Cooperative Micro–Differential Evolution for High–Dimensional Problems

Konstantinos E. Parsopoulos
Department of Mathematics
University of Patras
GR–26110 Patras, Greece
kostasp@math.upatras.gr

ABSTRACT

High–dimensional optimization problems appear very often in demanding applications. Although evolutionary algorithms constitute a valuable tool for solving such problems, their standard variants exhibit deteriorating performance as dimension increases. In such cases, cooperative approaches have proved to be very useful, since they divide the computational burden to a number of cooperating subpopulations. In contrast, Micro–evolutionary approaches constitute light versions of the original evolutionary algorithms that employ very small populations of just a few individuals to address optimization problems. Unfortunately, this property is usually accompanied by limited efficiency and proneness to get stuck in local minima. In the present work, an approach that combines the basic properties of cooperation and Micro-evolutionary algorithms is presented for the Differential Evolution algorithm. The proposed Cooperative Micro–Differential Evolution approach employs small cooperative subpopulations to detect subcomponents of the original problem solution concurrently. The subcomponents are combined through cooperation of subpopulations to build complete solutions of the problem. The proposed approach is illustrated on high-dimensional instances of five widely used test problems with very promising results. Comparisons with the standard Differential Evolution algorithm are also reported and their statistical significance is analyzed.

Categories and Subject Descriptors

G.1.6 [**Optimization**]: Global optimization,Unconstrained optimization; G.3 [**Probability and Statistics**]: Probabilistic algorithms

General Terms

Algorithms, Performance, Experimentation

Keywords

Differential Evolution, Micro–Differential Evolution, Evolutionary Algorithms, Cooperative Algorithms, Micro–Evolutionary Algorithms

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

1. INTRODUCTION

Modern applications often involve the solution of complex high–dimensional optimization problems. Evolutionary algorithms have been used widely for solving such problems, especially in cases where the underlying objective functions lack nice mathematical properties such as continuity and differentiability. Usually, the increased dimensionality of the problem poses obstacles on the employed algorithm, reducing its performance significantly. This deficiency is also known as the *curse of dimensionality* and its alleviation constitutes a subject of ongoing research.

Cooperative Evolutionary Algorithms (CEAs) have proved to be a valuable tool in cases where the standard evolutionary algorithms fail [9]. CEAs consist of a number of cooperative populations that attack low–dimensional subcomponents of the original problem and evolve them concurrently. Cooperation among subpopulations is responsible for bringing together their discoveries and build complete solutions of the original problem. Various cooperative approaches have been proposed and analyzed in literature [2, 8, 9, 13]. A typical example is the Cooperative Coevolutionary Genetic Algorithm (CCGA) of Potter and De Jong [8], which is based on Genetic Algorithms. Evolution Strategies and Particle Swarm Optimization have also been used as the basic algorithmic elements of cooperative approaches [4, 14, 16]. Recently, two cooperative approaches based on the Differential Evolution (DE) algorithm were proposed [7, 12]. Extensive experimentation has revealed also several deficiencies of cooperative schemes, such as the deteriorating performance in problems with correlated coordinate directions and the introduction of new local minima [8].

In contrast to CEAs, Micro–Evolutionary Algorithms (Micro–EAs) are instances of the standard evolutionary algorithms with very small population size and ability to handle simple fitness functions. Although Micro–EAs were primarily used for very simple problems and educational purposes, several attempts have been made to use them in demanding applications. For example, Micro–Genetic Algorithms (Micro–GAs), also called Tiny–GAs, have been studied in image processing problems [6]. Recently, Micro–Particle Swarm Optimization was proposed for tackling high–dimensional optimization problems [5], and a special version of Micro–Differential Evolution (Micro–DE) with opposition–based operators was used for image thresholding [11].

The small population size of Micro–EAs limits their exploration capabilities, especially in complex environments. This is usually caused due to the rapid convergence of the population to the most promising detected solutions, which decreases population diversity in early iterations and, consequently, deteriorates efficiency. As a countermeasure, diversity–preserving schemes are usually incorporated in the algorithm, along with proper techniques that prevent convergence to the same solution [5].

The present work combines the main concepts of CEAs and Micro–EAs to produce a Cooperative Micro–Differential Evolution (COMDE) algorithm. To the best of the author's knowledge, this is the first variant of DE that combines these two approaches. The proposed scheme aims at solving high–dimensional problems more efficiently than the standard DE algorithm. To achieve this, high–dimensional candidate solutions are divided into low–dimensional subcomponents, and each one is tackled with a small, low–dimensional subpopulation. Information sharing among subpopulations allows the construction of complete solutions for the evaluation of each individual with the original objective function. COMDE is tested against the standard DE algorithm on high–dimensional instances of five widely used test problems from the relative literature.

The remaining of the paper is organized as follows: Section 2 describes the DE algorithm, while Section 3 introduces the COMDE approach. Experimental results are reported and discussed in Section 4, and the paper concludes in Section 5.

2. DIFFERENTIAL EVOLUTION

The DE algorithm was developed by Storn and Price [10, 15] as a population–based stochastic optimization algorithm for numerical optimization problems. DE utilizes a population:

$$P = \{x_1, x_2, \ldots, x_N\},$$

of N individuals to probe the search space. The population is initialized randomly in the search space, usually following a uniform distribution. Each individual is an n–dimensional vector:

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^\top, \quad i = 1, 2, \ldots, N,$$

and serves as a candidate solution of the problem at hand. The population is evolved by applying two operators, namely *mutation* and *recombination*, which produce new candidate solutions. Then, the old and the new population are merged, and *selection* takes place to construct a new population that consists of the N best individuals. These operators are applied iteratively until a termination condition is met.

