2013 IEEE Congress on Evolutionary Computation
June 20-23, Cancún, México

Differential Evolution with Concurrent Fitness Based Local Search

Ilpo Poikolainen
Department of Mathematical Information Technology
University of Jyväskylä
Jyväskylä, Finland
Email: ilpo.poikolainen@jyu.fi

Ferrante Neri
Department of Mathematical Information Technology
University of Jyväskylä
Jyväskylä, Finland
Email: ferrante.neri@jyu.fi

Centre for Computational Intelligence,
School of Computer Science and Informatics, De Montfort University,
The Gateway, Leicester LE1 9BH, United Kingdom
Email: fneri@dmu.ac.uk

***Abstract*—This paper proposes a novel implementation of memetic structure for continuous optimization problems. The proposed algorithm, namely Differential Evolution with Concurrent Fitness Based Local Search (DEcfbLS), enhances the DE performance by including a local search concurrently applied on multiple individuals of the population. The selection of the individuals undergoing local search is based on a fitness-based adaptive rule. The most promising individuals are rewarded with a local search operator that moves along the axes and complements the normal search moves of DE structure. The application of local search is performed with a shallow termination rule. This design has been performed in order to overcome the limitations within the search logic on the original DE algorithm. The proposed algorithm has been tested on various problems in multiple dimensions. Numerical results show that the proposed algorithm is promising candidate to take part to competition on Real-Parameter Single Objective Optimization at CEC-2013. A comparison against modern meta-heuristics confirms that the proposed algorithm robustly displays a good performance on the testbed under consideration.**

I. Introduction

Differential Evolution (DE), see [1], is a popular and reliable optimizer that shows a robust performance across various continuous optimization problems. DE employs a population of solutions and generates its offspring by means of the scaled difference of two (or more) individuals of the population. The survivor selection is made between the parent individual and offspring by following the so called one-to-one spawning: the offspring directly replaces the parent that generated it when the offspring outperforms the parent. Thanks to its simplicity features and performance, after its original definition, DE has become popular among both researches and practitioners. Some examples of successful DE applications can be found in [1] and [2]. Practical applications include multisensor fusion problem, see [3], aerodynamic design, see [4], filter design, see [5] and many other engineering applications. Survey papers on DE framework and its variants are reported in [6] and [7]. The reasons behind the DE success are listed and analysed by the following points.

1) DE has rather simple implementation with respect to other computational intelligence optimization algorithms while still having very high performance on complex fitness landscapes including uni-modal, multi-modal, separable, non-separable etc, see [8].
2) Control parameters of DE are relatively few (only three classical DE), see [9]. Although a proper selection of these parameters is essential for guaranteeing a good performance on a given problem, a reasonably quick parameter tuning appears to often allow a good performance, see [10]. However, due to the dynamic nature of evolution/optimization, modern DE schemes often propose a dynamic/adaptive parameter setting or employment of multiple search rules. Some popular examples are given in the self-adaptive (and randomized) DE version proposed in [11], and in the controlled parameter randomization proposed in [12].
3) The computational complexity of DE is relatively low, thus making DE schemes flexibly applicable in large scale optimization problem. This advantage is especially evident when we consider the features of algorithms based on a covariance matrix adaptation, see e.g. [13]. Although the latter algorithms are very efficient (thanks to their theoretical foundations) in low dimensions, they might lead to an unacceptable computational cost (due to calculations involving square matrices with size equal to number of variables) for large scale problems.

Although DE is capable at displaying a high performance, this algorithm is characterized by some limitations, see [7]. If the algorithm does not manage to generate an offspring which outperforms the corresponding parent solution, the search is likely to be jeopardized and result into the undesired stagnation condition. If the search is repeated over and over again with similar step sizes without succeeding to improve upon the candidate solutions, DE can display a poor performance, see [14]. As shown in the analysis reported in [7], DE scheme is characterized by a limited amount of search moves. Some countermeasure to mitigate this issue include a randomization of parameters, see e.g. [12], [11], and [15], progressive population size reduction, see [16], and explicit inclusion of local search operators, see e.g. [17], [18], and [19].

978-1-4799-0454-9/13/$31.00 ©2013 IEEE

Modern DE-based algorithms can thus be divided into the two following categories, see [7]:

1) DE integrating an extra component This class includes those algorithms that use DE as an evolutionary framework assisted by other algorithmic components, e.g., local searchers or extra operators (see [20] [17] and [19]). The algorithms belonging to this category can be decomposed as DE framework and additional components
2) Modified strutures of DE This class includes those algorithms which modify the DE structure, search logic, selection etc. Some examples of such modifications are given in articles [11], [21], and [22]

In this paper we use integration of local searcher as extra component with DE to overcome possible problems of stagnation and to improve exploitative capabilities of DE algorithm. The proposed algorithm, at first, samples a population of candidate solutions within a decision space. Then, each population individual is refined with a a local search by using a small local budget. Small local budget appears very important for achieving some initial improvements without excessively biasing the search towards local optima. After this initialization, the main components a DE framework cooperates with a local search to enhance upon the performance of the candidate solutions. While DE is a population based component, deterministic Local Search (LS) is a single point optimizer which is performed on multiple population members on each activation.

The remaining of this paper is organized as follows. Section II describes the algorithmic structure of DEcfbLS and its components. Section III shows the experimental setup and numerical results of the study. Section IV gives the conclusions of this work.

II. Differential Evolution with Concurrent Fitness Based Local Search

In order to clarify the notation used in this paper, we refer to the minimization problem of an objective function $f(x)$, where a candidate solution x is a vector of n design variables in decision space D. This section describes the algorithmic framework of DEcfbLS by analysing the three different components that compose it:

1) Local Search Initialization
2) Differential Evolution
3) Deterministic Local Search

A. Local Search Initialization

At the beginning of optimization process, the population is randomly initialized (with uniform distribution) within the search space D. In our implementation we have set population size NP equal to 30 individuals. After the initialization, we perform a LS with very small local budget (4 iterations) over all the individuals of the initial population, see Algorithm 2.

B. Differential Evolution

After initialization and refining, DE is activated over the individuals of the population, see e.g. [23] and [1]. The working principles of this framework are here described. At each generation, for each solution x_i, three other solutions x_t, x_s and x_t are randomly selected from population. Then, a provisional offspring x_o is generated by mutation:

$$x_o = x_t + F(x_r - x_s), \tag{1}$$

where F is a parameter namely scale factor. Scale factor is a positive value tat usually is selected to be greater than 1. In our implementation we have set the scale factor F to be equal to 0.7. The mutation scheme shown in ed. (1) is known as DE/rand/1. Many other mutation schemes have been proposed in the literature as well as self-adaptive schemes that make use of multiple mutation strategies, for example see [11] and [6].

