Novel Differential Evolutionary Optimization Approach for an Integrated Motor-Magnetic Gear used for Propulsion Systems

CLAUDIA V. POP¹, M. ESSAID², L. IDOUUGHAR², AND D. FODOREAN¹, (Senior Member, IEEE)

¹Technical University of Cluj-Napoca, Electrical Machines and Drives Department, Cluj, 400114, Romania
²University of Haute-Alsace, IRIMAS, Mulhouse, France

Corresponding author: D. Fodorean (e-mail: daniel.fodorean@emd.utcluj.ro).

This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI - UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0776 / No. 36 PCCDI / 15.03.2018, within PNCDI III.

ABSTRACT This paper addresses the proposition of an electrical multiport propulsion system. This traction system contains an electric motor to which a magnetic gear is attached; it is called an integrated motor-magnetic gear (IMMG) propulsion unit. This IMMG has two shafts and two levels of torque and speed. Thus, it is suited for the propulsion of an electric tractor, which has different diameters for the front and rear wheels. The IMMG design is optimized based on the differential evolution (DE) algorithm. For such an algorithm, offline tuning of its parameters may fail given the different characteristics of optimization problems. Therefore, a parameter adaptation that considers feedback from the search process can be an alternative. In this work, we propose a novel online strategy to control the DE algorithm’s parameters. This strategy is based on a simple adaptation mechanism to generate a promising mutation parameter value for each individual in the population. The design process was evaluated through experiments, and a good correlation was found between the designed performances and tested results.

INDEX TERMS differential evolutionary optimization, integrated motor-magnetic gear, EV propulsion.

I. INTRODUCTION Multiport and variable transmission propulsion systems can benefit electric vehicles (EVs). One can obtain such a multiport system by connecting the electrical machine to a mechanical gear. This is useful since for certain operating conditions, such as when an extra torque or speed is needed. Today, for a standard EV, the electric motor is connected to a mechanical cascaded gear system, which affects the overall efficiency of the transmission. In addition, the mechanical gear involves several well-known drawbacks: physical contact (friction) between the teeth, need for lubrication, local heat, material fatigue, and consequent mechanical losses [1]. In the case of magnetic gears (MGs), these drawbacks are avoided [2], [3], [4].

In this context, our research proposes a propulsion unit that eliminates mechanical transmission. Instead, it utilizes a magnetic gear. Our second objective is to use this propulsion for an electric tractor, which has two levels of speed and torque, for the front and rear wheels. Thus, a multiport electromagnetic system is suited for our application.

Some of the existing multiport magnetic transmission solutions found in the literature are shown in Fig. 1.
Fig. 1a shows the flux MG structure, which is not very efficient, since not all permanent magnets (PMs) are involved in the transmission process. Instead, it offers access to four shafts of different diameters, [5]. Fig. 1b shows another multiport MG with at least six shafts. It is true that the efficiency is reduced in this case, but for some applications, it could be the only option [6]. Fig. 1c shows a toroidal MG, [7], again with at least three available shafts. It offers a variable transmission but with a very complex and expensive configuration. Fig. 1d shows a double rotor electrical machine, mainly excited through PMs. Sometimes, it is called a “counterrotating electrical machine” [8]. It is suited for generator power applications and has an acceptable power density. Sometimes only one shaft is available, and for other configurations, the field winding placed on the second rotor is fed via a brush-ring circuit. Nevertheless, such a variant looks appropriated for our application. Thus, we will have a closer look at the dual rotor variants. Different types of structures with dual rotor (DR) topologies have been found in the literature. We have identified their main characteristics and performances to better situate our work. In Table 1, the evaluated DR configurations are presented, and each structure has a magnetic gear as well as:

- A (first variant) permanent magnet synchronous machine (DR-PMSM1), [9], which has a magnetic coupling between the motor and the integrated MG, a good power density and efficiency (for the given power), but the power factor is poor.
- A synchronous reluctance machine (DR-SynRM), [10], which uses both rotors to strengthen the torque, and thus, there is only one shaft (one level of speed and torque).
- A (second variant) PMSM (DR-PMSM2), [11], having mechanical coupling and a high gear ratio, but for the given power, the efficiency is poor.
- A permanent magnet synchronous generator (DR-PMSG), [8], which uses cheap materials, and consequently, the power density is affected.
- A hybrid excited synchronous machine (DR-HESM), [12], which again uses both rotors to strengthen the torque, and its major drawback is the presence of brushes to feed the field winding.
- A flux switching synchronous machine (DR-FSSM), [13] which has a very reduced level of torque ripples, but the power factor is also quite poor.