The mutation operator produces a new vector, v_i, for each individual, x_i, $i = 1, 2, \ldots, N$, by combining some of the rest individuals of the population. Different operators have been proposed for this task, with the following constituting the most common ones:

$$v_i(t+1) = x_g(t) + F\Big(x_{r_1}(t) - x_{r_2}(t)\Big), \quad (1)$$

$$v_i(t+1) = x_{r_1}(t) + F\Big(x_{r_2}(t) - x_{r_3}(t)\Big), \quad (2)$$

$$v_i(t+1) = x_i(t) + F\Big(x_g(t) - x_i(t) + x_{r_1}(t) - x_{r_2}(t)\Big), \quad (3)$$

$$v_i(t+1) = x_g(t) + F\Big(x_{r_1}(t) - x_{r_2}(t) + x_{r_3}(t) - x_{r_4}(t)\Big), \quad (4)$$

$$v_i(t+1) = x_{r_1}(t) + F\Big(x_{r_2}(t) - x_{r_3}(t) + x_{r_4}(t) - x_{r_5}(t)\Big), \quad (5)$$

where t denotes the iteration counter; F is a fixed user–defined parameter; g denotes the index of the best individual in the population, i.e., the one with the smallest function value; and $r_i \in \{1, 2, \ldots, N\}$, $i = 1, 2, \ldots, 5$, are mutually different randomly selected indices that differ also from the index i. Thus, in order to be able to apply all mutation operators, it must hold that $N > 5$. All vector operations in Eqs. (1)–(5) are performed componentwise. The five operators will be henceforth denoted as OP1–OP5, respectively.

After mutation, a recombination operator is applied on the generated vectors, v_i, producing for each one a trial vector:

$$u_i = (u_{i1}, u_{i2}, \ldots, u_{in})^\top, \quad i = 1, 2, \ldots, N,$$

which is defined as follows:

$$u_{ij}(t+1) = \begin{cases} v_{ij}(t+1), & \text{if } R_j \leqslant CR \text{ or } j = \mathrm{RI}(i), \\ x_{ij}(t), & \text{if } R_j > CR \text{ and } j \neq \mathrm{RI}(i), \end{cases}$$

where $j = 1, 2, \ldots, n$; R_j is the j–th evaluation of a uniform random number generator in the range $[0, 1]$; $CR \in [0, 1]$ is a user–defined *crossover constant*; and $\mathrm{RI}(i)$ is an index randomly selected from the set $\{1, 2, \ldots, n\}$.

Finally, in the selection phase, the produced trial vectors, u_i, are compared against the corresponding individuals, x_i, and the best among them comprise the population in the next generation, i.e.:

$$x_i(t+1) = \begin{cases} u_i(t+1), & \text{if } f(u_i(t+1)) < f(x_i(t)), \\ x_i(t), & \text{otherwise}, \end{cases}$$

where $f(x)$ is the objective function under consideration.

3. THE PROPOSED APPROACH

Micro–DE has the same structure and operations with standard DE. The only difference is the population size, which is typically very small. Thus, although it is recommended to use populations of size up to $N = 10n$ [15], where n is the problem dimension, Micro–DE uses the smallest possible number of individuals. Taking into consideration the restriction, $N > 5$, that permits the application of all mutation operators, a population size, $N = 6$, can be considered a reasonable choice for Micro–DE.

Moreover, Micro–DE is expected to converge rapidly due to the small number of individuals. Usually, the ratio, n/N, is indicative of the difficulty met by an algorithm on a given problem. Small values of this ratio (less than 1) correspond to population with size larger than its dimension. On the other hand, values higher than 1 correspond to problem dimension higher than population size. Empirical evidence suggest that in most cases the higher the ratio is, the harder the problem becomes for the algorithm. Therefore, Micro–DE can be considered as a promising approach in rather low–dimensional problems.

The aforementioned deficiency can be addressed through the proposed COMDE approach, a cooperative scheme for Micro–DE. To put it formally, let, $n_1, n_2, \ldots, n_K$, be K positive integers such that:

$$n = \sum_{k=1}^{K} n_k,$$

where n is the dimension of the original problem. Then, a candidate solution vector of the original problem can be divided into K subcomponents, each one addressed by a

Table 1: Pseudocode of the proposed COMDE approach.

Input:	K (number of subpopulations); N_i (sizes of subpopulations); n_i (dimensions of subpopulations); $i = 1, 2, \ldots, K$; M (buffer vector); f (objective function)
Step 1.	**Initialize** subpopulations randomly within their search spaces (subspaces of the original one).
Step 2.	**Initialize** buffer vector, M, using a randomly selected individual from each subpopulation.
Step 3.	**While** (termination condition not met)
Step 4.	**Do** ($k = 1, \ldots, K$)
Step 5.	**Do** ($i = 1, \ldots, N_k$)
Step 6.	**Update** the individual $x_i^{[k]}$ with the standard DE operations.
Step 7.	**Evaluate** $x_i^{[k]}$ using Eq. (6) and the buffer M.
Step 8.	**Update** the best position $x_g^{[k]}$ of the population P_k.
Step 9.	**If** $\left(f\left(x_i^{[k]}\right) < f(M)\right)$ **Then**
Step 10.	**Copy** $x_i^{[k]}$ in the proper position of the buffer M.
Step 11.	**End If**
Step 12.	**End Do**
Step 13.	**End Do**
Step 14.	**End While**
Step 15.	**Print** buffer M and $f(M)$.

Table 2: Dimension and range for each test problem.

Problem	Dimension (n)	Range
TP1	300, 600, 900, 1200	$[-100, 100]^n$
TP2	300, 600, 900, 1200	$[-30, 30]^n$
TP3	300, 600, 900, 1200	$[-5.12, 5.12]^n$
TP4	300, 600, 900, 1200	$[-600, 600]^n$
TP5	300, 600, 900, 1200	$[-20, 30]^n$

Table 3: The total number of individuals and subpopulations of 6 individuals per problem dimension.