When the provisional offspring has been generated by mutation, the so-called exponential crossover is applied to combine the genes of the parent solution x_i and provisional offspring x_o. At first a copy of the parent solution x_i is performed. Then, one design variable is chosen at random and copied from the provisional offspring (also referred as mutant) to the corresponding position of x_i (its copy more precisely) to make sure atleast one parameter is exchanged. The source of subsequent trial parameters is determined by comparing predefined crossover rate Cr to uniformly distributed random number between 0 and 1 that is generated anew for each parameter, i.e, $rand_j(0,1)$. As long as $rand_j(0.1) \leq Cr$, parameters continue to be taken from mutant x_o. As soon as $rand_j(0,1) > Cr$, the copy process is interrupted. Thus, the current and all remaining parameters of the final offspring x_{off} are those from the parent solution x_i. The pseudo-code of the exponential crossover is shown in Algorithm 1.

Algorithm 1 Exponential crossover

```
x_off = x_i
generate j = round(n · rand(0,1))
x_off[j] = x_o[j]
k = 1
while rand(0,1) ≤ Cr AND k < n do
    x_off[j] = x_o[j]
    j = j + 1
    if j == n then
        j = 1
    end if
    k = k + 1
end while
```

In other words, this crossover operator generates offspring composed of the parent x_i and contains, within it, a section of the chromosome of the mutant vector x_o. According to its original definition, see e.g. [1], for a fixed Cr value the exploration feature of the crossover operator is dependant on the dimensionality of the problem. For example, if Cr is prearranged in a low dimensional problem, the offspring is composed mainly of the mutant vector (provisional offspring), while for a high dimensional problem, the offspring is mainly composed of the parent. In this study, we propose to slightly modify the definition of exponential crossover by fixing, instead of Cr, the approximate proportion of mutant genes within the offspring. Let us define this proportion, namely inheritance factor, as

$$\alpha_e \approx \frac{n_e}{n} \tag{2}$$

where n_e is the number of mutant genes we expect to copy from x_o into x_{off} in addition to the gene deterministically copied. In order to achieve that on average n_e are copied into the offspring we need to impose that

$$Cr^{n\alpha_e} = 0.5. \tag{3}$$

It can easily be seen that, for a chosen α_e, the crossover rate can be set on the basis of the dimensionality in the following way, see [24] for details:

$$Cr = \frac{1}{\sqrt[n\alpha_e]{2}}. \tag{4}$$

C. Deterministic local search

The deterministic local search is a single solution algorithm that attempts to exploit promising solutions by performing steps along the axis. This LS has been effectively used in [25], [26], and [27]. The search logic is similar to that of Hooke-Jeeves algorithm, see [28]. The purpose of this search algorithm is to exploit promising basins of attraction. The algorithm searches each design variable by performing steps along the axis and if the solution does not improve the search step-size is halved for the following iteration. More precisely, for each dimension j, the algorithm samples $x[j] - \rho$, where $x[j]$ is the j^{th} design variable of the selected point x on which LS is performed. If the perturbed solution is better than x, it replaces the old solution and moves to the next dimension $j+1$. If there is no out-performance, the search looks towards the opposite direction with a step-size $\rho/2$ before moving towards the following dimension. After having gone through each dimension if there was no improvement the step-size ρ is halved before subsequent iteration. For the sake of clarity, the pseudo-code of the proposed LS is given in Algorithm 2.

Algorithm 2 Determinisitic local search

```
while local budget condition do
    x_t = x_e
    x_s = x_e
    for i = 1 : n do
        x_s[i] = x_e[i] − ρ
        if f(x_s) ≤ f(x_t) then
            x_t = x_s
        else
            x_s[i] = x_e[i] + ρ/2
            if f(x_s) ≤ f(x_t) then
                x_t = x_s
            end if
        end if
    end for
    if f(x_t) ≤ f(x_e) then
        x_e = x_t
    else
        ρ = ρ/2
    end if
end while
```

D. Coordination of the local search

The functioning of this LS, albeit efficient, is characterized by a steepest descent pivot rule and thus requires up to $2 \times n$ Fitness Evaluations (FEs) per iteration. In order to prevent an excess use of LS budget, especially in the early stages of optimization process, each time LS is activated we allocate a local budget of $Iter = 40$ iterations (how many times step size is halved at maximum) for LS. In addition, both algorithmic components are given a similar budget and are alternatively activated. Both DE and LS component are given equal budget of FEs, thus we can activate LS K times, where K is calculated as follows:

$$K = \frac{maxFEs}{(LSFE * NP * 2)}, \tag{5}$$

where $LSFE$ is fitness evaluations used at most by LS and NP is population size. These K breakpoints are evenly spread along optimization process:

$$FEbp = \frac{maxFEs}{K} \tag{6}$$

and LS is activated every $FEbp$ fitness evaluations. It can be observed that if the LS component improves upon the starting solution it uses less FEs than when it fails to succeed also not all population members get selected for LS. Thus, in general, the number of FEs used by DE is greater or equal than that used by LS used over the whole optimization process.

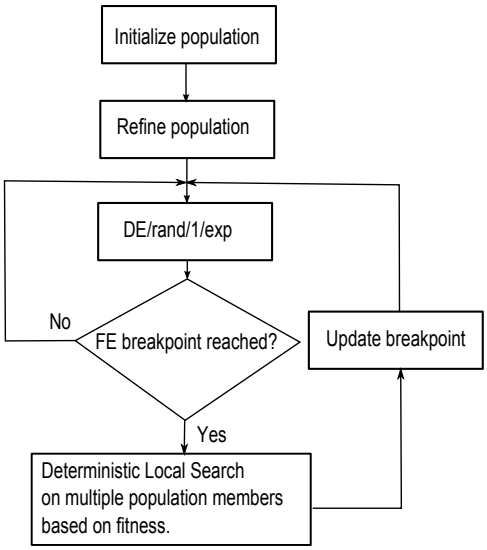

Fig. 1. Flowchart

At each breakpoint, the LS is performed on multiple individuals selected from the population. This selection is based on the individual performance (fitness of individuals) with respect to the average performance of the population. More specifically, the LS is performed over all those individuals whose fitness is better than the average fitness over all the individuals of the population. For the sake of clarity pseudo-code for selection is shown in Algorithm 3.

Algorithm 3 Selection for LS

```
for i = 1 : NP do
    if f(x_i) ≤ f_avg then
        activate LS on x_i
    end if
end for
```

The flowchart of the entire DEcfbLS is presented in Fig. 1 while the pseudocode displaying the implementation details of the proposed algorithm is given in Algorithm 4.

Algorithm 4 DEcfbLS

generate randomly NP individuals of the initial population and compute their fitness values
for $i = 1 : NP$ **do**
  apply LS on i with local budget of 4 iterations.
end for
while iterations is less than predefined budget **do**
  for $i = 1 : NP$ **do**
    select three individuals x_r, x_s, and x_t
    compute mutant individual $x_o = x_t + F(x_r - x_s)$
    compute exponential crossover between x_i and x_o thus generating x_{off}
    if $f(x_{off}) \leq f(x_i)$ **then**
      save index
    end if
  end for
  perform survivor selection where proper on the basis of saved indexes
  compute $FEbp$ according to eq. (6)
  if iterations is greater or equal than the breakpoint $FEbp$ **then**
    compute populations fitness average f_{avg}
    for $i = 1 : NP$ **do**
      if $f(i) < f_{avg}$ **then**
        apply LS on i with local budget of 40 iterations.
      end if
    end for
    update fitness evaluations breakpoint
  end if
end while

III. Numerical Results

In this study, the proposed algorithm has been run with following parameters $Np = 30$, $F = 0.7$, $\alpha_e = 0.5$, $Iter = 40$, and $\rho = 0.4$ times the width of decision space D along axis.

