4.68e+01 \pm 1.54e+01$	+	$5.76e-01 \pm 6.49e-01$	+
f_{12}	$\mathbf{6.63e+01 \pm 2.02e+01}$	$8.08e+02 \pm 9.37e+02$	=	$6.91e+01 \pm 2.20e+01$	=	$2.13e+02 \pm 5.62e+01$	+
f_{13}	$\mathbf{1.06e+02 \pm 2.51e+01}$	$1.65e+03 \pm 1.67e+03$	+	$1.50e+02 \pm 3.56e+01$	+	$2.58e+02 \pm 4.39e+01$	+
f_{14}	$5.69e+02 \pm 2.06e+02$	$5.39e+03 \pm 7.64e+02$	+	$1.20e+03 \pm 4.25e+02$	+	$\mathbf{6.57e+00 \pm 3.69e+00}$	-
f_{15}	$4.71e+03 \pm 8.51e+02$	$5.29e+03 \pm 6.36e+02$	+	$\mathbf{4.01e+03 \pm 7.00e+02}$	-	$4.03e+03 \pm 4.77e+02$	-
f_{16}	$1.63e+00 \pm 4.89e-01$	$\mathbf{1.23e-01 \pm 1.06e-01}$	-	$1.32e+00 \pm 8.61e-01$	-	$2.40e+00 \pm 4.03e-01$	+
f_{17}	$3.27e+01 \pm 7.89e-01$	$4.07e+03 \pm 8.51e+02$	+	$6.89e+01 \pm 1.24e+01$	+	$\mathbf{3.13e+01 \pm 4.89e-01}$	-
f_{18}	$8.87e+01 \pm 2.31e+01$	$3.95e+03 \pm 7.79e+02$	+	$\mathbf{8.31e+01 \pm 1.66e+01}$	=	$2.44e+02 \pm 5.78e+01$	+
f_{19}	$2.46e+00 \pm 5.06e-01$	$3.50e+00 \pm 9.05e-01$	+	$9.10e+00 \pm 4.94e+00$	+	$\mathbf{8.55e-01 \pm 1.71e-01}$	-
f_{20}	$1.19e+01 \pm 5.30e-01$	$1.50e+01 \pm 4.97e-02$	+	$\mathbf{1.09e+01 \pm 7.97e-01}$	-	$1.39e+01 \pm 4.52e-01$	+
f_{21}	$3.00e+02 \pm 7.94e+01$	$3.09e+02 \pm 8.58e+01$	+	$3.09e+02 \pm 7.63e+01$	+	$\mathbf{2.58e+02 \pm 7.21e+01}$	=
f_{22}	$7.70e+02 \pm 2.52e+02$	$6.92e+03 \pm 9.35e+02$	+	$1.11e+03 \pm 5.46e+02$	+	$\mathbf{1.21e+02 \pm 7.28e+01}$	-
f_{23}	$4.86e+03 \pm 6.99e+02$	$6.78e+03 \pm 7.36e+02$	+	$\mathbf{4.47e+03 \pm 7.32e+02}$	-	$5.26e+03 \pm 7.22e+02$	+
f_{24}	$2.79e+02 \pm 7.17e+00$	$7.93e+02 \pm 5.89e+02$	+	$\mathbf{2.31e+02 \pm 1.11e+01}$	-	$2.81e+02 \pm 1.08e+01$	+
f_{25}	$2.92e+02 \pm 5.00e+00$	$3.81e+02 \pm 1.54e+02$	+	$\mathbf{2.75e+02 \pm 1.55e+01}$	-	$3.03e+02 \pm 6.25e+00$	+
f_{26}	$2.16e+02 \pm 5.13e+01$	$4.66e+02 \pm 4.25e+02$	+	$2.16e+02 \pm 4.31e+01$	-	$\mathbf{2.02e+02 \pm 4.53e+00}$	-
f_{27}	$1.09e+03 \pm 4.99e+01$	$8.17e+02 \pm 2.09e+02$	-	$\mathbf{6.55e+02 \pm 1.13e+02}$	-	$1.07e+03 \pm 1.13e+02$	=
f_{28}	$\mathbf{3.00e+02 \pm 2.16e-13}$	$1.94e+03 \pm 3.38e+03$	+	$3.11e+02 \pm 1.11e+02$	+	$5.43e+02 \pm 5.77e+02$	+

Table 3. Average Error ± Standard Deviation and Wilcoxon Rank-Sum Test (reference =EPSDE-LS) on CEC2013 [15] in 50 dimensions.

