A Novel Center-based Differential Evolution Algorithm

Seyed Jalaleddin Mousavirad, Shahryar Rahnamayan, SMIEEE
Nature-Inspired Computational Intelligence (NICI) Lab
Department of Electrical, Computer, and Software Engineering
Ontario Tech University, Oshawa, Canada
jalalmoosavirad@gmail.com, shahryar.rahnamayan@uoit.ca

***Abstract*—Differential Evolution (DE) algorithm has been shown notable performance in solving complex optimization problems. In recent years, some variants of the DE algorithm have been proposed based on the concept of center-based sampling strategy. To the best of our knowledge, the related papers employed center-based sampling for population initialization or as the base vector in mutation operator. In fact, they were operation-level approaches applied during the optimization process, and none of them was about proposing a population-level approach to utilize center-based sampling to accelerate convergence rate of algorithms. This paper proposes a novel center-based sampling scheme for the DE algorithm that utilizes center-based sampling as a member of the population. In our scheme, one candidate solution is the center of the best candidate solutions, while other individuals in the population behave similarly to the standard DE algorithm. The center-based candidate solution is not updated using standard operators and is set to the center in each iteration. To validate our scheme, we benchmark our algorithm on CEC-2017 benchmark functions with three dimensions of 30, 50, and 100. Also, we design some experiments to analyze the behavior of the proposed center-based scheme. Our experiments demonstrate a significant improvement of the proposed algorithm on the majority of benchmark functions.**

***Index Terms*—Differential algorithm, center-based sampling, optimization, population, CEC-2017**

I. Introduction

In recent years, many real-world problems have been reformulated as an optimization problem. In an optimization problem, an objective function should be maximized or minimized regarding x as a decision variable vector.

Conventional optimization algorithms such as gradient-based approaches suffer from some drawbacks such as getting trapped in a local optimum and being sensitive to initial conditions [1]. In order to cope with these problems, population-based metaheuristic algorithms such as genetic algorithm (GA) [2], particle swarm optimization (PSO) [3], artificial bee colony (ABC) [4], and human mental search (HMS) [5] can be employed. Population-based algorithms are problem-independent algorithms with some stochastic operators which have been extensively used to solve real-world optimization problems due to their global search capability, robustness, and potential capability of parallelism [6]–[10].

Differential Evolution (DE) algorithm [11] is a robust population-based metaheuristic algorithm introduced by Storn and Price in 1995. The DE algorithm has three leading operators, namely, mutation, crossover, and selection. The aim of mutation is the creation of new candidate solutions (mutant vectors) based on scaled differences among candidate solutions. Crossover operator integrates the mutant vector with a target vector, while the selection operator selects better candidate solution between the target candidate solution and its offspring based on their objective function values and a greedy selection scheme.

Numerous studies have been conducted on DE algorithm for improving it using hybridization of DE with other algorithms [12], presenting new mutation and/or crossover operators [13], [14], and ensemble of different schemes [15]. Furthermore, DE algorithm is employed in different domains of optimization such as multi-objective optimization [16], many-objective optimization [17], and multi-modal optimization [18]. Additionally, DE is employed to solve various real-world optimization problems in different scientific fields such as image processing [19] and scheduling [20].

Center-based sampling strategy has been introduced by Rahnamayan and Wang in 2009 [21]. They investigated the likelihood of closeness to an unknown solution for a center point is more than a uniformly generated random point. Center-based sampling can be seen in several improved DE algorithms to enhance its performance. Hanan et al. [22] proposed a center-based mutation scheme for DE algorithm. In their work, the center of three randomly selected candidate solutions is computed as the mean value of a normal distribution. Then, a new candidate solution is generated based on the normal distribution and set as the base vector in the mutation operator. In another work [23], SHADE algorithm (Success-History Based Parameter Adaptation for Differential Evolution) is improved by replacing the base vector by a normal distribution which its mean is the center of three randomly selected candidate solutions. [24] proposed a multiple center-based DE algorithm, which in each generation, the current population is

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

divided into two distinct groups. In the first group, standard DE mutation is applied, while in the second group, a center-based mutation is employed. Liu et al. [25] computed the center of subpopulation and the center of population simultaneously. Then, they proposed two mutation schemes, and finally, the better-generated candidate solution is selected to replace the target vector.

Center-based sampling is also used for population initialization. Mahdavi et al. [26] proposed three center-based schemes, including center-based normal distribution sampling (CNS), central golden region (CGR), and hybrid random-center normal distribution sampling (HRCN) for population initialization. CGR outperformed other schemes. Khanum et al. [27] proposed a centroid population initialization for JADE algorithm. First, $3 \times NP$ candidate solutions are randomly created (P). Then, a new population as the initial population is generated based on the mean of three candidate solutions selected from P. In another work, Mousavirad et al. [28] proposed a center-based Latin Hypercube Initialization along with opposition-based DE to tackle deceptive optimization. In their algorithm, the search space is divided into cells and the center of each cell is selected as one of the initial members of population.

To the best of our knowledge, the center-based sampling strategy has been used for mutation and population initialization in DE algorithm so far. Our scheme benefits from a different mechanism to employ center-based sampling in DE algorithm, but in the population level, which makes it usable in other population-based algorithms. In our proposed scheme, the center point is a member of population. In other words, center point is injected into the current population as a new candidate solution. The experimental results show that it has a significant ability to direct the whole population towards the global optimum.

The remainder of this paper is organized as follows. Section II introduces DE algorithm briefly, while Section III demonstrates the concept of center-based sampling. The proposed center-based DE algorithm is explained in Section IV. To verify the performance of our proposed algorithm, Section V provides a discussion on the experimental results. Finally, the paper is concluded.

II. Differential evolution

Differential Evolution (DE) [11] is a simple yet powerful population-based metaheuristic algorithm for solving non-linear, non-convex, multi-modal, and non-differentiable optimization problems which has shown a substantial performance on various complex problems. After population initialization, DE updates the current population based on three main operators, namely, crossover, mutation, and selection.