All experiments have been performed on CEC2013 test suite [29] in 10, 30 and 50 dimensions over 28 test problems. For each problem, 51 runs have been performed for $10000 \times n$ fitness evaluations, where n is the dimensionality of the problem. For each test problem and each algorithm, the best, worst, median, mean and standard deviation over the 51 runs has been computed as function error value $(f(x) - f^*(x))$, where $f^*(x)$ is the global minimum of function. Also algorithm complexity has been considered by calculating three different runtime values $T0$, $T1$ and $T2$ as follows:

- $T0$ is calculated by running test program shown in table I.
- $T1$ is calculated by evaluating computation time of function 14 for 200000 evaluations of certain dimension D.
- $T2$ is calculated by running function 14 and computing total computation time with 200000 evalutations of the same D dimensional benchmark function 14.

The procedure for the computation of $T0$ is reported in Algorithm 5.

The complexity of algorithm is reflected by $\hat{T}2, T1, T0$ and $(\hat{T}2 - T1)/T0$. The machine used for complexity computations is PC with Intel(R) Core(TM)2 Quad CPU Q9650 3.00GHz, 8GB RAM.

Algorithm 5 compute T0

for $i = 1 : 1000000$ **do**
  $x = 0.55 + (double)i;$
  $x = x + x; x = x./2; x = x * x;$
  $x = sqrt(x); x = log(x); x = exp(x); y = x/x;$
end for

TABLE I. Computational Complexity

	$\mathbf{T0}$	$\mathbf{T1}$	$\widehat{\mathbf{T2}}$	$(\widehat{\mathbf{T2}} - \mathbf{T1})/\mathbf{T0}$
$\mathbf{D = 10}$	66.0	797.0	740.6	−0.9
$\mathbf{D = 30}$		1976.0	3167.0	18
$\mathbf{D = 50}$		3266.0	7523.8	64.5

Table I shows the computational complexity of the proposed algorithm while Tables II, III, and IV shows the numerical results in the competition format for the proposed DEcfbLS in 10, 30, and 50, respectively.

A. Comparison against state-of-art algorithms

In order to proved a better understanding of the DEcfbLS performance, the proposed algorithm has been compared with the following modern metaheuristics:

- Covariance Matrix Adaptive Evolution Strategy with increasing population size and restart (G-CMAES), proposed in [30], with initial population $\lambda_{start} = 10$ and factor for increasing the population size equal to 2. All the other parameters are set to standard values.
- Comprehensive Learning Particle Swarm Optimizer (CLPSO), proposed in [31], with population size 60.
- Modified Differential Evolution with p-best crossover, proposed in [32], with population size 100 and $q = 0.15$ as suggested in original paper.

TABLE II. Results in 10 dimensions

	DEcfbLS				
	Best	Worst	Median	Mean	Std
f1	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$
f2	$0.00e+00$	$4.38e+03$	$1.77e-05$	$1.01e+02$	$6.09e+02$
f3	$6.09e-06$	$6.37e+00$	$2.88e-01$	$1.14e+00$	$2.00e+00$
f4	$1.91e-07$	$3.21e+01$	$8.82e-03$	$8.19e-01$	$4.57e+00$
f5	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$
f6	$0.00e+00$	$9.81e+00$	$0.00e+00$	$1.35e+00$	$3.38e+00$
f7	$7.30e-02$	$4.32e+00$	$6.23e-01$	$8.71e-01$	$8.21e-01$
f8	$2.00e+01$	$2.06e+01$	$2.03e+01$	$2.03e+01$	$1.12e-01$
f9	$5.95e-01$	$5.40e+00$	$3.82e+00$	$3.54e+00$	$1.09e+00$
f10	$7.40e-03$	$7.63e-02$	$2.95e-02$	$3.29e-02$	$1.71e-02$
f11	$0.00e+00$	$9.95e-01$	$0.00e+00$	$1.95e-02$	$1.38e-01$
f12	$2.98e+00$	$1.09e+01$	$5.97e+00$	$6.15e+00$	$1.88e+00$
f13	$4.00e+00$	$2.18e+01$	$1.10e+01$	$1.19e+01$	$4.39e+00$
f14	$0.00e+00$	$1.25e-01$	$0.00e+00$	$1.84e-02$	$3.10e-02$
f15	$2.36e+02$	$8.32e+02$	$5.20e+02$	$5.27e+02$	$1.38e+02$
f16	$2.00e-04$	$1.34e+00$	$2.51e-01$	$2.79e-01$	$1.99e-01$
f17	$1.61e+00$	$1.05e+01$	$1.01e+01$	$9.80e+00$	$1.51e+00$
f18	$9.03e+00$	$2.24e+01$	$1.63e+01$	$1.65e+01$	$2.66e+00$
f19	$1.58e-01$	$4.14e-01$	$2.82e-01$	$2.90e-01$	$6.21e-02$
f20	$1.82e+00$	$3.43e+00$	$2.50e+00$	$2.56e+00$	$4.01e-01$
f21	$4.00e+02$	$4.00e+02$	$4.00e+02$	$4.00e+02$	$0.00e+00$
f22	$9.31e+00$	$1.17e+02$	$2.73e+01$	$3.08e+01$	$1.89e+01$
f23	$3.47e+02$	$1.18e+03$	$6.28e+02$	$6.56e+02$	$1.59e+02$
f24	$1.03e+02$	$2.00e+02$	$1.08e+02$	$1.13e+02$	$2.19e+01$
f25	$1.09e+02$	$2.16e+02$	$2.00e+02$	$1.82e+02$	$3.70e+01$
f26	$1.03e+02$	$2.00e+02$	$1.08e+02$	$1.10e+02$	$1.31e+01$
f27	$3.00e+02$	$4.00e+02$	$4.00e+02$	$3.90e+02$	$2.97e+01$
f28	$1.00e+02$	$3.00e+02$	$3.00e+02$	$2.41e+02$	$9.11e+01$