	EPSDE-LS	CMAES		MDE-pBX		CCPSO2	
f_1	$\mathbf{0.00e+00 \pm 0.00e+00}$	$2.27e-13 \pm 0.00e+00$	+	$3.32e-11 \pm 2.60e-10$	+	$7.05e-12 \pm 3.53e-11$	+
f_2	$6.61e+06 \pm 2.72e+06$	$\mathbf{2.27e-13 \pm 0.00e+00}$	-	$9.06e+05 \pm 4.90e+05$	-	$4.37e+06 \pm 2.29e+06$	-
f_3	$7.35e+06 \pm 1.79e+07$	$\mathbf{2.32e+04 \pm 9.57e+04}$	-	$1.42e+08 \pm 1.57e+08$	+	$3.09e+09 \pm 3.03e+09$	+
f_4	$5.51e+04 \pm 9.43e+03$	$\mathbf{2.27e-13 \pm 0.00e+00}$	-	$1.09e+03 \pm 8.33e+02$	-	$1.08e+05 \pm 3.86e+04$	+
f_5	$\mathbf{1.14e-13 \pm 1.97e-14}$	$1.95e-09 \pm 9.17e-10$	+	$2.54e-05 \pm 2.52e-04$	+	$3.92e-04 \pm 3.89e-03$	+
f_6	$4.36e+01 \pm 9.69e-01$	$\mathbf{4.29e+01 \pm 5.98e+00}$	-	$5.67e+01 \pm 2.24e+01$	+	$4.74e+01 \pm 1.34e+01$	+
f_7	$7.50e+01 \pm 1.59e+01$	$1.98e+04 \pm 1.96e+05$	+	$\mathbf{6.81e+01 \pm 1.22e+01}$	-	$1.43e+02 \pm 2.39e+01$	+
f_8	$2.12e+01 \pm 3.93e-02$	$\mathbf{2.11e+01 \pm 3.75e-02}$	=	$2.12e+01 \pm 4.36e-02$	+	$2.12e+01 \pm 3.86e-02$	=
f_9	$6.06e+01 \pm 2.04e+00$	$7.66e+01 \pm 8.71e+00$	+	$\mathbf{4.27e+01 \pm 6.99e+00}$	-	$5.87e+01 \pm 3.26e+00$	-
f_{10}	$5.21e-02 \pm 4.01e-02$	$\mathbf{2.70e-02 \pm 1.55e-02}$	-	$4.09e-01 \pm 5.57e-01$	+	$2.03e-01 \pm 1.80e-01$	+
f_{11}	$\mathbf{1.86e-01 \pm 2.65e-01}$	$2.46e+02 \pm 5.29e+02$	+	$1.21e+02 \pm 2.97e+01$	+	$9.07e-01 \pm 8.53e-01$	+
f_{12}	$\mathbf{1.53e+02 \pm 3.83e+01}$	$2.28e+03 \pm 1.53e+03$	+	$1.62e+02 \pm 3.45e+01$	=	$4.55e+02 \pm 8.03e+01$	+
f_{13}	$\mathbf{2.44e+02 \pm 4.17e+01}$	$3.26e+03 \pm 1.25e+03$	+	$3.22e+02 \pm 5.39e+01$	+	$5.69e+02 \pm 8.18e+01$	+
f_{14}	$7.52e+02 \pm 2.41e+02$	$8.74e+03 \pm 1.05e+03$	+	$2.79e+03 \pm 8.06e+02$	+	$\mathbf{7.35e+00 \pm 3.55e+00}$	-