Problem Dimension	Total number of individuals	Number of subpopulations
300	360	60
600	720	120
900	1080	180
1200	1440	240

different subpopulation, P_i, of size, N_i, and dimension, n_i, $i = 1, 2, \ldots, K$. Thus, each subpopulation is assigned the minimization of its corresponding subcomponent, which has strictly smaller dimension than n.

The subpopulations work in the same manner as for the original DE algorithm described in Section 2. However, an apparent issue arises regarding the evaluation of individuals with the objective function due to their different dimension. This problem can be addressed by using an information sharing mechanism in the form of a common memory buffer for all subpopulations, where they deposit their best individuals. This buffer is also called *context vector* and it is defined as an n–dimensional vector, $M = (m_1, m_2, \ldots, m_n)^\top$, where each subpopulation deposits its contribution. Hence, if:

$$s^{[k]} = \left(s_1^{[k]}, s_2^{[k]}, \ldots, s_{n_k}^{[k]}\right)^\top,$$

Table 4: COMDE subpopulation parameters.

Parameter	Description	Value
N_k	subpopulation size	6
n_k	subpopulation dimension	5
$t_{\max}$	maximum iterations	10^3
F	DE parameter	0.5
CR	DE parameter	0.7

is the n_k–dimensional vector (with $n_k < n$) contributed by the k–th subpopulation, P_k, $k = 1, 2, \ldots, K$, then the buffer vector is defined as:

$$M = \Big(\underbrace{s_1^{[1]}, \ldots, s_{n_1}^{[1]}}_{s^{[1]} \text{ of } P_1}, \underbrace{s_1^{[2]}, \ldots, s_{n_2}^{[2]}}_{s^{[2]} \text{ of } P_2}, \ldots, \underbrace{s_1^{[K]}, \ldots, s_{n_K}^{[K]}}_{s^{[K]} \text{ of } P_K}\Big)^\top.$$

Then, the i–th individual of the j–th subpopulation:

$$x_i^{[j]} = \left(x_{i1}^{[j]}, x_{i2}^{[j]}, \ldots, x_{i,n_j}^{[j]}\right)^\top,$$

is evaluated using the buffer vector, M, by substituting the components that correspond to the contribution of the j–th population with the actual components of $x_i^{[j]}$, while the rest components of the buffer remain unaffected, i.e.:

$$f\left(x_i^{[j]}\right) = f\left(M_i^{[j]}\right), \qquad (6)$$

where,

$$M_i^{[j]} = \Big(s_1^{[1]}, \ldots, s_{n_1}^{[1]}, \ldots, \underbrace{x_{i1}^{[j]}, \ldots, x_{i,n_j}^{[j]}}_{\text{individual } x_i^{[j]}}, \ldots, s_1^{[K]}, \ldots, s_{n_K}^{[K]}\Big)^\top,$$

$i = 1, 2, \ldots, N_j$; $j = 1, 2, ..., K$.

A straightforward choice for the contribution of each subpopulation is its overall best position, i.e., $s^{[k]} = x_g^{[k]}$, which results in a buffer that contains all best positions of the sub-

Table 5: Results for TP1.

Oper.	Dim.	K	Mean	StD
OP1	300	1	$1.4144e+05$	$1.8617e+04$
		60	$8.6876e+04$	$1.5732e+04$
	600	1	$4.4721e+05$	$3.1952e+04$
		120	$1.8909e+05$	$2.0218e+04$
	900	1	$8.3351e+05$	$5.9673e+04$
		180	$2.7960e+05$	$2.6643e+04$
	1200	1	$1.1977e+06$	$6.1182e+04$
		240	$3.5644e+05$	$2.8171e+04$
OP2	300	1	$3.2582e+05$	$2.0170e+04$
		60	$9.3599e+04$	$2.2051e+04$
	600	1	$1.0247e+06$	$4.0161e+04$
		120	$2.0026e+05$	$3.4124e+04$
	900	1	$1.8648e+06$	$6.1357e+04$
		180	$3.1485e+05$	$4.5966e+04$
	1200	1	$2.9638e+06$	$1.9296e+05$
		240	$4.1765e+05$	$3.6677e+04$
OP3	300	1	$6.8858e+04$	$6.4519e+03$
		60	$5.6902e+04$	$8.9052e+03$
	600	1	$2.6147e+05$	$2.0403e+04$
		120	$1.0856e+05$	$1.1119e+04$
	900	1	$5.1087e+05$	$3.6903e+04$
		180	$1.5760e+05$	$1.2894e+04$
	1200	1	$7.9133e+05$	$6.1529e+04$
		240	$2.1152e+05$	$1.5359e+04$
OP4	300	1	$1.8365e+05$	$2.0640e+04$
		60	$1.4505e+04$	$1.0504e+04$
	600	1	$8.1520e+05$	$2.7794e+04$
		120	$2.6381e+04$	$1.2269e+04$
	900	1	$1.4362e+06$	$3.8525e+04$
		180	$4.8767e+04$	$1.8794e+04$
	1200	1	$2.1113e+06$	$7.7417e+04$
		240	$6.7389e+04$	$2.1419e+04$
OP5	300	1	$8.4961e+05$	$2.1281e+04$
		60	$1.4309e+04$	$8.5621e+03$
	600	1	$1.7740e+06$	$2.6561e+04$
		120	$4.8479e+04$	$2.0343e+04$
	900	1	$2.7107e+06$	$2.9989e+04$
		180	$9.4577e+04$	$1.9779e+04$
	1200	1	$3.6601e+06$	$3.2939e+04$
		240	$1.5937e+05$	$2.5546e+04$

populations:

$$M = \Big(\underbrace{x_{g1}^{[1]}, \ldots, x_{g,n_1}^{[1]}}_{x_g^{[1]} \text{ of } P_1}, \underbrace{x_{g1}^{[2]}, \ldots, x_{g,n_2}^{[2]}}_{x_g^{[2]} \text{ of } P_2}, \ldots, \underbrace{x_{g1}^{[K]}, \ldots, x_{g,n_K}^{[K]}}_{x_g^{[K]} \text{ of } P_K} \Big)^{\top}.$$

Therefore, by definition, the buffer constitutes the best position ever attained by the algorithm, i.e., it is the best obtained approximation of the actual minimizer.