Mutation generates a mutant vector, $\overrightarrow{v_i} = (v_{i,1}, v_{i,2}, ..., v_{i,D})$, for each candidate solution as

$$\overrightarrow{v_i} = \overrightarrow{x_{r1}} + F * (\overrightarrow{x_{r2}} - \overrightarrow{x_{r3}}) \quad (1)$$

where $\overrightarrow{x_{ri}}$ is a randomly selected candidate solution ($\overrightarrow{x_{ri}} \neq \overrightarrow{x_{rj}}$ for each i and j) and F is scaling factor.

Crossover operator integrates the mutant vector with the target (parent) vector. To this end, binomial crossover, as a well-known strategy, is used which is defined as

$$u_{i,j} = \begin{cases} v_{i,j}, & rand(0,1) \leq CR \text{ } or \text{ } j == j_{rand} \\ x_{i,j}, & \text{otherwise.} \end{cases} \quad (2)$$

where $\overrightarrow{u_i}$ is called trial vector, CR is a constant called crossover rate, and j_{rand} is a random integer number between 1 and the number of dimensions.

Eventually, the aim of selection operator is to select a candidate solution between the trial vector, u_i, target vector, x_i, based on their objective function values, and the corresponding minimum one (for a minimization problem) should be selected for the next generation.

III. Center-based sampling strategy

Rahnamayan and Wang [21] introduced the concept of center-based sampling. They investigated the likelihood of closeness to an unknown solution for a center point and random point and indicated that the likelihood of points being closer to an unknown solution is much more significant towards the center of the search space compared to randomly generated points. The Monte-Carlo simulations indicated that as the candidate solution gets closer to the center, the likelihood of closeness to the unknown solution has risen sharply. In addition to Monte-Carlo simulation, Rahnamayan and Wang [29] presented mathematical proof for center-based sampling to support their numerical results.

For a search space in the interval of [a,b], the center is defined for dimensions 1 and n as follows:

For 1-dimension:

$$c = \frac{a+b}{2}, \quad (3)$$

For n-dimensions:

$$c_i = \frac{a_i + b_i}{2}, \quad (4)$$

where $i = 1, ..., D$ and D is the search space dimensions.

From the literature, center-based sampling has been used to improve the performance of DE algorithm in two ways during population initialization [26]–[28] and as the base vector for mutation equation [21]–[23], while it seems to has the potential to use in various strategies.

IV. Proposed center-based scheme

In the current paper, a center-based candidate solution is created to explicitly as the center of the N best candidate solutions. In our proposed scheme (DE_Center_P), population is divided into two parts: $N_P - 1$ candidate solutions update their position as the standard mutation and crossover, while one candidate solution is responsible for maintaining the center of the N best candidate solutions. Algorithm 1 presents the proposed algorithm in the form of pseudo-code.

The algorithm commences with an initial population with N_P candidate solutions. The first $N_P - 1$ candidate solutions generate randomly with a uniform distribution. After calculating the objective functions for $N_P - 1$ candidate solutions,

Input : D: Dimension of problem, Max_{NFC}: Maximum number of function evaluations, N_P: Population size, F: Scaling factor, C_R: Crossover probability, N: The number of best candidate solutions
Output: $\overrightarrow{x^*}$: the best solution

Generate the initial population Pop uniform randomly for the first $N_P - 1$ candidate solutions;
Evaluate the fitness for each candidate solution;
Select three parents, $\overrightarrow{x_{i1}}$, $\overrightarrow{x_{i2}}$, and $\overrightarrow{x_{i3}}$, randomly from the current population, with $\overrightarrow{x_{i1}} \neq \overrightarrow{x_{i2}} \neq \overrightarrow{x_{i3}}$;
Select N best candidate solutions as $\overrightarrow{x_{b1}}$, $\overrightarrow{x_{b2}}$,...,$\overrightarrow{x_{bN}}$;
$Pop(N_P) = \frac{\overrightarrow{x_{b1}}+\overrightarrow{x_{b2}}+...+\overrightarrow{x_{bN}}}{N}$;
$NFE = N_P$;
while $NFE < MAX_{NFE}$ **do**
  for $i \leftarrow 1$ **to** $N_P - 1$ **do**
    Select three parents, $\overrightarrow{x_{i1}}$, $\overrightarrow{x_{i2}}$, and $\overrightarrow{x_{i3}}$, randomly from the current population, with $\overrightarrow{x_{i1}} \neq \overrightarrow{x_{i2}} \neq \overrightarrow{x_{i3}}$;
    $\overrightarrow{v_i} = \overrightarrow{x_{i1}} + F * (\overrightarrow{x_{i2}} - \overrightarrow{x_{i3}})$;
    for $j \leftarrow 0$ **to** D **do**
      if $rand_j[0,1] < C_R$ or $j == j_{rand}$ **then**
        $u_{i,j} = v_{i,j}$
      else
        $u_{i,j} = x_{i,j}$
      end
    end
    Calculate objective function of $\overrightarrow{u_i}$;
    if $f(\overrightarrow{u_i}) < f(\overrightarrow{x_i})$ **then**
      $\overrightarrow{x} \leftarrow \overrightarrow{u_i}$;
    else
      $\overrightarrow{x} \leftarrow \overrightarrow{x_i}$;
    end
    $Pop(i) \leftarrow \overrightarrow{x}$;
  end
  Select N best candidate solutions as $\overrightarrow{x_{b1}}$, $\overrightarrow{x_{b2}}$,...,$\overrightarrow{x_{bN}}$;
  $Pop(N_P) = \frac{\overrightarrow{x_{b1}}+\overrightarrow{x_{b2}}+...+\overrightarrow{x_{bN}}}{N}$;
  $NFE = NFE + N_P$;
end
$\overrightarrow{x^*} \leftarrow$ the best candidate solution in pop

Algorithm 1: The proposed algorithm in the form of pseudo-code

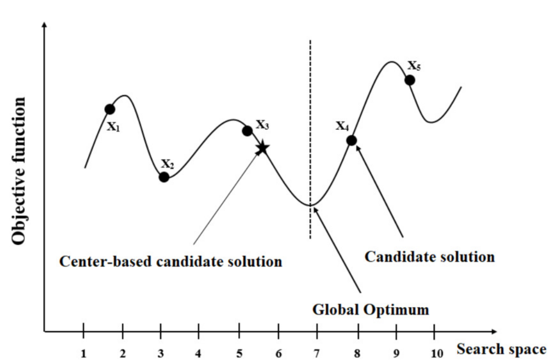


Fig. 1: Visualization of center-based candidate solution for a one-dimensional problem.