f_{15}	$9.07e+03 \pm 1.21e+03$	$9.04e+03 \pm 8.70e+02$	=	$\mathbf{7.58e+03 \pm 8.01e+02}$	-	$8.31e+03 \pm 8.71e+02$	-
f_{16}	$2.24e+00 \pm 5.52e-01$	$\mathbf{8.00e-02 \pm 4.27e-02}$	-	$1.93e+00 \pm 8.76e-01$	-	$2.75e+00 \pm 5.96e-01$	+
f_{17}	$5.66e+01 \pm 1.60e+00$	$6.84e+03 \pm 1.10e+03$	+	$1.79e+02 \pm 3.56e+01$	+	$\mathbf{5.16e+01 \pm 3.28e-01}$	-
f_{18}	$1.96e+02 \pm 6.47e+01$	$7.01e+03 \pm 9.83e+02$	+	$\mathbf{1.86e+02 \pm 3.17e+01}$	=	$4.87e+02 \pm 9.77e+01$	+
f_{19}	$4.71e+00 \pm 8.61e-01$	$6.26e+00 \pm 1.54e+00$	+	$3.94e+01 \pm 2.10e+01$	+	$\mathbf{1.49e+00 \pm 2.32e-01}$	-
f_{20}	$2.15e+01 \pm 6.05e-01$	$2.50e+01 \pm 9.74e-02$	+	$\mathbf{2.01e+01 \pm 9.17e-01}$	-	$2.33e+01 \pm 8.19e-01$	+
f_{21}	$6.20e+02 \pm 4.37e+02$	$7.95e+02 \pm 3.57e+02$	+	$8.91e+02 \pm 3.44e+02$	+	$\mathbf{4.42e+02 \pm 3.45e+02}$	=
f_{22}	$9.07e+02 \pm 3.04e+02$	$1.18e+04 \pm 1.34e+03$	+	$3.22e+03 \pm 1.06e+03$	+	$\mathbf{1.11e+02 \pm 9.60e+01}$	-
f_{23}	$9.37e+03 \pm 1.26e+03$	$1.18e+04 \pm 9.41e+02$	+	$\mathbf{9.08e+03 \pm 1.05e+03}$	=	$1.09e+04 \pm 1.34e+03$	+
f_{24}	$3.54e+02 \pm 7.63e+00$	$1.74e+03 \pm 1.02e+03$	+	$\mathbf{2.88e+02 \pm 1.56e+01}$	-	$3.60e+02 \pm 9.64e+00$	+
f_{25}	$3.81e+02 \pm 6.60e+00$	$5.07e+02 \pm 2.06e+02$	+	$\mathbf{3.68e+02 \pm 1.48e+01}$	-	$3.97e+02 \pm 1.08e+01$	+
f_{26}	$3.99e+02 \pm 1.02e+02$	$7.71e+02 \pm 8.75e+02$	+	$3.55e+02 \pm 7.46e+01$	-	$\mathbf{2.15e+02 \pm 4.95e+01}$	-
f_{27}	$1.86e+03 \pm 5.99e+01$	$1.32e+03 \pm 3.23e+02$	-	$\mathbf{1.23e+03 \pm 1.49e+02}$	-	$1.82e+03 \pm 8.56e+01$	-
f_{28}	$6.49e+02 \pm 8.45e+02$	$2.80e+03 \pm 4.35e+03$	+	$\mathbf{5.05e+02 \pm 5.99e+02}$	-	$7.24e+02 \pm 1.08e+03$	+