TABLE III. RESULTS IN 30 DIMENSIONS

	DEcfbLS				
	Best	Worst	Median	Mean	Std
f1	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$
f2	$4.77e+04$	$5.48e+05$	$1.89e+05$	$1.99e+05$	$1.07e+05$
f3	$6.94e+00$	$2.86e+07$	$2.77e+05$	$2.11e+06$	$4.64e+06$
f4	$2.14e+01$	$2.49e+03$	$2.36e+02$	$3.82e+02$	$5.12e+02$
f5	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$
f6	$2.96e-02$	$2.64e+01$	$6.61e+00$	$7.08e+00$	$4.17e+00$
f7	$2.33e+01$	$8.93e+01$	$5.29e+01$	$5.68e+01$	$1.66e+01$
f8	$2.07e+01$	$2.11e+01$	$2.09e+01$	$2.09e+01$	$9.44e-02$
f9	$1.48e+01$	$3.12e+01$	$2.45e+01$	$2.40e+01$	$2.86e+00$
f10	$0.00e+00$	$6.89e-02$	$1.23e-02$	$2.01e-02$	$1.73e-02$
f11	$0.00e+00$	$9.95e-01$	$0.00e+00$	$5.85e-02$	$2.34e-01$
f12	$3.68e+01$	$6.87e+01$	$5.47e+01$	$5.42e+01$	$8.80e+00$
f13	$5.57e+01$	$1.37e+02$	$1.02e+02$	$1.01e+02$	$1.86e+01$
f14	$8.14e+00$	$7.62e+01$	$3.08e+01$	$3.29e+01$	$1.35e+01$
f15	$1.94e+03$	$7.35e+03$	$3.10e+03$	$3.43e+03$	$1.08e+03$
f16	$9.05e-02$	$3.90e+00$	$3.57e-01$	$7.27e-01$	$8.93e-01$
f17	$3.28e+01$	$3.75e+01$	$3.52e+01$	$3.53e+01$	$1.14e+00$
f18	$5.26e+01$	$1.02e+02$	$8.03e+01$	$7.94e+01$	$1.10e+01$
f19	$1.22e+00$	$1.93e+00$	$1.47e+00$	$1.50e+00$	$1.74e-01$
f20	$9.77e+00$	$1.36e+01$	$1.18e+01$	$1.17e+01$	$6.52e-01$
f21	$2.00e+02$	$4.44e+02$	$3.00e+02$	$3.36e+02$	$9.78e+01$
f22	$9.83e+01$	$4.66e+02$	$2.46e+02$	$2.56e+02$	$9.13e+01$
f23	$2.12e+03$	$4.77e+03$	$3.59e+03$	$3.59e+03$	$4.99e+02$
f24	$2.29e+02$	$2.77e+02$	$2.66e+02$	$2.64e+02$	$9.15e+00$
f25	$2.60e+02$	$2.93e+02$	$2.85e+02$	$2.83e+02$	$5.79e+00$
f26	$2.00e+02$	$2.00e+02$	$2.00e+02$	$2.00e+02$	$6.62e-03$
f27	$7.73e+02$	$1.05e+03$	$9.38e+02$	$9.38e+02$	$5.75e+01$
f28	$3.00e+02$	$3.00e+02$	$3.00e+02$	$3.00e+02$	$0.00e+00$

TABLE IV. RESULTS IN 50 DIMENSIONS

	DEcfbLS				
	Best	Worst	Median	Mean	Std
f1	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$
f2	$3.16e+05$	$2.42e+06$	$5.73e+05$	$6.55e+05$	$3.74e+05$
f3	$2.04e+07$	$1.39e+09$	$1.87e+08$	$2.20e+08$	$2.14e+08$
f4	$7.86e+01$	$1.25e+04$	$6.16e+02$	$1.21e+03$	$1.94e+03$
f5	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$
f6	$4.34e+01$	$4.34e+01$	$4.34e+01$	$4.34e+01$	$0.00e+00$
f7	$7.74e+01$	$1.25e+02$	$1.07e+02$	$1.05e+02$	$9.53e+00$
f8	$2.09e+01$	$2.13e+01$	$2.11e+01$	$2.11e+01$	$9.52e-02$
f9	$4.00e+01$	$5.37e+01$	$4.73e+01$	$4.71e+01$	$3.17e+00$
f10	$0.00e+00$	$6.64e-02$	$3.20e-02$	$3.21e-02$	$1.67e-02$
f11	$4.20e-03$	$1.79e+01$	$4.02e+00$	$4.64e+00$	$4.38e+00$
f12	$6.37e+01$	$2.06e+02$	$1.31e+02$	$1.34e+02$	$3.55e+01$
f13	$1.63e+02$	$3.67e+02$	$2.39e+02$	$2.47e+02$	$4.87e+01$
f14	$4.72e+01$	$4.41e+02$	$2.23e+02$	$2.31e+02$	$8.64e+01$
f15	$4.99e+03$	$1.45e+04$	$5.96e+03$	$6.25e+03$	$1.40e+03$
f16	$2.00e-01$	$4.15e+00$	$8.09e-01$	$1.63e+00$	$1.37e+00$
f17	$5.95e+01$	$7.04e+01$	$6.62e+01$	$6.58e+01$	$2.31e+00$
f18	$1.12e+02$	$1.98e+02$	$1.55e+02$	$1.57e+02$	$2.09e+01$
f19	$2.46e+00$	$3.68e+00$	$2.98e+00$	$2.95e+00$	$2.89e-01$
f20	$1.89e+01$	$2.30e+01$	$2.20e+01$	$2.17e+01$	$8.51e-01$
f21	$2.00e+02$	$1.12e+03$	$2.00e+02$	$5.24e+02$	$3.98e+02$
f22	$3.58e+02$	$1.04e+03$	$6.63e+02$	$6.89e+02$	$1.50e+02$
f23	$5.89e+03$	$1.56e+04$	$7.04e+03$	$7.77e+03$	$2.18e+03$
f24	$3.06e+02$	$3.43e+02$	$3.31e+02$	$3.31e+02$	$7.40e+00$
f25	$3.18e+02$	$3.82e+02$	$3.61e+02$	$3.60e+02$	$9.94e+00$
f26	$2.00e+02$	$2.00e+02$	$2.00e+02$	$2.00e+02$	$3.37e-02$
f27	$1.29e+03$	$1.71e+03$	$1.54e+03$	$1.55e+03$	$9.50e+01$
f28	$4.00e+02$	$4.00e+02$	$4.00e+02$	$4.00e+02$	$0.00e+00$

Each algorithm has been run 51 times for $10000 \times n$ fitness evaluations. The same problems and dimensionality values mentioned above have been considered in this section. Tables V, VI, and VII show the average final fitness $\pm$ the standard deviation and the associated statistical significance according to the Wicloxon test with confidence level 0.95, see [33]. For each pair-wise comparison on the final values obtained by DEcfbLS against GCMAES, MDE-pBX and CCPSO2 ("+", "-" and "=" indicate, respectively, a better, worse or equivalent performance of DEcfbLS against the other algorithms).

TABLE VIII. HOLM-BONFERRONI TEST ON THE FITNESS (REFERENCE = DECFBLS)

i	Optimizer	z	p	α/i	Hypothesis
3	GCMAES	-7.77e+00	3.96e-15	1.67e-02	Rejected
2	CLPSO	-4.24e+00	1.10e-05	2.50e-02	Rejected
1	MDEpBX	-3.35e+00	4.09e-04	5.00e-02	Rejected

Numerical results show that for all the considered dimensionality values, the proposed DEcfbLS tends to outperform the other algorithms. The most challenging competitor appeared to be the MDE-pBX. However, the latter scheme is outperformed of the vast majority of the problems by the proposed memetic structure. This result is confirmed by the Holm-Bonferroni prodedure [34], which we performed as described in [35], with confidence level 0.05. As shown in Table VIII, DEcfbLS is ranked first among the four algorithms, with the null-hypothesis rejected in all the pair-wise comparisons. In other words, DEcfbLS is significantly the best algorithm amongst the four considered in this study and for the problems under investigation.