TABLE I: Results of the algorithms on CEC-2017 benchmark functions for D=30.

Functions		DE	DE_Center_P	IAR
F1	Mean	**8.68E+03**	**4.82E+03≈**	**1.82**
	Std	1.47E+04	4.97E+03	
F2	Mean	3.86E+19	**2.12E+12‡**	**1.82E+07**
	Std	1.93E+20	5.26E+12	
F3	Mean	9.43E+03	**1.68E+03‡**	**6.62**
	Std	8.94E+03	9.45E+02	
F4	Mean	**4.35E+02**	4.83E+02†	0.42
	Std	2.08E+01	2.67E+01	
F5	Mean	6.85E+02	**5.42E+02‡**	**4.40**
	Std	9.19E+00	1.96E+01	
F6	Mean	**6.00E+02**	**6.00E+02≈**	1.00
	Std	1.60E-04	4.41E-02	
F7	Mean	9.12E+02	**7.82E+02‡**	**2.59**
	Std	1.71E+01	1.79E+01	
F8	Mean	9.89E+02	**8.34E+02‡**	**5.56**
	Std	1.21E+01	9.06E+00	
F9	Mean	**9.00E+02**	**9.00E+02≈**	1.00
	Std	4.94E-01	1.50E+01	
F10	Mean	8.76E+03	**4.63E+03‡**	**2.14**
	Std	3.83E+02	7.70E+02	
F11	Mean	**1.13E+03**	**1.14E+03≈**	0.75
	Std	2.07E+01	3.07E+01	
F12	Mean	2.55E+05	**3.86E+04‡**	**6.79**
	Std	3.62E+05	2.58E+04	
F13	Mean	**1.82E+04**	**7.55E+03≈**	**2.70**
	Std	2.15E+04	1.11E+04	
F14	Mean	1.47E+03	**1.45E+03‡**	**1.40**
	Std	6.76E+00	1.91E+01	
F15	Mean	1.61E+03	**1.56E+03‡**	**1.83**
	Std	8.88E+01	5.73E+01	
F16	Mean	3.09E+03	**2.50E+03‡**	**1.66**
	Std	2.67E+02	4.74E+02	
F17	Mean	2.17E+03	**1.93E+03‡**	**2.04**
	Std	2.01E+02	2.08E+02	
F18	Mean	**1.09E+04**	**1.25E+04≈**	0.85
	Std	8.97E+03	9.84E+03	
F19	Mean	**1.92E+03**	**1.95E+03≈**	0.40
	Std	5.71E+00	1.52E+02	
F20	Mean	**2.32E+03**	**2.32E+03≈**	1.00
	Std	2.28E+02	2.42E+02	
F21	Mean	2.48E+03	**2.34E+03‡**	**1.58**
	Std	7.34E+00	1.53E+01	
F22	Mean	9.99E+03	**3.00E+03‡**	**9.74**
	Std	3.09E+02	1.53E+03	
F23	Mean	2.83E+03	**2.69E+03‡**	**1.36**
	Std	1.09E+01	1.10E+01	
F24	Mean	3.01E+03	**2.86E+03‡**	**1.33**
	Std	9.53E+00	1.70E+01	
F25	Mean	**2.88E+03**	**2.88E+03≈**	1.00
	Std	1.16E+00	1.06E+01	
F26	Mean	5.26E+03	**3.99E+03‡**	**1.91**
	Std	2.03E+02	1.45E+02	
F27	Mean	**3.20E+03**	**3.20E+03≈**	1.00
	Std	1.32E-04	7.77E+00	
F28	Mean	3.30E+03	**3.22E+03‡**	**1.19**
	Std	1.75E-04	2.22E+01	
F29	Mean	3.83E+03	**3.46E+03‡**	**1.66**
	Std	2.48E+02	1.35E+02	
F30	Mean	**3.22E+03**	6.83E+03†	0.06
	Std	8.07E+00	1.79E+03	
w/t/l			18/10/2	mean IAR=2.26

N best candidate solutions are selected to generate the last candidate solution as

$$\overrightarrow{x_{center}} = \frac{\overrightarrow{x_{b1}} + ...\overrightarrow{x_{bi}} + ... + \overrightarrow{x_{bN}}}{N} \tag{5}$$

where $\overrightarrow{x_{bi}}$ is the i-th best candidate solution.

TABLE II: Results of the algorithms on CEC-2017 benchmark functions for D=50.