Instead of the best from each subpopulation, a randomly selected individual could be alternatively used. This scheme would result in a COMDE approach with slower convergence but higher diversity. Clearly, the type of buffer update can affect the convergence properties of the algorithm substantially. Also, in some approaches, a restart of the subpopulations is performed to avoid the rapid diversity loss caused by their small sizes. COMDE does not use restart because DE is a greedy algorithm that stores its best positions in the population; thus, a population restart would destroy all information obtained in previous iterations.

Table 6: Hypothesis testing for TP1.

Oper.	Dim.	Improvement	p–value	Decision
OP1	300	38.6%	$1.6132e-10$	Reject
	600	57.7%	$3.0199e-11$	Reject
	900	66.5%	$3.0199e-11$	Reject
	1200	70.2%	$3.0199e-11$	Reject
OP2	300	71.3%	$3.0199e-11$	Reject
	600	80.5%	$3.0199e-11$	Reject
	900	83.1%	$3.0199e-11$	Reject
	1200	85.9%	$3.0199e-11$	Reject
OP3	300	17.4%	$2.6784e-06$	Reject
	600	58.5%	$3.0199e-11$	Reject
	900	69.2%	$3.0199e-11$	Reject
	1200	73.3%	$3.0199e-11$	Reject
OP4	300	92.1%	$3.0199e-11$	Reject
	600	96.8%	$3.0199e-11$	Reject
	900	96.6%	$3.0199e-11$	Reject
	1200	96.8%	$3.0199e-11$	Reject
OP5	300	98.3%	$3.0199e-11$	Reject
	600	97.3%	$3.0199e-11$	Reject
	900	96.5%	$3.0199e-11$	Reject
	1200	95.6%	$3.0199e-11$	Reject

The COMDE algorithm is reported in pseudocode in Table 1, and it is illustrated on high–dimensional instances of widely used benchmark problems in the following Section.

4. EXPERIMENTAL RESULTS

COMDE was applied on high–dimensional instances of the following widely used test problems:

TEST PROBLEM 1 (TP1 - Sphere) [15]. This n–dimensional problem is defined as:

$$f(x) = \sum_{i=1}^{n} x_i^2. \tag{7}$$

It has a global minimizer, $x^* = (0, \ldots, 0)^{\top}$, with $f(x^*) = 0$.

TEST PROBLEM 2 (TP2 - Generalized Rosenbrock) [15]. This n–dimensional problem is defined as:

$$f(x) = \sum_{i=1}^{n-1} \Big(100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \Big). \tag{8}$$

It has a global minimizer, $x^* = (1, \ldots, 1)^{\top}$, with $f(x^*) = 0$.

TEST PROBLEM 3 (TP3 - Rastrigin) [15]. This n–dimensional problem is defined as:

$$f(x) = 10n + \sum_{i=1}^{n} \big(x_i^2 - 10\cos(2\pi x_i) \big). \tag{9}$$

It has a global minimizer, $x^* = (0, \ldots, 0)^{\top}$, with $f(x^*) = 0$.

TEST PROBLEM 4 (TP4 - Griewank) [15]. This n–dimensional problem is defined as:

$$f(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1. \tag{10}$$

Table 7: Results for TP2.

Oper.	Dim.	K	Mean	StD
OP1	300	1	$1.6130e+08$	$4.0454e+07$
		60	$1.2478e+08$	$4.4738e+07$
	600	1	$6.6025e+08$	$9.9172e+07$
		120	$2.5380e+08$	$6.8593e+07$
	900	1	$1.2624e+09$	$1.5150e+08$
		180	$3.7215e+08$	$6.9986e+07$
	1200	1	$1.9992e+09$	$2.3594e+08$
		240	$5.1075e+08$	$8.6568e+07$
OP2	300	1	$1.1266e+09$	$1.0802e+08$
		60	$1.7087e+08$	$3.9198e+07$
	600	1	$7.2234e+09$	$4.5945e+08$
		120	$4.0705e+08$	$6.5600e+07$
	900	1	$1.2507e+10$	$3.8169e+08$
		180	$6.2499e+08$	$9.1419e+07$
	1200	1	$1.6998e+10$	$1.7632e+08$
		240	$8.7136e+08$	$8.9369e+07$
OP3	300	1	$4.2435e+07$	$1.2132e+07$
		60	$4.6087e+07$	$1.9138e+07$
	600	1	$2.4110e+08$	$3.6450e+07$
		120	$8.7002e+07$	$2.2511e+07$
	900	1	$5.6231e+08$	$8.6604e+07$
		180	$1.2340e+08$	$2.7540e+07$
	1200	1	$1.0265e+09$	$1.5032e+08$
		240	$1.6986e+08$	$2.3775e+07$
OP4	300	1	$4.3794e+08$	$6.4088e+07$
		60	$2.3383e+07$	$3.6772e+07$
	600	1	$2.5769e+09$	$2.0915e+08$
		120	$4.0509e+07$	$4.7292e+07$
	900	1	$5.5220e+09$	$4.4165e+08$
		180	$6.4051e+07$	$4.8790e+07$
	1200	1	$8.9109e+09$	$6.9988e+08$
		240	$8.5344e+07$	$3.7267e+07$
OP5	300	1	$3.8045e+09$	$1.1274e+08$
		60	$2.3442e+07$	$2.2751e+07$
	600	1	$8.0603e+09$	$1.9497e+08$
		120	$7.4900e+07$	$3.5246e+07$
	900	1	$1.2528e+10$	$1.3650e+08$
		180	$1.0436e+08$	$4.0071e+07$
	1200	1	$1.6951e+10$	$3.0242e+08$
		240	$2.3820e+08$	$8.4809e+07$

It has a global minimizer, $x^* = (0, \ldots, 0)^\top$, with $f(x^*) = 0$.