When dealing with dual rotor topologies, there is usually a magnetic connection between the rotating armatures and, rarely, a mechanical connection. In the first case, the flux on both rotors is controlled and affected by the stator’s flux. Thus, a very good power density is obtained since both rotors share the same yoke [8]. On the other hand, when a mechanical connection is used, a supplementary yoke (of the second rotor) exists. Instead, there is no magnetic coupling between the electric motor and magnetic gear. Thus, it follows that in the case of a short circuit on the motor’s winding, only the motor’s PMs might be affected, while the magnets within the MG are protected. Thus, for our propulsion system, we seek to have the best possible power density while avoiding magnetic coupling between the motor and the MG. For the given application, we propose an integrated motor-magnetic gear (IMMG) configuration with two available shafts and two different torque and speed levels. The motor is actually a PMSM with an outer rotor, on top of which we add an MG capable of increasing the speed based on a specific magnetic pole configuration.

The analysis employed here concerns the optimization of this IMMG based on the differential evolution (DE) algorithm, which is a powerful population-based approach. For this DE algorithm, offline tuning of its parameters may fail given the different characteristics of optimization problems. Therefore, a parameter adaptation that considers feedback from the search process can be an alternative. Here, we propose a novel online strategy to control the DE parameters. This strategy is based on a simple adaptation mechanism to generate a promising mutation parameter value for each individual in the population. In addition, a powerful dimension-based mechanism is proposed to adapt the crossover probability for different characteristics of optimization problems. Therefore, this adaptive differential evolution is applied to improve the design of the IMMG. The algorithm performance tends to be very competitive when compared with powerful, recent state-of-the-art algorithms. To recap, the novelties of this work are:

- Proposition of a multipropulsion unit having two shafts with two levels of torque and speed;
The propulsion unit is optimized based on a novel differential evolution (DE) optimization algorithm. The optimization algorithm uses an online strategy to control the parameters of the DE algorithm based on a simple adaptation mechanism to generate a promising mutation parameter value for each individual in the population. The quality of the proposed algorithm is verified by comparing its performances with those obtained by other algorithms and approaches. This new approach was applied to the proposed IMMG.

The paper is organized as follows: in section II, the configuration of the studied IMMG is presented; next, in section III, the optimization algorithm is introduced, as well as the obtained results; in section IV, the experimental validation of the studied IMMG is presented to demonstrate the operation of the optimized multipropulsion system; in the appendix, a nomenclature and abbreviations section aids in the readability of the manuscript.

II. THE INTEGRATED MOTOR-GEAR PROPULSION SYSTEM WITH TWO SHAFTS

Fig. 2 shows the main active parts of the studied IMMG, as well as the assembled structure. The stator is the interior part, designed with 39 slots or teeth. A three-phase copper winding is considered here. Then, the PMSM’s air-gap follows. The connection between the PMSM and MG is made through a doubled rotor with the same number of poles and a surface that magnetically disconnects the two rotors. A second layer of air-gap follows. Next, we have the static part with ferromagnetic teeth, alternating with nonconducting material. A third layer of air-gap follows, as well as the high-speed rotor (with its yoke and PMs).