Functions		DE	DE_Center_P	IAR
F1	Mean	**5.07E+03**	**3.47E+03**≈	**1.47**
	Std	4.66E+03	3.72E+03	
F2	Mean	**1.02E+39**	1.52E+39†	0.67
	Std	4.84E+39	6.11E+39	
F3	Mean	2.41E+05	**6.64E+04**‡	**3.64**
	Std	5.65E+04	1.50E+04	
F4	Mean	**4.49E+02**	5.26E+02†	0.39
	Std	2.42E+01	5.16E+01	
F5	Mean	8.61E+02	**5.79E+02**‡	**4.57**
	Std	2.04E+01	2.00E+01	
F6	Mean	**6.00E+02**	**6.00E+02**≈	1.00
	Std	7.71E-02	5.00E-01	
F7	Mean	1.12E+03	**8.75E+02**‡	**2.40**
	Std	1.63E+01	4.10E+01	
F8	Mean	1.15E+03	**8.93E+02**‡	**3.76**
	Std	6.57E+01	2.21E+01	
F9	Mean	**9.09E+02**	1.04E+03†	0.06
	Std	1.04E+01	1.34E+02	
F10	Mean	1.54E+04	**8.39E+03**‡	**1.95**
	Std	3.99E+02	1.58E+03	
F11	Mean	**1.20E+03**	**1.20E+03**≈	1.00
	Std	6.55E+01	4.37E+01	
F12	Mean	1.96E+06	**9.84E+05**‡	**1.99**
	Std	1.47E+06	6.92E+05	
F13	Mean	**1.13E+04**	**9.25E+03**≈	**1.26**
	Std	1.39E+04	9.06E+03	
F14	Mean	**7.39E+03**	**5.49E+03**≈	**1.46**
	Std	1.20E+04	4.09E+03	
F15	Mean	3.24E+04	**6.93E+03**‡	**5.69**
	Std	4.17E+04	4.53E+03	
F16	Mean	4.69E+03	**3.25E+03**‡	**1.87**
	Std	5.37E+02	6.52E+02	
F17	Mean	3.34E+03	**2.98E+03**‡	**1.28**
	Std	3.78E+02	3.10E+02	
F18	Mean	1.09E+05	**5.11E+04**‡	**2.17**
	Std	6.76E+04	3.62E+04	
F19	Mean	**1.00E+04**	**8.32E+03**≈	**1.26**
	Std	1.39E+04	7.36E+03	
F20	Mean	**3.47E+03**	**3.31E+03**≈	**1.12**
	Std	3.53E+02	2.25E+02	
F21	Mean	2.66E+03	**2.38E+03**‡	**2.00**
	Std	1.70E+01	2.27E+01	
F22	Mean	1.67E+04	**9.17E+03**‡	**2.08**
	Std	3.92E+02	1.38E+03	
F23	Mean	3.07E+03	**2.82E+03**‡	**1.48**
	Std	3.71E+01	3.02E+01	
F24	Mean	3.28E+03	**2.99E+03**‡	**1.49**
	Std	1.60E+01	3.30E+01	
F25	Mean	**2.94E+03**	3.06E+03†	0.79
	Std	2.32E+01	3.11E+01	
F26	Mean	7.35E+03	**4.73E+03**‡	**2.23**
	Std	4.57E+02	2.06E+02	
F27	Mean	**3.20E+03**	3.38E+03†	0.74
	Std	1.30E-04	8.71E+01	
F28	Mean	**3.30E+03**	**3.30E+03**≈	1.00
	Std	1.67E-04	1.46E+01	
F29	Mean	5.04E+03	**3.60E+03**‡	**3.06**
	Std	3.66E+02	2.33E+02	
F30	Mean	**7.09E+03**	8.24E+05†	0.004
	Std	4.81E+03	1.40E+05	
w/t/l			16/8/4	mean IAR=1.80

TABLE III: Results of the algorithms on CEC-2017 benchmark functions for D=100.

Functions		DE	DE_Center_P	IAR
F1	Mean	**9.77E+03**	**4.63E+07**≈	**2.13**
	Std	1.21E+04	2.31E+08	
F2	Mean	7.36E+92	**7.21E+81**‡	**1.02E+11**
	Std	3.68E+93	3.60E+82	
F3	Mean	1.58E+06	**3.53E+05**‡	**4.48**
	Std	3.59E+05	4.44E+04	
F4	Mean	**5.94E+02**	6.86E+02†	0.68
	Std	5.55E+01	5.05E+01	
F5	Mean	1.21E+03	**7.78E+02**‡	**2.55**
	Std	3.11E+02	4.02E+01	
F6	Mean	**6.01E+02**	6.08E+02†	0.13
	Std	4.44E-01	2.64E+00	
F7	Mean	1.74E+03	**1.31E+03**‡	**1.70**
	Std	4.17E+01	9.72E+01	
F8	Mean	1.51E+03	**1.10E+03**‡	**2.37**
	Std	3.02E+02	5.95E+01	
F9	Mean	**1.91E+03**	4.15E+03†	0.31
	Std	1.49E+03	1.69E+03	
F10	Mean	3.28E+04	**2.06E+04**‡	**1.62**
	Std	5.47E+02	2.42E+03	
F11	Mean	3.50E+03	**1.71E+03**‡	**3.93**
	Std	1.22E+03	2.60E+02	
F12	Mean	6.72E+06	**3.23E+06**‡	**2.08**
	Std	4.00E+06	1.59E+06	
F13	Mean	**7.94E+03**	**6.54E+03**≈	**1.27**
	Std	9.64E+03	4.76E+03	
F14	Mean	4.64E+05	**5.78E+04**‡	**8.20**
	Std	3.51E+05	3.22E+04	
F15	Mean	6.26E+03	**3.63E+03**‡	**2.23**
	Std	6.46E+03	3.09E+03	
F16	Mean	1.01E+04	**4.46E+03**‡	**2.97**
	Std	3.67E+02	1.03E+03	
F17	Mean	7.00E+03	**4.67E+03**‡	**1.78**
	Std	7.39E+02	6.09E+02	
F18	Mean	9.74E+05	**3.16E+05**‡	**3.09**
	Std	4.26E+05	1.62E+05	
F19	Mean	**3.88E+03**	**6.07E+03**≈	0.47
	Std	3.09E+03	5.24E+03	
F20	Mean	7.07E+03	**5.33E+03**‡	**1.52**
	Std	7.01E+02	6.15E+02	
F21	Mean	3.08E+03	**2.59E+03**‡	**2.00**
	Std	2.74E+02	3.74E+01	
F22	Mean	3.46E+04	**2.22E+04**‡	**1.62**
	Std	5.29E+02	2.50E+03	
F23	Mean	**3.05E+03**	3.14E+03†	0.89
	Std	3.71E+01	4.96E+01	
F24	Mean	4.07E+03	**3.61E+03**‡	**1.38**
	Std	2.76E+02	6.61E+01	
F25	Mean	**3.26E+03**	3.32E+03†	0.93
	Std	7.68E+01	5.41E+01	
F26	Mean	1.13E+04	**9.69E+03**‡	**1.23**
	Std	3.48E+03	5.70E+02	
F27	Mean	**3.20E+03**	3.53E+03†	0.60
	Std	1.42E-04	5.57E+01	
F28	Mean	**3.30E+03**	3.43E+03†	0.79
	Std	7.72E-05	3.50E+01	
F29	Mean	8.30E+03	**5.43E+03**‡	**2.13**
	Std	8.11E+02	5.41E+02	
F30	Mean	**1.02E+04**	**9.68E+03**≈	**1.08**
	Std	9.45E+03	3.03E+03	
w/t/l			19/4/7	mean IAR=1.94

In each iteration, all candidate solutions except the last one are updated based on the standard crossover, mutation, and selection operators, while the center-based candidate solution does not employ these operators. In each iteration, the last candidate solution is the average of the N best candidate solutions (Eq. 5). One of the characteristics of the proposed scheme is that it does not need any extra fitness evaluations.