Test Problem 5 (TP5 - Ackley) [1]. This n–dimensional problem is defined as:

$$f(x) = 20 + \exp(1) - 20 \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right). \quad (11)$$

It has a global minimizer, $x^* = (0, \ldots, 0)^\top$, with $f(x^*) = 0$.

Each test problem was considered for dimensions, $n = 300$, 600, 900, and 1200. The corresponding n–dimensional search spaces are reported in Table 2. COMDE divides candidate solution vectors in K, 5–dimensional subcomponents and uses a subpopulation of 6 individuals on each. Hence, using the notation of Section 3, it follows that $n_k = 5$ and $N_k = 6$ for all $k = 1, 2, \ldots, K$, and $K = n/5$; therefore, the ratio n_k/N_k remains smaller than 1 for all subpopulations. For example, in the 600–dimensional case, the problem is divided in $600/5 = 120$ subcomponents; thus, 120 subpopulations are used, each consisting of six 5–dimensional individuals. The number of subpopulations, as well as the total number of individuals used per problem dimension are reported in Table 3.

Table 8: Hypothesis testing for TP2.

Oper.	Dim.	Improvement	p–value	Decision
OP1	300	22.6%	$8.5641e-04$	Reject
	600	61.6%	$3.0199e-11$	Reject
	900	70.5%	$3.0199e-11$	Reject
	1200	74.5%	$3.0199e-11$	Reject
OP2	300	84.8%	$3.0199e-11$	Reject
	600	94.4%	$3.0199e-11$	Reject
	900	95.0%	$3.0199e-11$	Reject
	1200	94.9%	$3.0199e-11$	Reject
OP3	300	−8.6%	$5.8945e-01$	Accept
	600	63.9%	$3.0199e-11$	Reject
	900	78.1%	$3.0199e-11$	Reject
	1200	83.5%	$3.0199e-11$	Reject
OP4	300	94.7%	$3.0199e-11$	Reject
	600	98.4%	$3.0199e-11$	Reject
	900	98.8%	$3.0199e-11$	Reject
	1200	99.0%	$3.0199e-11$	Reject
OP5	300	99.4%	$3.0199e-11$	Reject
	600	99.1%	$3.0199e-11$	Reject
	900	99.2%	$3.0199e-11$	Reject
	1200	98.6%	$3.0199e-11$	Reject

All DE operators, OP1–OP5, defined by Eqs. (1)–(5), were considered in the experiments. A maximum number of 1000 iterations was allowed for each subpopulation in all cases. We must notice that iterations are performed concurrently for all subpopulations. Thus, COMDE has significant parallelization capabilities, since each subpopulation can be assigned to a different processor, while the buffer update can be either synchronous or asynchronous. Regarding the DE parameters, the common setting, $F = 0.5$, $CR = 0.7$, was used for all subpopulations. All parameter values are summarized in Table 4. We must notice that parameters were arbitrarily set to reasonable values without any further fine–tuning that could enhance the algorithm's performance.

For each test problem, operator, and dimension, 30 independent experiments were performed. At each experiment, the best solution achieved after 1000 iterations was recorded along with its function value. The obtained function values were analyzed statistically, in terms of their mean value and standard deviation averaged over the 30 experiments. For comparison purposes, the experiments were repeated also for the standard DE, using a single population with the same parameters as COMDE. In order to have fair comparisons, the population size of standard DE was set equal to the total number of individuals employed by all subpopulations in COMDE per case. This number is reported in the second column of Table 3. DE was allowed to perform the same number of iterations as COMDE, and its performance was also statistically analyzed.

Table 9: **Results for TP3.**

Oper.	Dim.	K	Mean	StD
OP1	300	1	$1.6540e+03$	$1.0024e+02$
		60	$1.6969e+03$	$9.6112e+01$
	600	1	$4.2253e+03$	$2.2691e+02$
		120	$3.3438e+03$	$1.6715e+02$
	900	1	$7.1628e+03$	$2.6416e+02$
		180	$5.0532e+03$	$2.1358e+02$
	1200	1	$1.0174e+04$	$3.8489e+02$
		240	$6.6904e+03$	$1.6206e+02$
OP2	300	1	$3.9075e+03$	$7.0155e+01$
		60	$1.4579e+03$	$1.1371e+02$
	600	1	$8.5470e+03$	$1.0257e+02$
		120	$3.4638e+03$	$1.2190e+02$
	900	1	$1.3257e+04$	$1.2749e+02$
		180	$5.6513e+03$	$1.7424e+02$
	1200	1	$1.7929e+04$	$2.1022e+02$
		240	$7.8643e+03$	$2.1178e+02$
OP3	300	1	$2.2727e+03$	$8.2036e+02$
		60	$1.1791e+03$	$6.4883e+01$
	600	1	$3.0793e+03$	$8.9632e+02$
		120	$2.3899e+03$	$1.0092e+02$
	900	1	$5.3564e+03$	$2.1050e+02$
		180	$3.6284e+03$	$1.1715e+02$
	1200	1	$8.0055e+03$	$2.5775e+02$
		240	$4.8156e+03$	$1.6551e+02$
OP4	300	1	$3.9327e+03$	$8.4663e+01$
		60	$1.4660e+03$	$7.2126e+01$
	600	1	$8.4721e+03$	$1.0971e+02$
		120	$3.1654e+03$	$1.3256e+02$
	900	1	$1.3020e+04$	$1.5311e+02$
		180	$4.9847e+03$	$1.7586e+02$
	1200	1	$1.7618e+04$	$2.1850e+02$
		240	$6.6017e+03$	$2.7700e+02$
OP5	300	1	$4.9805e+03$	$4.7388e+01$
		60	$7.2558e+02$	$9.7278e+01$
	600	1	$1.0285e+04$	$1.0541e+02$
		120	$1.9104e+03$	$1.5805e+02$
	900	1	$1.5690e+04$	$1.0655e+02$
		180	$3.1601e+03$	$1.7140e+02$
	1200	1	$2.1033e+04$	$1.2561e+02$
		240	$4.5055e+03$	$2.1214e+02$