4 Conclusions

This paper proposes a Memetic Computing structure composed of a DE framework, which makes use of an ensemble of crossover/mutation strategies and parameters, and a pool of three local search methods.

Table 4. Average Error ± Standard Deviation and Wilcoxon Rank-Sum Test (reference =EPSDE-LS) on CEC2010 [29] in 1000 dimensions.

	EPSDE-LS	CMAES		MDE-pBX		CCPSO2	
f_1	$1.99e+02 \pm 9.10e+02$	$6.95e+04 \pm 9.91e+03$	+	$1.05e+09 \pm 6.58e+08$	+	$\mathbf{6.47e-14 \pm 1.41e-13}$	-
f_2	$3.70e+02 \pm 8.88e+01$	$1.01e+04 \pm 4.63e+02$	+	$7.02e+03 \pm 2.38e+02$	+	$\mathbf{1.36e+02 \pm 1.11e+02}$	-
f_3	$1.04e+01 \pm 1.32e+00$	$1.99e+01 \pm 1.12e-02$	+	$1.93e+01 \pm 4.76e-02$	+	$\mathbf{7.34e-11 \pm 1.05e-10}$	-
f_4	$3.93e+11 \pm 1.78e+11$	$\mathbf{5.55e+10 \pm 4.75e+09}$	-	$3.21e+12 \pm 9.76e+11$	+	$2.14e+12 \pm 1.27e+12$	+
f_5	$\mathbf{7.97e+07 \pm 1.29e+07}$	$6.65e+08 \pm 1.19e+08$	+	$1.54e+08 \pm 2.77e+07$	+	$3.92e+08 \pm 7.98e+07$	+
f_6	$\mathbf{1.93e+01 \pm 1.73e-01}$	$1.98e+07 \pm 5.87e+04$	+	$3.65e+06 \pm 1.75e+06$	+	$1.71e+07 \pm 4.45e+06$	+
f_7	$\mathbf{3.24e+03 \pm 1.88e+04}$	$3.08e+06 \pm 2.04e+05$	+	$6.79e+06 \pm 1.01e+07$	+	$7.60e+09 \pm 9.72e+09$	+
f_8	$3.47e+07 \pm 2.14e+07$	$\mathbf{4.44e+06 \pm 3.21e+05}$	-	$2.03e+08 \pm 1.63e+08$	+	$5.46e+07 \pm 4.16e+07$	+
f_9	$5.77e+07 \pm 1.96e+07$	$\mathbf{7.27e+04 \pm 1.07e+04}$	-	$1.68e+09 \pm 1.00e+09$	+	$5.01e+07 \pm 7.68e+06$	-
f_{10}	$4.77e+03 \pm 1.76e+02$	$1.03e+04 \pm 4.04e+02$	+	$7.33e+03 \pm 2.55e+02$	+	$\mathbf{4.57e+03 \pm 2.75e+02}$	-
f_{11}	$\mathbf{1.53e+02 \pm 1.65e+01}$	$2.18e+02 \pm 1.77e-01$	+	$2.06e+02 \pm 2.40e+00$	+	$2.00e+02 \pm 5.98e+00$	+
f_{12}	$2.70e+04 \pm 5.91e+03$	$\mathbf{1.64e-19 \pm 4.18e-20}$	-	$2.92e+05 \pm 6.60e+04$	+	$6.12e+04 \pm 8.14e+04$	+
f_{13}	$1.45e+03 \pm 8.16e+02$	$\mathbf{4.53e+01 \pm 6.59e+01}$	-	$2.88e+09 \pm 3.17e+09$	+	$1.14e+03 \pm 5.42e+02$	-
f_{14}	$2.83e+08 \pm 2.44e+07$	$\mathbf{7.69e+04 \pm 1.06e+04}$	-	$1.04e+09 \pm 1.97e+08$	+	$1.60e+08 \pm 3.35e+07$	-
f_{15}	$9.12e+03 \pm 4.17e+02$	$1.04e+04 \pm 5.58e+02$	+	$\mathbf{7.44e+03 \pm 2.80e+02}$	-	$9.31e+03 \pm 5.52e+02$	+
f_{16}	$3.99e+02 \pm 2.34e+00$	$3.97e+02 \pm 2.92e-01$	-	$\mathbf{3.84e+02 \pm 1.22e+00}$	-	$3.95e+02 \pm 1.45e+00$	-
f_{17}	$1.36e+05 \pm 1.42e+04$	$\mathbf{4.17e-19 \pm 7.23e-20}$	-	$4.35e+05 \pm 8.33e+04$	+	$1.41e+05 \pm 1.44e+05$	+
f_{18}	$9.01e+04 \pm 4.09e+05$	$\mathbf{1.59e+02 \pm 1.67e+02}$	-	$3.73e+10 \pm 1.95e+10$	+	$5.62e+03 \pm 4.13e+03$	-
f_{19}	$2.95e+06 \pm 1.90e+05$	$\mathbf{3.38e+01 \pm 1.36e+01}$	-	$9.22e+05 \pm 1.06e+05$	-	$1.14e+06 \pm 1.22e+06$	-
f_{20}	$1.74e+04 \pm 4.61e+04$	$\mathbf{7.51e+02 \pm 9.99e+01}$	-	$4.18e+10 \pm 2.02e+10$	+	$1.42e+03 \pm 1.19e+02$	-