Figure 1 using an example shows the concept of center in a 1-D problem with $N_P = 6$ where 5 candidate solutions (circular points) are generated as usual DE algorithm. We assume N=3. The set of three best candidate solutions is $B = \{\overrightarrow{x_2}, \overrightarrow{x_3}, \overrightarrow{x_4}\}$ which their corresponding positions are $P = \{3, 5, 8\}$ and center-based candidate solution ($\overrightarrow{x_{center}}$) is 5.33.

A. Analyzing behavior of center-based candidate solution

In this paper, we provide a center-based candidate solution based on N best candidate solutions. We can say that center-based candidate solution acts such as a multi-parent crossover, which N best candidate solutions contribute together. As a result, it can be said that the proposed algorithm benefits from the advantages of multi-parent crossover.

Another valuable point is that the center-based candidate solution most likely is better than population's worst member; intuitively, the center point of N best candidate solutions has a higher chance to be better than the worst member in the current population. Figure 3 shows intuitively this concept. To this end, we put forward two scenarios: 1) the global optimum is inside the population's cloud and 2) the global optimum is outside the population's cloud. From the figure, we can observe that in both scenarios, the center-based candidate solution (shown by a yellow star) is closer to global optimum (shown by a red square) than at least one candidate solution (the worst candidate solution). In order to verify it experimentally, we conducted an experiment on CEC-2017 benchmark functions. Employed settings are mentioned in Section V. The percent of center-based candidate solution is better than or similar to the worst candidate solution was 100% in all benchmark functions which clearly demonstrated that center-based candidate solution is similar to or better than the worst candidate solution in the current population; see Figure 3.

The center-based candidate solution is not only better than the worst candidate solution, but also is better than the best candidate solution with a high probability. Figure 4 shows the percent of center-based candidate solution is better than the best candidate solution compared to other candidate solutions. On average, the center-based candidate solution was better in 45% of cases than the best candidate solution; it is indicating the power of center-based scheme and demonstrating that center-based candidate solution has a significant ability to be the best candidate solution. Also, it shows that center-based candidate solution was better than the best candidate solution in at least 27% cases.

In the next experiment, we update center-based candidate solution in two methods, including 1) standard crossover and mutation operators and 2) based on the average of N best candidate solutions. The result can be seen in Figure 2, which clearly shows the effectiveness and more stability of the center-based updating scheme for center-based candidate solution.

V. Numerical results

To assess our proposed scheme, some experimental studies are conducted on CEC-2017 benchmark functions [30] with dimensions 30, 50, and 100 and in comparison to DE algorithm. CEC-2017 includes of unimodal functions ($F1 - F3$), multi-modal functions ($F4 - F10$), hybrid multi-modal functions ($F11 - F20$), and composite functions ($F21 - F30$). As a result, it is a suitable set of benchmark functions to evaluate our scheme efficiently.

In this study, the maximum number of function evaluations is set to $3000 \times D$ in all experiments where D is the problem's dimensions. Furthermore, F, CR, and N are selected 0.5, 0.9, and 3, respectively. Because of the random behavior of the algorithms, each algorithm is run 25 times independently, and statistical results, including mean and standard deviation, are reported. In order to evaluate the results statistically, a two-sided Wilcoxon statistical test with a confidence interval of 95% is carried out between DE_Center_P and DE algorithm. For each function, the best results based on two-sided Wilcoxon statistical test is boldfaced. Also, we provided the improved accuracy rate for each function which shows the relative improvement that yielded by the proposed algorithm and is defined as

$$IAR = \frac{Error\ of\ DE}{Error\ of\ DE\_Center\_P} \tag{6}$$

Error of DE_Center_P (or DE)

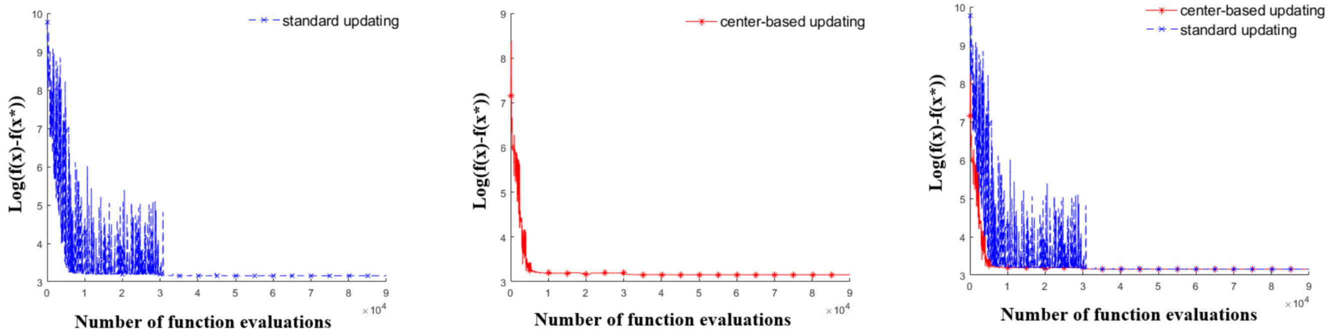
$$= f(x) - f(x^*)$$

Where $f(x)$ is the obtained objective function value and $f(x^*)$ is the optimal value. A value greater than 1 implies that DE_Center_P outperforms DE. In the last column of each table, we calculate the mean IAR. To avoid being biased toward large IAR in the tables (for example 1.82E07 in Table I), large values have been omitted in calculating the mean.