In addition, hypothesis tests were conducted to ensure statistical significance of the derived conclusions. Therefore, for each test problem, COMDE was compared against DE using the nonparametric Wilcoxon rank–sum test [3] with the null hypothesis that the two samples of function values, obtained by COMDE and DE in 30 experiments, come from identical continuous distributions with equal medians, against the alternative of different medians. The decision for acceptance or rejection of the null hypothesis in a 95% level of significance, as well as the corresponding p–value, were recorded for each test problem. Besides that, the performance improvement percentage between COMDE and DE, in terms of the obtained solution values averaged over the 30 experiments, was computed for all cases.

All results and statistical tests are reported in Tables 5–14. More specifically, for each test problem, operator, and dimension, the mean value and standard deviation of the obtained solution values after 1000 iterations, in the 30 independent experiments, are reported both for the COMDE (table rows with $K > 1$) and the standard DE (table rows with $K = 1$). Also, the p–values and decision of hypothesis testing are reported per problem and operator, along with the improvement attained by COMDE, with negative values denoting worse performance of COMDE against DE.

Table 10: Hypothesis testing for TP3.

Oper.	Dim.	Improvement	p–value	Decision
OP1	300	−2.6%	$5.9428e-02$	Accept
	600	20.9%	$3.3384e-11$	Reject
	900	29.5%	$3.0199e-11$	Reject
	1200	34.2%	$3.0199e-11$	Reject
OP2	300	62.7%	$3.0199e-11$	Reject
	600	59.5%	$3.0199e-11$	Reject
	900	57.4%	$3.0199e-11$	Reject
	1200	56.1%	$3.0199e-11$	Reject
OP3	300	48.1%	$1.9527e-03$	Reject
	600	22.4%	$3.3384e-11$	Reject
	900	32.3%	$3.0199e-11$	Reject
	1200	39.8%	$3.0199e-11$	Reject
OP4	300	62.7%	$3.0199e-11$	Reject
	600	62.6%	$3.0199e-11$	Reject
	900	61.7%	$3.0199e-11$	Reject
	1200	62.5%	$3.0199e-11$	Reject
OP5	300	85.4%	$3.0199e-11$	Reject
	600	81.4%	$3.0199e-11$	Reject
	900	79.9%	$3.0199e-11$	Reject
	1200	78.6%	$3.0199e-11$	Reject

As a first observation, we can see that COMDE has significantly improved performance for all operators. Especially for OP4 and OP5, there was a tremendous improvement over 90% in three test problems (TP1, TP2, and TP4), while, for the rest problems, their improvement remained the highest among all operators. Remarkable improvement was observed also for the OP2 operator. If we take a closer look at the aforementioned three operators, defined in Eqs. (2), (4), and (5), we will observe that they employ the highest number of randomly selected individuals from the population. Indeed, OP2 consists of one difference vector that combines three randomly selected individuals, in contrast to the similar operator OP1, which also has one difference vector but with two randomly selected individuals. Similarly, both OP4 and OP5 consist of two difference vectors, involving four and five randomly selected individuals, respectively, while OP3, which also uses two difference vectors, involves only two randomly selected individuals. This indicates the increasingly beneficial effect of COMDE when the number of involved randomly selected individuals in the operators is increased.

As a second observation, we see that the two most benefited operators, OP4 and OP5, of COMDE also exhibit the highest overall performance in all test problems and dimensions, in terms of the reported mean values. Indeed, OP4 is the best for all dimensions in TP2 and TP4, while the same holds for OP5 in TP3 and TP5. Only in TP1, OP5 was the best among all operators for the 300–dimensional case, while OP4 was the best for all other dimensions. This verifies the instrumental contribution of the COMDE approach to these operators.

Table 11: Results for TP4.