The IMMG’s design is based on a magnetic equivalent circuit (MEC) [14]. If for the motor side the classic design approach was used (the torque being proportional to the number of pairs of poles, and the product of flux and currents on the orthogonal axis), for the MG’s sizing some information is briefly introduced. The MG’s MEC is divided in multiple elements, based on the number of the stationary pole pieces. For each piece, belonging to a certain material (air, steel, PM), a specific magnetic reluctance is calculated, and next the magnetic flux is obtained. Thus, by multiplying the magnetomotive force of the PMs (dependend on the rotor position angle θ) of the outer rotor with the modulated flux of the inner part of the MG (the inner rotor and the static teeth part), one can obtain the output torque of the MG. To summarize the above, the generic magnetic reluctance of each portion of the magnetic circuit is calculated based on (1), the magnetic flux based on (2) and the output torque based on (3).

\[
R_{m,x} = \frac{l_{m,x}}{(\mu_x \cdot S_x)} \quad (1)
\]

\[
\Phi_{m,in} = \frac{F_{m,in}}{R_{m,eq}} \quad (2)
\]

\[
T_{out} = \sum_{j=1}^{N_{p}} \Phi_{m,in,j} \times \Delta F_{m,\text{out},j} \quad (3)
\]

The IMMG’s main input data, as well as the analytically obtained performances, are depicted in Table 2. At the level of the low-speed rotor, a power of 1580 W was obtained for an efficiency of 0.91. Instead, when the higher rotor shaft is used, it involves a 6% efficiency drop, and the output power is 1500 W. This efficiency drop is mainly due to the mechanical loss component on the MG side produced by the complex bearing arrangement.

| Parameter                                      | Value | Units |
|-----------------------------------------------|-------|-------|
| Supplying dc battery voltage                  | 48 V  |       |
| The number of pairs of poles for the low-speed rotor | 17    |       |
| The number of teeth of the static part        | 22    |       |
| The number of pairs of poles for the high-speed rotor | 5     |       |
| Supplying frequency                           | 119 Hz|       |
| Low speed rotor                               | 420 r/min |       |
| High speed rotor                              | 1428 r/min |       |
| Average power of the double-layer rotor       | 1580 W |       |
| Average torque of the double-layer rotor      | 36.8 Nm |       |
| Average power of the high-speed rotor         | 1500 W |       |
| Average torque of the high-speed rotor        | 10.1 Nm |       |

FIGURE 2. The main active parts of the studied IMMG: a) inner stator of the PMSM; b) double sided low speed rotor; c) MG’s static part; d) MG’s outer rotor; e) 3D view of the assembled structure.
With respect to the bearing variant for the outer rotor of the MG, this is currently the best solution. In the future, an investigation on the outer rotor bearing needs to be done to reduce the mechanical losses as much as possible and consequently to increase the efficiency of the propulsion system. Nevertheless, it is an acceptable efficiency for such a power level while offering two levels of torque and speed. It is worth mentioning that the most significant advantage of this structure is the speed increase capability, without increasing the supplying frequency. It is known that the iron losses are proportional to the square of the frequency. Thus, by keeping the supplying frequency at a reduced level, the efficiency of the system is not affected. The second important advantage of the IMMG is with respect to the switching stress on its inverter. Since the fed frequency is 3.4 times lower even when the converter's efficiency is unaffected.

**III. OPTIMIZATION DESIGN OF THE STUDIED PROPULSION SYSTEM**

This section is devoted to explaining the optimization algorithm, particularly our parameter adaptation strategy. Subsection A briefly defines the differential evolution (DE) algorithm. Subsection B explains how the scale factor, \( F \), is adapted. Accordingly, the proposed mutation framework is investigated in subsection C. A novel dimension-based adaptation strategy for the crossover parameter, \( CR \), is explained in subsection D. The combination of the algorithmic components is introduced in subsection E. A comparative study is performed within subsection F, while the convergence and load computation of the proposed algorithm are depicted in sub-sections G and H, respectively. The optimization results of the considered IMMG are given in subsection I.

**A. DIFFERENTIAL EVOLUTION**

DE is a population-based stochastic search algorithm that relies on three search operators: mutation, crossover, and selection that are applied to the population. Supposing that the solution vector contains \( D \) variables, \( x_i^G = [x_{i1}^G, x_{i2}^G, ..., x_{iD}^G] \) represents solution \( i \) at generation \( G \), the DE procedure can be represented in five phases:

1) **Initialization**: The mutation parameter \( F \), the crossover parameter \( CR \), the population size \( N_P \), and the number of generations \( G_{\text{max}} \) are set. The starting population of individuals is randomly initialized.