Table I summarizes the results of DE_Center_P against DE algorithm. $w/t/l$ in the last row indicates that DE_Center_P wins in w functions, ties in t functions, and loses in l functions. From the table, DE_Center_P outperforms DE in 18 out of 30 functions, while it lost in only 2 out of 30 functions, which indicates the effectiveness of the proposed center-based scheme. DE_Center_P provided similar performance in 10 functions compared to DE. DE_Center_P yielded better results than DE algorithm for all unimodal functions, and also it performs better in this type of function in range of 1.82 to 1.82E07 times, which indicates the proposed algorithm has higher exploitation power than DE. Also, DE_Center_P obtained better or similar results in all multi-modal functions (F4-F20) except F4, which shows the significant ability of the proposed algorithm in terms of exploration. Also, the proposed algorithms obtained better results in 7 out of 10 composite functions and 2 similar results.

The convergence curves of DE_Center_P compared to DE algorithm in handling some selected functions are illustrated in Figure 5. From the figures, we can conclude that convergence in DE_Center_P is better than DE algorithm.

For further confirmation of efficacy of DE_Center_P, we have fulfilled some experiments in higher dimensions.



(a) Standard updating

(b) Center-based updating

(c) Standard updating VS. center-based updating in one figure.

Fig. 2: Standard updating Vs. center-based updating for center-based candidate solution.

(a) The global optimum is inside the population.

(b) The global optimum is outside the population.

Fig. 3: Center-based candidate solution is most likely better than population's worst member; which is located in edge of the population cloud (a), or back side of the cloud (b) depending on the global solution is surrounded by population or not.

Tables II and III indicate the results for D=50 and 100, respectively. From Table II, DE_Center_P outperformed DE in 18 out of 30 functions, and similar results in 8 functions. Such results are also obtained for D=100 (i.e. better results in 19 functions and similar results in 4 functions).

In this paper, we introduced a parameter, *N*, which determines the number of candidate solutions to calculate x_{center}. we conducted a sensitivity analysis on *N* with different values. Table IV shows the numerical results. It can be seen that different values for N has not a great impact on performance of DE_Center_P. However, N=3, 7, and 25 were slightly better than others.

All in all, the extensive set of experiments clearly verify that center-based candidate solution effectively directs the entire population towards the global optimum.

VI. Conclusion Remarks

This paper proposes a novel center-based differential evolution algorithm named DE_Center_P. In DE_Center_P, a candidate solution in the population is formed based on center-based sampling. The position of center-based candidate solution is updated based on the mean of N best candidate solutions in each iteration. Center is closer to an unknown solution compared to other points. As a result, although there is only one center-based candidate solution in the population, it as a leader directs more effectively the whole population towards the global optimum. We investigated that center-based candidate solution is not only better than the worst candidate solution, but also is better than the best candidate solution with a high probability. To verify the proposed algorithm, we conducted some experiments on CEC-2017 benchmark functions with D=30, 50, and 100. Experimental results clearly indicate that proposed center-based DE outperformed the standard DE algorithm. The concept of center-based population can be extended in other versions of DE algorithm such as SHADE, even in other population-based algorithms such as PSO. Also, a center-based multi-objective variant is under investigation.

References

[1] S. J. Mousavirad, G. Schaefer, and I. Korovin, "A global-best guided human mental search algorithm with random clustering strategy," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 3174–3179, IEEE, 2019.

[2] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing*, vol. 4, no. 2, pp. 65–85, 1994.

[3] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *IEEE international conference on evolutionary computation proceedings*, pp. 69–73, IEEE, 1998.

[4] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *Journal of global optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[5] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Human mental search: a new population-based metaheuristic optimization algorithm," *Applied Intelligence*, vol. 47, no. 3, pp. 850–887, 2017.

[6] X. Wu and A. Che, "A memetic differential evolution algorithm for energy-efficient parallel machine scheduling," *Omega*, vol. 82, pp. 155–165, 2019.

[7] S. J. Mousavirad, H. Ebrahimpour-Komleh, and G. Schaefer, "Effective image clustering based on human mental search," *Applied Soft Computing*, vol. 78, pp. 209–220, 2019.

[8] S. J. Mousavirad, A. A. Bidgoli, H. Ebrahimpour-Komleh, and G. Schaefer, "A memetic imperialist competitive algorithm with chaotic maps for multi-layer neural network training," *International Journal of Bio-Inspired Computation*, In press.

[9] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Human mental search-based multilevel thresholding for image segmentation," *Applied Soft Computing*, 2019.