IV. CONCLUSION

This paper presents a memetic approach based on a DE framework and a simple local search that moves along the axes. The proposed algorithm, namely Differential Evolution with concurrent fitness based Local Search (DEcfbLS) is composed of a DE/rand/1/exp framework and a local search operator. After first step of population initialization, this local search is applied for one time and with a small local budget to all the individuals of the population. This operation appears to have a great impact on the final performance. Subsequently, the local search activations are scheduled by means of criterion based on the exhausted budget. More specifically, DE is performed until a breakpoint is met and LS is activated on multiple population members. The individuals undergoing LS are selected on the basis of their fitness with respect to the fitness of the other population members (the most promising individuals are those undergoing LS). Thus, the total DE and LS budget is made comparable. This scheme, although simple, appears to effectively balance global and local search necessities and offer a respectable performance on a range of various problems. Numerical results demonstrate that DEcfbLS displays a very good performance with respect to modern metaheuristics representing the-state-of-the-art in computational intelligence optimization. The proposed DEcfbLS is potentially a promising candidate to take part to competition on Real-Parameter Single Objective Optimization at CEC-2013.

ACKNOWLEDGMENT

This research is supported by the Academy of Finland, Akatemiatutkija 130600, "Algorithmic design issues in Memetic Computing". The numerical experiments have been carried out on the computer network of the University of Jyväskylä by means of the software for distributed optimization Kimeme [36].

REFERENCES

[1] K. V. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Springer, 2005.

TABLE V. AVERAGE FITNESS ± STANDARD DEVIATION AND WILCOXON RANK-SUM TEST ON THE FITNESS IN 10D (REFERENCE = DECFBLS)

	DEcfbLS	GCMAES		MDEpBX		CLPSO	
f1	$\mathbf{0.00e+00 \pm 0.00e+00}$	$7.95e+03 \pm 7.28e+03$	+	$\mathbf{0.00e+00 \pm 0.00e+00}$	=	$\mathbf{0.00e+00 \pm 0.00e+00}$	=
f2	$\mathbf{1.01e+02 \pm 6.09e+02}$	$2.28e+04 \pm 2.99e+04$	+	$4.47e+02 \pm 8.73e+02$	+	$1.22e+06 \pm 6.11e+05$	+
f3	$\mathbf{1.14e+00 \pm 2.00e+00}$	$6.58e+06 \pm 1.73e+07$	+	$9.56e+03 \pm 4.38e+04$	-	$9.67e+06 \pm 7.61e+06$	+
f4	$8.19e-01 \pm 4.57e+00$	$1.19e+04 \pm 8.61e+03$	+	$\mathbf{8.25e-04 \pm 5.54e-03}$	-	$6.81e+03 \pm 2.37e+03$	+
f5	$\mathbf{0.00e+00 \pm 0.00e+00}$	$1.09e+04 \pm 1.74e+04$	+	$\mathbf{0.00e+00 \pm 0.00e+00}$	=	$\mathbf{0.00e+00 \pm 0.00e+00}$	=
f6	$\mathbf{1.35e+00 \pm 3.38e+00}$	$7.06e+03 \pm 5.57e+03$	+	$6.35e+00 \pm 4.69e+00$	+	$3.62e+00 \pm 3.40e+00$	+
f7	$\mathbf{8.71e-01 \pm 8.21e-01}$	$1.42e+13 \pm 4.70e+13$	+	$7.24e+00 \pm 9.11e+00$	+	$1.69e+01 \pm 4.76e+00$	+
f8	$\mathbf{2.03e+01 \pm 1.12e-01}$	$2.13e+01 \pm 1.82e-01$	+	$2.05e+01 \pm 8.15e-02$	+	$2.03e+01 \pm 6.42e-02$	+
f9	$3.54e+00 \pm 1.09e+00$	$1.90e+01 \pm 2.45e+00$	+	$\mathbf{2.20e+00 \pm 1.43e+00}$	-	$4.47e+00 \pm 6.22e-01$	+
f10	$\mathbf{3.29e-02 \pm 1.71e-02}$	$1.96e+03 \pm 1.87e+03$	+	$1.36e-01 \pm 1.24e-01$	+	$1.14e+00 \pm 2.24e-01$	+
f11	$1.95e-02 \pm 1.38e-01$	$8.28e+02 \pm 4.02e+02$	+	$2.48e+00 \pm 1.70e+00$	+	$\mathbf{0.00e+00 \pm 0.00e+00}$	=
f12	$\mathbf{6.15e+00 \pm 1.88e+00}$	$6.60e+02 \pm 3.49e+02$	+	$1.09e+01 \pm 5.21e+00$	+	$1.39e+01 \pm 3.52e+00$	+
f13	$\mathbf{1.19e+01 \pm 4.39e+00}$	$8.41e+02 \pm 7.26e+02$	+	$2.03e+01 \pm 9.51e+00$	+	$1.74e+01 \pm 5.51e+00$	+
f14	$\mathbf{1.84e-02 \pm 3.10e-02}$	$1.02e+03 \pm 3.02e+02$	+	$1.14e+02 \pm 9.80e+01$	+	$2.11e-01 \pm 8.61e-02$	+
f15	$\mathbf{5.27e+02 \pm 1.38e+02}$	$9.81e+02 \pm 2.53e+02$	+	$7.99e+02 \pm 2.85e+02$	+	$7.62e+02 \pm 1.37e+02$	+
f16	$2.79e-01 \pm 1.99e-01$	$\mathbf{6.38e-04 \pm 4.51e-03}$	-	$6.01e-01 \pm 4.43e-01$	+	$9.83e-01 \pm 2.00e-01$	+
f17	$\mathbf{9.80e+00 \pm 1.51e+00}$	$1.14e+01 \pm 8.32e-01$	+	$1.29e+01 \pm 2.05e+00$	+	$1.02e+01 \pm 5.16e-02$	+
f18	$\mathbf{1.65e+01 \pm 2.66e+00}$	$2.61e+02 \pm 3.58e+02$	=	$1.97e+01 \pm 4.99e+00$	+	$3.12e+01 \pm 4.35e+00$	+
f19	$\mathbf{2.90e-01 \pm 6.21e-02}$	$6.49e-01 \pm 1.15e-01$	+	$6.48e-01 \pm 2.22e-01$	+	$3.43e-01 \pm 1.17e-01$	+
f20	$\mathbf{2.56e+00 \pm 4.01e-01}$	$2.93e+00 \pm 6.26e-01$	+	$2.89e+00 \pm 5.42e-01$	+	$2.94e+00 \pm 2.53e-01$	+
f21	$4.00e+02 \pm 0.00e+00$	$3.24e+02 \pm 1.21e+02$	-	$4.00e+02 \pm 0.00e+00$	=	$\mathbf{2.83e+02 \pm 6.63e+01}$	-
f22	$3.08e+01 \pm 1.89e+01$	$1.70e+03 \pm 4.35e+02$	+	$1.19e+02 \pm 1.00e+02$	+	$\mathbf{1.86e+01 \pm 1.45e+01}$	-
f23	$\mathbf{6.56e+02 \pm 1.59e+02}$	$1.77e+03 \pm 3.37e+02$	+	$8.83e+02 \pm 3.33e+02$	+	$1.09e+03 \pm 1.70e+02$	+
f24	$\mathbf{1.13e+02 \pm 2.19e+01}$	$1.65e+02 \pm 5.82e+01$	=	$2.04e+02 \pm 4.82e+00$	+	$1.56e+02 \pm 1.39e+01$	+
f25	$1.82e+02 \pm 3.70e+01$	$1.88e+02 \pm 3.35e+01$	-	$2.01e+02 \pm 2.12e+00$	=	$\mathbf{1.77e+02 \pm 1.79e+01}$	-
f26	$\mathbf{1.10e+02 \pm 1.31e+01}$	$1.23e+02 \pm 4.38e+01$	-	$1.41e+02 \pm 4.23e+01$	+	$1.18e+02 \pm 5.06e+00$	+
f27	$3.90e+02 \pm 2.97e+01$	$\mathbf{3.00e+02 \pm 1.10e-03}$	-	$3.01e+02 \pm 3.92e+00$	-	$3.45e+02 \pm 1.95e+01$	-
f28	$\mathbf{2.41e+02 \pm 9.11e+01}$	$5.48e+02 \pm 7.65e+02$	+	$3.09e+02 \pm 6.90e+01$	=	$2.50e+02 \pm 6.13e+01$	=