Table 5. Holm test on the Fitness, reference algorithm = EPSDE-LS (Rank = 3.50e+00)

j	Optimizer	$Rank$	z_j	p_j	δ/j	Hypothesis
1	MDE-pBX	3.08e+00	-2.36e+00	9.06e-03	5.00e-02	Rejected
2	EPSDE	2.98e+00	-2.90e+00	1.86e-03	2.50e-02	Rejected
3	CCPSO2	2.79e+00	-3.97e+00	3.53e-05	1.67e-02	Rejected
4	CMAES	2.47e+00	-5.75e+00	4.55e-09	1.25e-02	Rejected

The ensemble is a simple and efficient self-adaptive technique that allows the successful strategies to be propagated in the future generations while blocking the propagation of unsuccessful strategies. This framework is empowered by a pool of three local search algorithms whose activation is coordinated by a randomized criterion. These three local search algorithms are Nelder-Mead, Powell, and Rosenbrock algorithms. The proposed algorithm has been tested over a diverse testbed in various dimensions ranging from 10 to 1000 and compared against modern meta-heuristics representing the state-of-the-art in optimization. The EPSDE-LS should be considered as a first successful attempt to extend the concept of ensemble to structures composed of multiple local search operators. This algorithmic design has been performed by following the philosophy of Memetic Computing and the simplistic combination of its operators has been inspired by the Ockham's Razor principle applied to algorithmic design. Despite its simplicity in the meme coordination, the resulting algorithm displays a great ability to adapt to diverse fitness landscapes, thus proving a powerful tool for addressing complex optimization problems.

Acknowledgements

INCAS[3] is co-funded by the Province of Drenthe, the Municipality of Assen, the European Fund for Regional Development and the Ministry of Economic Affairs, Peaks in the Delta. The numerical experiments have been carried out on the network of the De Montfort University with the software for distributed optimization Kimeme [3].

References

1. Brest, J., Greiner, S., Bošković, B., Mernik, M., Žumer, V.: Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. IEEE Transactions on Evolutionary Computation 10(6), 646–657 (2006)
2. Caraffini, F., Neri, F., Iacca, G., Mol, A.: Parallel memetic structures. Information Sciences 227(0), 60 – 82 (2013)
3. Cyber Dyne Srl Home Page: Kimeme (2013), `http://cyberdynesoft.it/`
4. Das, S., Suganthan, P.: Differential Evolution: A Survey of the State-of-the-Art. Evolutionary Computation, IEEE Transactions on 15(1), 4–31 (feb 2011)
5. Das, S., Abraham, A., Chakraborty, U.K., Konar, A.: Differential Evolution with a Neighborhood-based Mutation Operator. IEEE Transactions on Evolutionary Computation 13(3), 526–553 (2009)
6. Garcia, S., Fernandez, A., Luengo, J., Herrera, F.: A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Computing 13(10), 959–977 (2008)
7. Hansen, N.: The CMA Evolution Strategy (2012), `http://www.lri.fr/~hansen/cmaesintro.html`
8. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). Evolutionary Computation 11(1), 1–18 (2003)
9. Holm, S.: A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 6(2), 65–70 (1979)
10. Iacca, G., Mallipeddi, R., Mininno, E., Neri, F., Suganthan, P.N.: Super-fit and Population Size Reduction Mechanisms in Compact Differential Evolution. In: Proceedings of IEEE Symposium on Memetic Computing. pp. 21–28 (2011)
11. Iacca, G., Neri, F., Mininno, E., Ong, Y.S., Lim, M.H.: Ockham's Razor in Memetic Computing: Three Stage Optimal Memetic Exploration. Information Sciences 188, 17–43 (2012)
12. Iacca, G., Caraffini, F., Neri, F.: Multi-strategy coevolving aging particle optimization. International Journal of Neural Systems 24(01), 1450008 (2014)
13. Islam, S., Das, S., Ghosh, S., Roy, S., Suganthan, P.: An Adaptive Differential Evolution Algorithm With Novel Mutation and Crossover Strategies for Global Numerical Optimization. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 42(2), 482–500 (april 2012)
14. Li, X., Yao, X.: Cooperatively Coevolving Particle Swarms for Large Scale Optimization. Evolutionary Computation, IEEE Transactions on 16(2), 210–224 (april 2012)
15. Liang, J.J., Qu, B.Y., Suganthan, P.N., Hernndez-Daz, A.G.: Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization. Tech. Rep. 201212, Zhengzhou University, Zhengzhou, China (2013)
16. Mallipeddi, R., Suganthan, P.N., Pan, Q.K., Tasgetiren, M.F.: Differential evolution algorithm with ensemble of parameters and mutation strategies. Applied Soft Computing 11(2), 1679–1696 (2011), the Impact of Soft Computing for the Progress of Artificial Intelligence
17. Mezura-Montes, E., Velazquez-Reyes, J., Coello Coello, C.: Modified differential evolution for constrained optimization. In: IEEE Congress on Evolutionary Computation, 2006. pp. 25–32 (2006)