Oper.	Dim.	K	Mean	StD
OP1	300	1	$1.2399e+03$	$1.7837e+02$
		60	$8.2556e+02$	$1.4660e+02$
	600	1	$4.1313e+03$	$2.9879e+02$
		120	$1.6579e+03$	$1.2883e+02$
	900	1	$7.4216e+03$	$4.8889e+02$
		180	$2.4738e+03$	$2.1885e+02$
	1200	1	$1.0853e+04$	$6.1849e+02$
		240	$3.3135e+03$	$2.4706e+02$
OP2	300	1	$2.8930e+03$	$1.4252e+02$
		60	$8.5437e+02$	$1.8164e+02$
	600	1	$9.2399e+03$	$4.0236e+02$
		120	$1.9105e+03$	$2.7263e+02$
	900	1	$1.6908e+04$	$7.1325e+02$
		180	$2.7540e+03$	$2.9849e+02$
	1200	1	$2.7056e+04$	$1.8457e+03$
		240	$3.7251e+03$	$3.6052e+02$
OP3	300	1	$6.2043e+02$	$9.2146e+01$
		60	$4.6329e+02$	$6.8827e+01$
	600	1	$2.3797e+03$	$1.7747e+02$
		120	$9.9943e+02$	$1.0220e+02$
	900	1	$4.6113e+03$	$4.1597e+02$
		180	$1.4127e+03$	$1.0873e+02$
	1200	1	$7.3171e+03$	$5.6150e+02$
		240	$1.9647e+03$	$1.2357e+02$
OP4	300	1	$1.6291e+03$	$1.3325e+02$
		60	$1.2757e+02$	$7.3330e+01$
	600	1	$7.3704e+03$	$2.9143e+02$
		120	$2.4710e+02$	$1.1863e+02$
	900	1	$1.3037e+04$	$5.4661e+02$
		180	$4.2938e+02$	$1.2733e+02$
	1200	1	$1.8819e+04$	$6.7970e+02$
		240	$5.7269e+02$	$1.6323e+02$
OP5	300	1	$7.7043e+03$	$1.4479e+02$
		60	$1.2792e+02$	$8.5594e+01$
	600	1	$1.5848e+04$	$3.0308e+02$
		120	$4.8975e+02$	$1.9774e+02$
	900	1	$2.4276e+04$	$3.3834e+02$
		180	$9.5644e+02$	$2.2423e+02$
	1200	1	$3.2922e+04$	$3.1154e+02$
		240	$1.4481e+03$	$2.6392e+02$

We shall also note that the null hypothesis was rejected in all but two cases, namely the 300–dimensional cases of OP3 in TP2 and OP1 in TP3. These two exceptions are both characterized by a slight worsening of the COMDE performance compared to the corresponding standard DE with respect to the reported mean values, although this is not accompanied by statistical significance.

Regarding their robustness, as it is expressed by the reported standard deviations, OP3 can be distinguished as the most robust operator, especially for higher dimensions. As we can see in Table 5 for TP1, the COMDE version of OP3 has the smallest standard deviations among all operators of both DE and COMDE for $n \geqslant 600$. In the case of $n = 300$, its standard DE counterpart was the most robust, while for COMDE, OP5 had the smallest standard deviation.

The same holds for TP2, as reported in Table 7, except the case of $n = 300$, where OP3 exhibited the smallest standard deviations among all operators for both DE and COMDE. For the rest dimensions of TP2, COMDE under OP3 was the most robust. However, this is not the case for TP3 and TP5, with OP3 achieving in many cases the best standard deviations among all COMDE operators but not overall.

Table 12: Hypothesis testing for TP4.

Oper.	Dim.	Improvement	p–value	Decision
OP1	300	33.4%	$4.1997e-10$	Reject
	600	59.9%	$3.0199e-11$	Reject
	900	66.7%	$3.0199e-11$	Reject
	1200	69.5%	$3.0199e-11$	Reject
OP2	300	70.5%	$3.0199e-11$	Reject
	600	79.3%	$3.0199e-11$	Reject
	900	83.7%	$3.0199e-11$	Reject
	1200	86.2%	$3.0199e-11$	Reject
OP3	300	25.3%	$3.6459e-08$	Reject
	600	58.0%	$3.0199e-11$	Reject
	900	69.4%	$3.0199e-11$	Reject
	1200	73.1%	$3.0199e-11$	Reject
OP4	300	92.2%	$3.0199e-11$	Reject
	600	96.6%	$3.0199e-11$	Reject
	900	96.7%	$3.0199e-11$	Reject
	1200	97.0%	$3.0199e-11$	Reject
OP5	300	98.3%	$3.0199e-11$	Reject
	600	96.9%	$3.0199e-11$	Reject
	900	96.1%	$3.0199e-11$	Reject
	1200	95.6%	$3.0199e-11$	Reject

Summarizing the results, it is shown experimentally that COMDE can be a very promising approach, producing for all operators superior results than standard DE. There are only two exceptions to this observation, namely the 300–dimensional cases of OP3 in TP2 and OP1 in TP3. Both these operators involve the best individual of the population, which seems to be beneficial for the specific problems, although this result is not statistically significant in a 95% level. Nevertheless, in higher–dimensional cases even this advantage was surpassed by COMDE, which has shown the potential to occupy a salient place among the alternatives for high–dimensional problems.

5. CONCLUSIONS

COMDE, an approach that combines cooperative with Micro–DE was introduced and experimentally assessed on widely used test problems for dimensions ranging from 300 up to 1200. The proposed approach was also compared to the standard DE algorithm under the five most common DE operators. Preliminary results are very encouraging, exhibiting significant improvement in performance of all operators, especially as problem dimension increases.

Further research is needed to fully reveal the potential of COMDE and identify possible drawbacks in cases where typical cooperative approaches meet obstacles, such as the case of problems with highly–correlated coordinate directions. Nevertheless, COMDE has shown to be a valuable tool in high–dimensional cases regardless of the employed operator. Different DE parameter settings shall also be considered in future works to determine possible effects on the algorithm's performance.

Table 13: Results for TP5.