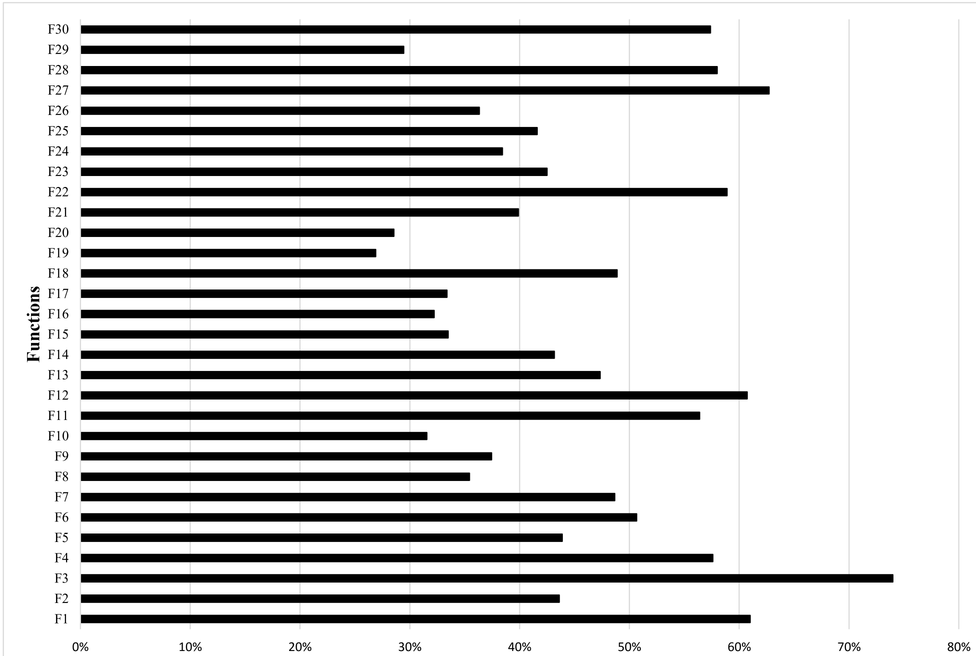
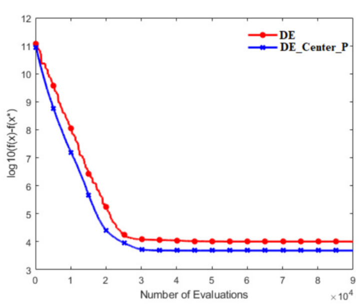
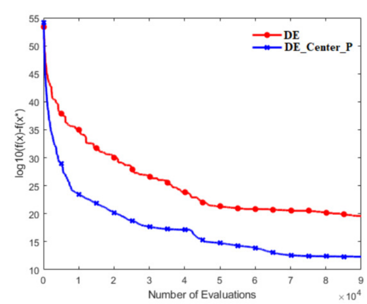


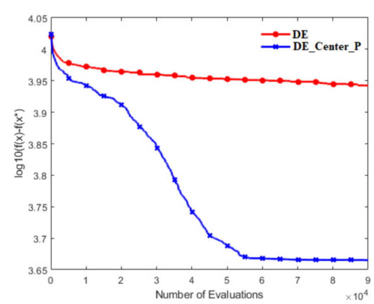
Fig. 4: The percent of that which the center-based candidate solution is better than the current best candidate solution (mean: %47, median: %43, maximum: %74, minimum: %27, and standard deviation: %12).



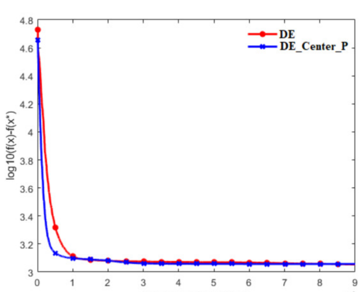
(a) F1



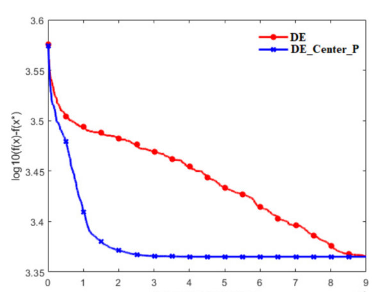
(b) F2



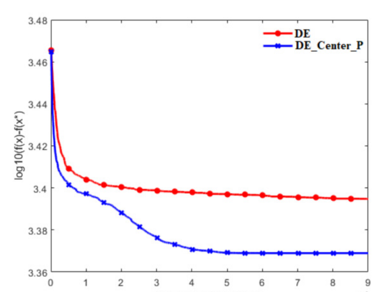
(c) F10



(d) F11



(e) F20



(f) F21

Fig. 5: Convergence curves for some selected benchmark functions

TABLE IV: Center-based DE with various N values.

Functions	DE	3	4	5	6	7	25
F1	8.68E+03	4.82E+03≈	1.22E+04≈	3.23E+03≈	4.83E+03≈	3.25E+03≈	5.09E+03≈
F2	3.86E+19	2.12E+12‡	4.16E+12‡	1.76E+14‡	1.70E+13‡	8.88E+12‡	1.93E+13‡
F3	9.43E+03	1.68E+03‡	2.25E+03‡	1.94E+03‡	2.97E+03‡	2.40E+03‡	3.92E+03‡
F4	4.35E+02	4.83E+02†	4.89E+02†	4.83E+02†	4.86E+02†	4.87E+02†	4.98E+02†
F5	6.85E+02	5.42E+02‡	5.44E+02‡	5.39E+02‡	5.34E+02‡	5.37E+02‡	5.39E+02‡
F6	6.00E+02	6.00E+02≈	6.00E+02≈	6.00E+02≈	6.00E+02≈	6.00E+02≈	6.00E+02≈
F7	9.12E+02	7.82E+02‡	7.80E+02‡	7.78E+02‡	7.75E+02‡	7.80E+02‡	7.69E+02‡
F8	9.89E+02	8.34E+02‡	8.37E+02‡	8.40E+02‡	8.35E+02‡	8.35E+02‡	8.32E+02‡
F9	9.00E+02	9.10E+02†	9.06E+02†	9.15E+02†	9.04E+02†	9.08E+02†	9.10E+02†
F10	8.76E+03	4.63E+03‡	4.55E+03‡	4.82E+03‡	4.64E+03‡	4.49E+03‡	4.25E+03‡
F11	1.13E+03	1.14E+03≈	1.15E+03†	1.14E+03≈	1.15E+03†	1.16E+03≈	1.14E+03≈
F12	2.55E+05	3.86E+04‡	3.89E+04‡	6.04E+04‡	5.18E+04‡	3.64E+04‡	5.39E+04‡
F13	1.82E+04	7.55E+03≈	8.32E+03≈	1.06E+04≈	5.82E+03‡	6.43E+03≈	8.26E+03≈
F14	1.47E+03	1.45E+03‡	1.44E+03‡	1.44E+03‡	1.45E+03‡	1.45E+03‡	1.45E+03‡
F15	1.61E+03	1.56E+03‡	1.54E+03‡	1.55E+03‡	1.79E+03≈	1.56E+03‡	1.55E+03‡
F16	3.09E+03	2.50E+03‡	2.40E+03‡	2.60E+03‡	2.60E+03‡	2.54E+03‡	2.31E+03‡
F17	2.17E+03	1.93E+03‡	1.97E+03‡	1.97E+03‡	1.92E+03‡	1.94E+03‡	1.86E+03‡
F18	1.09E+04	1.25E+04≈	1.74E+04†	1.29E+04≈	1.35E+04≈	1.46E+04≈	1.18E+04≈
F19	1.92E+03	1.95E+03≈	1.92E+03≈	1.93E+03≈	1.98E+03≈	1.93E+03≈	1.99E+03≈
F20	2.32E+03	2.32E+03≈	2.30E+03≈	2.30E+03≈	2.28E+03≈	2.27E+03≈	2.29E+03≈
F21	2.48E+03	2.34E+03‡	2.33E+03‡	2.33E+03‡	2.33E+03‡	2.33E+03‡	2.33E+03‡
F22	9.99E+03	3.00E+03‡	3.81E+03‡	3.07E+03‡	3.58E+03‡	3.81E+03‡	3.33E+03‡
F23	2.83E+03	2.69E+03‡	2.69E+03‡	2.69E+03‡	2.69E+03‡	2.68E+03‡	2.68E+03‡
F24	3.01E+03	2.86E+03‡	2.85E+03‡	2.85E+03‡	2.86E+03‡	2.86E+03‡	2.85E+03‡
F25	2.88E+03	2.88E+03≈	2.88E+03≈	2.88E+03≈	2.88E+03≈	2.88E+03≈	2.88E+03≈
F26	5.26E+03	3.99E+03‡	4.00E+03†	4.00E+03≈	4.04E+03‡	4.04E+03‡	4.02E+03‡
F27	3.20E+03	3.20E+03≈	3.20E+03≈	3.20E+03≈	3.20E+03≈	3.20E+03≈	3.20E+03≈
F28	3.30E+03	3.22E+03‡	3.21E+03‡	3.22E+03‡	3.22E+03≈	3.21E+03‡	3.23E+03‡
F29	3.83E+03	3.46E+03‡	3.43E+03‡	3.44E+03‡	3.41E+03‡	3.42E+03‡	3.42E+03‡
F30	3.22E+03	6.83E+03†	6.92E+03†	7.30E+03†	6.97E+03†	7.08E+03†	7.26E+03†
w/t/l		18/9/3	17/7/6	17/10/3	17/9/4	18/9/3	18/9/3