18. Nelder, A., Mead, R.: A simplex method for function optimization. Computation Journal Vol 7, 308–313 (1965)
19. Neri, F., Iacca, G., Mininno, E.: Disturbed Exploitation compact Differential Evolution for Limited Memory Optimization Problems. Information Sciences 181(12), 2469–2487 (2011)
20. Neri, F., Tirronen, V.: On Memetic Differential Evolution Frameworks: a Study of Advantages and Limitations in Hybridization. In: Proceedings of the IEEE World Congress on Computational Intelligence. pp. 2135–2142 (2008)
21. Neri, F., Tirronen, V.: Recent Advances in Differential Evolution: A Review and Experimental Analysis. Artificial Intelligence Review 33(1–2), 61–106 (2010)
22. Powell, M.J.D.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. The Computer Journal 7(2), 155–162 (Jan 1964)
23. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes in C. Cambridge University Press, Cambridge, UK, 2nd edn. (1992)
24. Price, K., Storn, R.: Differential evolution: A simple evolution strategy for fast optimization. Dr. Dobb's J. Software Tools 22(4), 18–24 (1997)
25. Price, K.: An Introduction to Differential Evolution. In: Corne, D., Dorigo, M., Glover, F., Dasgupta, D., Moscato, P., Poli, R., Price, K.V. (eds.) New Ideas in Optimization, pp. 79–108. McGraw-Hill (1999)
26. Price, K.V., Storn, R., Lampinen, J.: Differential Evolution: A Practical Approach to Global Optimization. Springer (2005)
27. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. IEEE Transactions on Evolutionary Computation 13(2), 398–417 (2009)
28. Rosenbrock, H.H.: An automatic Method for finding the greatest or least Value of a Function. The Computer Journal 3(3), 175–184 (1960)
29. Tang, K., Li, X., Suganthan, P.N., Yang, Z., Weise, T.: Benchmark Functions for the CEC'2010 Special Session and Competition on Large-Scale Global Optimization. Tech. rep., University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL): Hefei, Anhui, China (2010)
30. Tirronen, V., Neri, F., Kärkkäinen, T., Majava, K., Rossi, T.: An Enhanced Memetic Differential Evolution in Filter Design for Defect Detection in Paper Production. Evolutionary Computation 16(4), 529–555 (2008)
31. Weber, M., Neri, F., Tirronen, V.: A Study on Scale Factor/Crossover Interaction in Distributed Differential Evolution. Artificial Intelligence Review 39(3), 195–224 (2013)
32. Wessing, S., Preuss, M., Rudolph, G.: When parameter tuning actually is parameter control. In: Proceesings of the Conference on Genetic and Evolutionary Computation. pp. 821–828. ACM (2011)
33. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin 1(6), 80–83 (1945)
34. Zaharie, D.: Control of population diversity and adaptation in differential evolution algorithms. In: Matousek, D., Osmera, P. (eds.) Proceedings of MENDEL International Conference on Soft Computing. pp. 41–46 (2003)
35. Zaharie, D.: Influence of crossover on the behavior of differential evolution algorithms. Appl. Soft Comput. 9(3), 1126–1138 (Jun 2009)
36. Zhang, J., Sanderson, A.: Jade: Adaptive differential evolution with optional external archive. Evolutionary Computation, IEEE Transactions on 13(5), 945–958 (2009)