TABLE VI. AVERAGE FITNESS ± STANDARD DEVIATION AND WILCOXON RANK-SUM TEST ON THE FITNESS IN 30D (REFERENCE = DECFBLS)

	DEcfbLS	GCMAES		MDEpBX		CLPSO	
f1	$\mathbf{0.00e+00 \pm 0.00e+00}$	$4.42e+03 \pm 2.94e+03$	+	$\mathbf{0.00e+00 \pm 0.00e+00}$	=	$\mathbf{0.00e+00 \pm 0.00e+00}$	=
f2	$1.99e+05 \pm 1.07e+05$	$\mathbf{7.65e+03 \pm 4.66e+03}$	-	$8.57e+04 \pm 5.00e+04$	-	$1.84e+07 \pm 4.43e+06$	+
f3	$\mathbf{2.11e+06 \pm 4.64e+06}$	$8.56e+06 \pm 1.25e+07$	+	$1.48e+07 \pm 2.46e+07$	+	$1.40e+09 \pm 5.54e+08$	+
f4	$3.82e+02 \pm 5.12e+02$	$5.28e+03 \pm 3.44e+03$	+	$\mathbf{1.33e+01 \pm 3.92e+01}$	-	$2.79e+04 \pm 3.15e+03$	+
f5	$\mathbf{0.00e+00 \pm 0.00e+00}$	$2.62e+03 \pm 1.18e+03$	+	$\mathbf{0.00e+00 \pm 0.00e+00}$	=	$\mathbf{0.00e+00 \pm 0.00e+00}$	=
f6	$\mathbf{7.08e+00 \pm 4.17e+00}$	$2.15e+03 \pm 1.09e+03$	+	$3.34e+01 \pm 3.00e+01$	+	$6.76e+01 \pm 9.13e+00$	+
f7	$\mathbf{5.68e+01 \pm 1.66e+01}$	$1.64e+13 \pm 8.15e+13$	+	$6.09e+01 \pm 1.73e+01$	=	$7.21e+01 \pm 7.80e+00$	+
f8	$\mathbf{2.09e+01 \pm 9.44e-02}$	$2.15e+01 \pm 7.45e-02$	+	$2.10e+01 \pm 5.23e-02$	+	$2.10e+01 \pm 4.75e-02$	+
f9	$2.40e+01 \pm 2.86e+00$	$5.97e+01 \pm 4.92e+00$	+	$\mathbf{2.27e+01 \pm 4.25e+00}$	=	$2.81e+01 \pm 1.73e+00$	+
f10	$\mathbf{2.01e-02 \pm 1.73e-02}$	$1.52e+03 \pm 6.51e+02$	+	$1.44e-01 \pm 8.45e-02$	+	$3.57e+00 \pm 6.63e-01$	+
f11	$5.85e-02 \pm 2.34e-01$	$6.27e+02 \pm 2.94e+02$	+	$4.70e+01 \pm 1.32e+01$	+	$\mathbf{0.00e+00 \pm 0.00e+00}$	-
f12	$\mathbf{5.42e+01 \pm 8.80e+00}$	$1.05e+03 \pm 8.28e+02$	+	$7.18e+01 \pm 1.85e+01$	+	$1.43e+02 \pm 1.71e+01$	+
f13	$\mathbf{1.01e+02 \pm 1.86e+01}$	$1.60e+03 \pm 1.28e+03$	+	$1.46e+02 \pm 3.27e+01$	+	$1.80e+02 \pm 1.42e+01$	+
f14	$3.29e+01 \pm 1.35e+01$	$3.57e+03 \pm 1.06e+03$	+	$1.16e+03 \pm 4.50e+02$	+	$\mathbf{1.74e+01 \pm 2.37e+01}$	-
f15	$\mathbf{3.43e+03 \pm 1.08e+03}$	$4.50e+03 \pm 7.47e+02$	+	$3.92e+03 \pm 6.42e+02$	+	$4.62e+03 \pm 3.64e+02$	+
f16	$7.27e-01 \pm 8.93e-01$	$\mathbf{6.24e-03 \pm 6.80e-03}$	-	$1.26e+00 \pm 7.53e-01$	+	$1.97e+00 \pm 2.77e-01$	+
f17	$3.53e+01 \pm 1.14e+00$	$3.78e+03 \pm 1.19e+03$	+	$7.09e+01 \pm 1.28e+01$	+	$\mathbf{3.15e+01 \pm 2.73e-01}$	-
f18	$\mathbf{7.94e+01 \pm 1.10e+01}$	$3.94e+03 \pm 8.53e+02$	+	$8.15e+01 \pm 1.34e+01$	=	$1.99e+02 \pm 1.41e+01$	+
f19	$1.50e+00 \pm 1.74e-01$	$2.45e+00 \pm 4.67e-01$	+	$9.86e+00 \pm 7.61e+00$	+	$\mathbf{1.39e+00 \pm 3.12e-01}$	-
f20	$1.17e+01 \pm 6.52e-01$	$1.41e+01 \pm 2.04e+00$	+	$\mathbf{1.08e+01 \pm 7.21e-01}$	-	$1.30e+01 \pm 4.56e-01$	+
f21	$3.36e+02 \pm 9.78e+01$	$\mathbf{2.24e+02 \pm 4.24e+01}$	-	$3.23e+02 \pm 8.67e+01$	=	$3.02e+02 \pm 5.03e+00$	=
f22	$2.56e+02 \pm 9.13e+01$	$4.50e+03 \pm 1.57e+03$	+	$1.24e+03 \pm 4.50e+02$	+	$\mathbf{1.21e+02 \pm 7.91e+00}$	-
f23	$\mathbf{3.59e+03 \pm 4.99e+02}$	$5.70e+03 \pm 9.43e+02$	+	$4.48e+03 \pm 7.47e+02$	+	$5.59e+03 \pm 3.59e+02$	+
f24	$2.64e+02 \pm 9.15e+00$	$3.01e+02 \pm 3.53e+02$	-	$\mathbf{2.32e+02 \pm 1.10e+01}$	-	$2.47e+02 \pm 4.51e+00$	-
f25	$2.83e+02 \pm 5.79e+00$	$\mathbf{2.54e+02 \pm 1.71e+01}$	-	$2.78e+02 \pm 1.12e+01$	-	$2.97e+02 \pm 1.07e+01$	+
f26	$\mathbf{2.00e+02 \pm 6.62e-03}$	$2.62e+02 \pm 2.45e+02$	-	$2.26e+02 \pm 5.21e+01$	-	$2.01e+02 \pm 3.45e-01$	+
f27	$9.38e+02 \pm 5.75e+01$	$\mathbf{4.01e+02 \pm 1.28e+02}$	-	$6.48e+02 \pm 1.18e+02$	-	$8.19e+02 \pm 1.92e+02$	-
f28	$3.00e+02 \pm 0.00e+00$	$5.76e+02 \pm 1.11e+03$	+	$\mathbf{2.88e+02 \pm 4.71e+01}$	+	$3.00e+02 \pm 8.00e-02$	+