[10] S. J. Mousavirad, G. Schaefer, and H. Ebrahimpour-Komleh, "A benchmark of population-based metaheuristic algorithms for high-dimensional multi-level image thresholding," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2394–2401, IEEE, 2019.

[11] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[12] Y.-J. Zheng, X.-L. Xu, H.-F. Ling, and S.-Y. Chen, "A hybrid fireworks optimization method with differential evolution operators," *Neurocomputing*, vol. 148, pp. 75–82, 2015.

[13] S.-Y. Park and J.-J. Lee, "Stochastic opposition-based learning using a beta distribution in differential evolution," *IEEE transactions on cybernetics*, vol. 46, no. 10, pp. 2184–2194, 2016.

[14] S. J. Mousavirad and S. Rahnamayan, "Differential evolution algorithm based on a competition scheme," in *14th International Conference on Computer Science and Education*, 2019.

[15] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, and P. N. Suganthan, "Ensemble of differential evolution variants," *Information Sciences*, vol. 423, pp. 172–186, 2018.

[16] T. Robič and B. Filipič, "Differential evolution for multiobjective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 520–533, Springer, 2005.

[17] S. Bandyopadhyay and A. Mukherjee, "An algorithm for many-objective optimization with reduced objective computations: A study in differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 400–413, 2015.

[18] S. Hui and P. N. Suganthan, "Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization," *IEEE transactions on cybernetics*, vol. 46, no. 1, pp. 64–74, 2016.

[19] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Multilevel image thresholding using entropy of histogram and recently developed population-based metaheuristic algorithms," *Evolutionary Intelligence*, vol. 10, no. 1-2, pp. 45–75, 2017.

[20] S. Zhou, L. Xing, X. Zheng, N. Du, L. Wang, and Q. Zhang, "A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times," *IEEE Transactions on Cybernetics*, 2019.

[21] S. Rahnamayan and G. G. Wang, "Center-based sampling for population-based algorithms," in *2009 IEEE Congress on Evolutionary Computation*, pp. 933–938, IEEE, 2009.

[22] H. Hiba, S. Mahdavi, and S. Rahnamayan, "Differential evolution with center-based mutation for large-scale optimization," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, IEEE, 2017.

[23] H. Hiba, M. El-Abd, and S. Rahnamayan, "Improving SHADE with center-based mutation for large-scale optimization," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1533–1540, IEEE, 2019.

[24] H. Hiba, A. Ibrahim, and S. Rahnamayan, "Large-scale optimization using center-based differential evolution with dynamic mutation scheme," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 3189–3196, IEEE, 2019.

[25] G. Liu, C. Xiong, and Z. Guo, "Enhanced differential evolution using random-based sampling and neighborhood mutation," *Soft Computing*, vol. 19, no. 8, pp. 2173–2192, 2015.

[26] S. Mahdavi, S. Rahnamayan, and K. Deb, "Center-based initialization of cooperative co-evolutionary algorithm for large-scale optimization," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 3557–3565, IEEE, 2016.

[27] R. A. Khanum and M. A. Jan, "Centroid-based initialized JADE for global optimization," in *2011 3rd Computer Science and Electronic Engineering Conference (CEEC)*, pp. 115–120, IEEE, 2011.

[28] S. J. Mousavirad, A. Asilian bidgoli, and S. Rahnamayan, "Tackling deceptive optimization problems using opposition-based DE with center-based latin hypercube initialization," in *14th International Conference on Computer Science and Education*, IEEE, 2019.

[29] S. Rahnamayan and G. G. Wang, "Toward effective initialization for large-scale search spaces," *Trans Syst*, vol. 8, no. 3, pp. 355–367, 2009.

[30] G. Wu, R. Mallipeddi, and P. Suganthan, "Problem definitions and evaluation criteria for the cec 2017 competition on constrained real-parameter optimization," *National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report*, 2017.