Oper.	Dim.	K	Mean	StD
OP1	300	1	$1.5861e+01$	$4.5127e-01$
		60	$1.3792e+01$	$5.1525e-01$
	600	1	$1.7268e+01$	$2.9403e-01$
		120	$1.3651e+01$	$3.6977e-01$
	900	1	$1.7834e+01$	$2.2379e-01$
		180	$1.3741e+01$	$4.2847e-01$
	1200	1	$1.8119e+01$	$1.4448e-01$
		240	$1.3794e+01$	$3.0914e-01$
OP2	300	1	$1.7617e+01$	$2.0652e-01$
		60	$1.3949e+01$	$6.2472e-01$
	600	1	$1.9150e+01$	$9.8862e-02$
		120	$1.4605e+01$	$3.6721e-01$
	900	1	$1.9587e+01$	$8.6341e-02$
		180	$1.4902e+01$	$3.2222e-01$
	1200	1	$1.9803e+01$	$7.1190e-02$
		240	$1.5002e+01$	$2.6469e-01$
OP3	300	1	$1.2653e+01$	$4.0916e-01$
		60	$1.1879e+01$	$4.4926e-01$
	600	1	$1.4987e+01$	$2.1141e-01$
		120	$1.1866e+01$	$4.2328e-01$
	900	1	$1.5832e+01$	$2.0624e-01$
		180	$1.1812e+01$	$3.5390e-01$
	1200	1	$1.6472e+01$	$2.2578e-01$
		240	$1.1767e+01$	$2.7186e-01$
OP4	300	1	$1.6207e+01$	$2.6336e-01$
		60	$1.0769e+01$	$1.0645e+00$
	600	1	$1.9011e+01$	$9.2298e-02$
		120	$1.0726e+01$	$7.9374e-01$
	900	1	$1.9504e+01$	$9.6574e-02$
		180	$1.0997e+01$	$5.4687e-01$
	1200	1	$1.9684e+01$	$7.3283e-02$
		240	$1.0927e+01$	$4.1879e-01$
OP5	300	1	$2.0282e+01$	$4.4779e-02$
		60	$6.6546e+00$	$1.8917e+00$
	600	1	$2.0447e+01$	$1.6909e-02$
		120	$8.3028e+00$	$9.1099e-01$
	900	1	$2.0487e+01$	$1.0475e-02$
		180	$9.4568e+00$	$8.4387e-01$
	1200	1	$2.0506e+01$	$1.0876e-02$
		240	$1.0048e+01$	$5.1338e-01$

Table 14: Hypothesis testing for TP5.

Oper.	Dim.	Improvement	p–value	Decision
OP1	300	13.0%	$3.0199e-11$	Reject
	600	20.9%	$3.0199e-11$	Reject
	900	23.0%	$3.0199e-11$	Reject
	1200	23.9%	$3.0199e-11$	Reject
OP2	300	20.8%	$3.0199e-11$	Reject
	600	23.7%	$3.0199e-11$	Reject
	900	23.9%	$3.0199e-11$	Reject
	1200	24.2%	$3.0199e-11$	Reject
OP3	300	6.1%	$1.5964e-07$	Reject
	600	20.8%	$3.0199e-11$	Reject
	900	25.4%	$3.0199e-11$	Reject
	1200	28.6%	$3.0199e-11$	Reject
OP4	300	33.6%	$3.0199e-11$	Reject
	600	43.6%	$3.0199e-11$	Reject
	900	43.6%	$3.0199e-11$	Reject
	1200	44.5%	$3.0199e-11$	Reject
OP5	300	67.2%	$3.0180e-11$	Reject
	600	59.4%	$3.0199e-11$	Reject
	900	53.8%	$3.0199e-11$	Reject
	1200	51.0%	$3.0199e-11$	Reject

6. REFERENCES

[1] D. H. Ackley. *A Connectionist Machine for Genetic Hillclimbing*. Kluwer, Boston, 1987.

[2] H. G. Cobb. Is the genetic algorithm a cooperative learner? In *Foundations of Genetic Algorithms 2*, pages 277–296. Morgan Kaufmann, 1992.

[3] W. J. Conover. *Practical Nonparametric Statistics*. Wiley, 1998.

[4] M. El-Abd. *Cooperative Models of Particle Swarm Optimizers*. PhD thesis, Dept. Elect. Comput. Eng., Univ. Waterloo, Waterloo, Ontario, Canada, 2008.

[5] T. Huang and A. S. Mohan. Micro–particle swarm optimizer for solving high dimensional optimization problems. *Applied Mathematics and Computation*, 181(2):1148–1154, 2006.

[6] M. Köppen, K. Franke, and R. Vicente-Garcia. Tiny GAs for image processing applications. *IEEE Computational Intelligence Magazine*, 1(2):17–26, 2006.

[7] O. Olorunda and A. P. Engelbrecht. Differential evolution in high–dimensional search spaces. In *Proc. IEEE CEC'07*, pages 1934–1941, Singapore, 2007.

[8] M. A. Potter and K. De Jong. A cooperative coevolutionary approach to function optimization. In Y. Davidor and H.-P. Schwefel, editors, *Proc. PPSN'94*, pages 249–257. Springer–Verlag, 1994.

[9] M. A. Potter and K. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evol. Comput.*, 8(1):1–29, 2000.

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

[11] S. Rahnamayan and H. R. Tizhoosh. Image thresholding using micro opposition-based differential evolution (micro-ODE). In *Proc. IEEE CEC'08*, pages 1409–1416, Hong Kong, 2008.

[12] Y.-J. Shi, H.-F. Teng, and Z.-Q. Li. Cooperative co–evolutionary differential evolution for function optimization. In *Lecture Notes in Computer Science*, volume 3611, pages 1080–1088. Springer, 2005.

[13] R. E. Smith, S. Forrest, and A. S. Perelson. Searching for diverse, cooperative populations with genetic algorithms. *Evol. Comput.*, 1(2):127–149, 1993.

[14] D. Sofge, K. De Jong, and A. Schultz. A blended population approach to cooperative coevoultion for decomposition of complex problems. In *Proc. IEEE CEC'02*, pages 413–418, 2002.

[15] R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Opt.*, 11:341–359, 1997.

[16] F. Van den Bergh and A. P. Engelbrecht. A cooperative approach to particle swarm optimization. *IEEE Trans. Evol. Comput.*, 8(3):225–239, 2004.