[2] V. P. Plagianakos, D. K. Tasoulis, and M. N. Vrahatis, "A Review of Major Application Areas of Differential Evolution," in *Advances in Differential Evolution*, ser. Studies in Computational Intelligence, U. K. Chakraborty, Ed. Springer, 2008, vol. 143, pp. 197–238.

[3] R. Joshi and A. C. Sanderson, "Minimal representation multisensor fusion using differential evolution," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 29, no. 1, pp. 63–76, 1999.

[4] T. Rogalsky and R. W. Derksen, "Hybridization of Differential Evolution for Aerodynamic Design," in *Proceedings of the 8th Annual Conference of the Computational Fluid Dynamics Society of Canada*, June 2000, pp. 729–736.

[5] N. Karaboga and B. Cetinkaya, "Design of Digital FIR Filters Using Differential Evolution Algorithm," *Circuits, Systems, and Signal Processing*, vol. 25, no. 5, pp. 649–660, October 2006.

[6] S. Das and P. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 4–31, feb. 2011.

[7] F. Neri and V. Tirronen, "Recent Advances in Differential Evolution: A Review and Experimental Analysis," *Artificial Intelligence Review*, vol. 33, no. 1–2, pp. 61–106, 2010.

[8] M. Weber, V. Tirronen, and F. Neri, "Scale Factor Inheritance Mechanism in Distributed Differential Evolution," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 14, no. 11, pp. 1187–1207, 2010.

[9] J. Brest, A. Zamuda, B. Bošković, and V. Žumer, "An Analysis of the Control Parameters'Adaptation in DE," in *Advances in Differential Evolution*, ser. Studies in Computational Intelligence, U. K. Chakraborty, Ed. Springer, 2008, vol. 143, pp. 89–110.

[10] N. Salvatore, A. Caponio, F. Neri, S. Stasi, and G. L. Cascella, "Optimization of Delayed-State Kalman Filter-based Algorithm via

TABLE VII. AVERAGE FITNESS ± STANDARD DEVIATION AND WILCOXON RANK-SUM TEST ON THE FITNESS IN 50D (REFERENCE = DECFBLS)

	DEcfbLS	GCMAES		MDEpBX		CLPSO	
f1	$\mathbf{0.00e+00 \pm 0.00e+00}$	$2.67e+03 \pm 1.07e+03$	+	$\mathbf{0.00e+00 \pm 0.00e+00}$	=	$\mathbf{0.00e+00 \pm 0.00e+00}$	=
f2	$6.55e+05 \pm 3.74e+05$	$\mathbf{7.96e+03 \pm 4.11e+03}$	-	$4.59e+05 \pm 2.43e+05$	-	$3.58e+07 \pm 5.81e+06$	+
f3	$2.20e+08 \pm 2.14e+08$	$\mathbf{1.01e+06 \pm 1.78e+06}$	-	$9.84e+07 \pm 1.60e+08$	-	$2.77e+09 \pm 6.96e+08$	+
f4	$1.21e+03 \pm 1.94e+03$	$4.58e+03 \pm 2.72e+03$	+	$\mathbf{2.36e+01 \pm 2.36e+01}$	-	$3.42e+04 \pm 3.57e+03$	+
f5	$\mathbf{0.00e+00 \pm 0.00e+00}$	$2.13e+03 \pm 6.58e+02$	+	$\mathbf{0.00e+00 \pm 0.00e+00}$	=	$\mathbf{0.00e+00 \pm 0.00e+00}$	=
f6	$\mathbf{4.34e+01 \pm 0.00e+00}$	$1.98e+03 \pm 9.49e+02$	+	$5.57e+01 \pm 2.33e+01$	+	$4.67e+01 \pm 5.09e-01$	+
f7	$1.05e+02 \pm 9.53e+00$	$6.65e+14 \pm 3.78e+15$	+	$\mathbf{6.46e+01 \pm 1.37e+01}$	-	$8.88e+01 \pm 6.24e+00$	-
f8	$\mathbf{2.11e+01 \pm 9.52e-02}$	$2.15e+01 \pm 5.98e-02$	+	$2.12e+01 \pm 4.37e-02$	+	$2.11e+01 \pm 3.28e-02$	=
f9	$4.71e+01 \pm 3.17e+00$	$9.78e+01 \pm 5.23e+00$	+	$\mathbf{4.43e+01 \pm 7.26e+00}$	-	$5.40e+01 \pm 2.41e+00$	+
f10	$\mathbf{3.21e-02 \pm 1.67e-02}$	$1.35e+03 \pm 4.56e+02$	+	$1.53e-01 \pm 2.67e-01$	+	$1.44e+01 \pm 2.11e+00$	+
f11	$4.64e+00 \pm 4.38e+00$	$5.13e+02 \pm 2.36e+02$	+	$1.19e+02 \pm 2.73e+01$	+	$\mathbf{0.00e+00 \pm 0.00e+00}$	-
f12	$\mathbf{1.34e+02 \pm 3.55e+01}$	$2.33e+03 \pm 1.34e+03$	+	$1.59e+02 \pm 3.07e+01$	+	$3.25e+02 \pm 2.65e+01$	+
f13	$\mathbf{2.47e+02 \pm 4.87e+01}$	$3.64e+03 \pm 1.05e+03$	+	$3.20e+02 \pm 4.24e+01$	+	$3.92e+02 \pm 2.01e+01$	+
f14	$2.31e+02 \pm 8.64e+01$	$7.46e+03 \pm 9.56e+02$	+	$2.76e+03 \pm 7.65e+02$	+	$\mathbf{2.97e+01 \pm 2.13e+01}$	-
f15	$\mathbf{6.25e+03 \pm 1.40e+03}$	$8.56e+03 \pm 8.83e+02$	+	$7.61e+03 \pm 8.52e+02$	+	$9.57e+03 \pm 4.86e+02$	+
f16	$1.63e+00 \pm 1.37e+00$	$\mathbf{3.03e-03 \pm 2.72e-03}$	-	$1.85e+00 \pm 8.83e-01$	+	$2.60e+00 \pm 2.92e-01$	+
f17	$6.58e+01 \pm 2.31e+00$	$7.07e+03 \pm 1.20e+03$	+	$1.74e+02 \pm 3.90e+01$	+	$\mathbf{5.35e+01 \pm 5.13e-01}$	-
f18	$\mathbf{1.57e+02 \pm 2.09e+01}$	$6.96e+03 \pm 1.09e+03$	+	$1.84e+02 \pm 2.96e+01$	+	$4.11e+02 \pm 2.07e+01$	+
f19	$2.95e+00 \pm 2.89e-01$	$4.42e+00 \pm 6.69e-01$	+	$3.61e+01 \pm 1.68e+01$	+	$\mathbf{2.68e+00 \pm 3.69e-01}$	-
f20	$2.17e+01 \pm 8.51e-01$	$2.01e+01 \pm 3.02e+00$	-	$\mathbf{1.98e+01 \pm 8.35e-01}$	-	$2.22e+01 \pm 3.52e-01$	+
f21	$\mathbf{5.24e+02 \pm 3.98e+02}$	$6.85e+02 \pm 4.33e+02$	=	$8.67e+02 \pm 3.67e+02$	+	$5.32e+02 \pm 2.15e+02$	+
f22	$6.89e+02 \pm 1.50e+02$	$1.05e+04 \pm 1.33e+03$	+	$3.24e+03 \pm 1.15e+03$	+	$\mathbf{5.55e+01 \pm 2.84e+01}$	-
f23	$\mathbf{7.77e+03 \pm 2.18e+03}$	$1.15e+04 \pm 1.05e+03$	+	$9.00e+03 \pm 1.16e+03$	+	$1.14e+04 \pm 6.12e+02$	+
f24	$3.31e+02 \pm 7.40e+00$	$3.79e+02 \pm 5.21e+02$	-	$\mathbf{2.84e+02 \pm 1.42e+01}$	-	$3.01e+02 \pm 7.53e+00$	-
f25	$3.60e+02 \pm 9.94e+00$	$\mathbf{3.16e+02 \pm 6.31e+01}$	-	$3.67e+02 \pm 1.44e+01$	+	$4.22e+02 \pm 6.94e+00$	+
f26	$\mathbf{2.00e+02 \pm 3.37e-02}$	$3.58e+02 \pm 5.17e+02$	=	$3.46e+02 \pm 8.23e+01$	+	$2.04e+02 \pm 8.57e-01$	+
f27	$1.55e+03 \pm 9.50e+01$	$\mathbf{8.00e+02 \pm 2.42e+02}$	-	$1.28e+03 \pm 1.63e+02$	-	$1.67e+03 \pm 1.86e+02$	+
f28	$\mathbf{4.00e+02 \pm 0.00e+00}$	$1.35e+03 \pm 2.06e+03$	=	$5.42e+02 \pm 7.05e+02$	-	$4.00e+02 \pm 1.02e-03$	+

Differential Evolution for Sensorless Control of Induction Motors," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 1, pp. 385–394, 2010.

[11] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[12] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[13] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[14] J. Lampinen and I. Zelinka, "On Stagnation of the Differential Evolution Algorithm," in *Proceedings of 6th International Mendel Conference on Soft Computing*, P. Ošmera, Ed., 2000, pp. 76–83.

[15] J. Zhang and A. C. Sanderson, "JADE: Adaptive Differential Evolution with Optional External Archive," vol. 13, no. 5, 2009, pp. 945–958.

[16] J. Brest and M. S. Maučec, "Population size reduction for the differential evolution algorithm," *Applied Intelligence*, vol. 29, no. 3, pp. 228–247, 2008.

[17] N. Noman and H. Iba, "Accelerating Differential Evolution Using an Adaptive Local Search," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 107–125, 2008.

[18] A. Caponio, F. Neri, and V. Tirronen, "Super-fit control adaptation in memetic differential evolution frameworks," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 13, pp. 811–831, 2009.

[19] F. Neri and V. Tirronen, "Scale Factor Local Search in Differential Evolution," *Memetic Computing Journal*, vol. 1, no. 2, pp. 153–171, 2009.

[20] Z. Z. Liu, F. L. Luo, and M. A. Rahman, "Robust and precision motion control system of linear-motor direct drive for high-speed X-Y table positioning mechanism," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 5, pp. 1357–1363, Oct. 2005.

[21] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential Evolution with a Neighborhood-based Mutation Operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.

[22] S. Islam, S. Das, S. Ghosh, S. Roy, and P. Suganthan, "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*, vol. 42, no. 2, pp. 482–500, april 2012.

[23] R. Storn, "Differential evolution design of an IIR-filter," in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 268–273.

[24] F. Neri, G. Iacca, and E. Mininno, "Disturbed Exploitation compact Differential Evolution for Limited Memory Optimization Problems," *Information Sciences*, vol. 181, no. 12, pp. 2469–2487, 2011.

[25] G. Iacca, F. Neri, E. Mininno, Y. S. Ong, and M. H. Lim, "Ockham's Razor in Memetic Computing: Three Stage Optimal Memetic Exploration," *Information Sciences*, vol. 188, pp. 17–43, 2012.

[26] F. Caraffini, F. Neri, G. Iacca, and A. Mol, "Parallel Memetic Structures," *Information Sciences*, vol. 227, pp. 60–82, 2013.

[27] I. Poikolainen, G. Iacca, F. Neri, E. Mininno, and M. Weber, "Shrinking three stage optimal memetic exploration," in *Proceedings of the Fifth International Conference on Bioinspired Optimization Methods and their Applications*, 2012, pp. 61–74.

[28] R. Hooke and T. A. Jeeves, "Direct search solution of numerical and statistical problems," *Journal of the ACM*, vol. 8, pp. 212–229, Mar. 1961.

[29] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernndez-Daz, "Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization," Zhengzhou University and Nanyang Technological University, Zhengzhou China and Singapore, Tech. Rep. 201212, 2013.

[30] A. Auger and N. Hansen, "A Restart CMA Evolution Strategy With Increasing Population Size," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2005, pp. 1769–1776.

[31] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[32] S. G. S. R. S. S. P. Islam, S.M.; Das, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on System Man and Cybernetics-part B*, vol. 42, pp. 482–500, 2012.

[33] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[34] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.

[35] S. Garcia, A. Fernandez, J. Luengo, and F. Herrera, "A study of statis-

tical techniques and performance measures for genetics-based machine learning: accuracy and interpretability," *Soft Computing*, vol. 13, no. 10, pp. 959–977, 2008.

[36] Cyber Dyne Srl Home Page, "Kimeme," 2012, http://cyberdynesoft.it/.