2) **Mutation**: For each individual \( x_i^G \) in the parent population, a mutant vector \( v_i^{G+1} \) is calculated as follows:

\[
v_i^{G+1} = x_i^G + F \cdot (x_{j1}^G - x_{j2}^G) \tag{4}
\]

3) **Crossover**: Binomial crossover is the basic search operator of DE, where for each individual \( x_i^G \) and for each dimension \( j \), a trial vector \( u_i^{G+1} \) is generated as follows:

\[
u_i^{G+1} = \begin{cases} v_i^{G+1} & \text{if } j = \mathcal{I}_j \text{ or } R_j < CR \\ x_i^G & \text{otherwise} \end{cases} \tag{5}
\]

It should be mentioned that other operators, such as the exponential crossover, may also be applied.

4) **Evaluation**: For each individual \( x_i^G \), the trial vector is evaluated, and it replaces the parent individual if it has a better fitness.

5) \( G \) is incremented, phases 2 to 5 are repeated, and \( G \) is less than \( G_{\text{max}} \).

Regarding the mutation phase, in addition to DE/rand/1 mentioned above, numerous mutation strategies have been proposed in the scientific literature, such as:

- DE/rand/2 [15]:
  \[
v_i^{G+1} = x_i^G + F \cdot (x_{j1}^G - x_{j2}^G) + F \cdot (x_{j3}^G - x_{j4}^G) \tag{6}
\]

- DE/best/2 [16]:
  \[
v_i^{G+1} = x_i^{\text{best}} + F \cdot (x_{j1}^G - x_{j2}^G) + F \cdot (x_{j3}^G - x_{j4}^G) \tag{7}
\]

- DE/current-to-best/1 [16]:
  \[
v_i^{G+1} = x_i^G + F \cdot (x_{j1}^G - x_{j2}^G) + F \cdot (x_{j3}^G - x_{j4}^G) \tag{8}
\]

- DE/current-to-best/2 [15]:
  \[
v_i^{G+1} = x_i^{\text{best}} + F \cdot (x_{j1}^G - x_{j2}^G) + F \cdot (x_{j3}^G - x_{j4}^G) \tag{9}
\]

(All the meaning of the introduced variables is given in the nomenclature section.)

**B. LINEAR BIAS REDUCTION FOR “F” PARAMETER ADAPTATION**

In this subsection, our approach to adapt the mutation scale factor is explained in detail. It strongly depends on the proposition used in LSHADE [17], which computes a location parameter using the Lehmer mean of the successful proposition used in LSHADE [17], which computes a location parameter using the Lehmer mean of the successful parameter denoted as \( S_F \) at each generation as follows:

\[
\text{mean}_L(S_F) = \frac{\sum_{f \in F} F^m \cdot F^m}{\sum_{f \in F} F^m} \tag{6}
\]

However, a structural bias problem in parameter adaptation was raised in [18], which states that when \( m \) is small (\( m = 2 \) in LSHADE [17]), small values of \( F \) will be generated, resulting in local search behavior. Following the suggestions of [18], a large value of the \( m \) parameter is set and linearly reduced to calibrate the exploration and exploitation capabilities of our proposal as follows:

\[
m = m_{\text{min}} + (m_{\text{init}} - m_{\text{min}}) \frac{\text{MaxIT} - IT}{\text{MaxIT}} \tag{7}
\]

Afterward, \( \text{mean}_L(S_F) \) will be stored as a location parameter in an archive called \( \text{memory}_{SF} \). It should be noted that when \( \text{memory}_{SF} \) is full, its oldest entry is removed. Finally, a location parameter \( \mu_F \) is randomly selected, and the \( F \) parameter value for each individual is generated using a Cauchy distribution as follows:

\[
\mu_F = \text{mean}_L(S_F) + \text{std}_L(S_F) \cdot \text{rand} \tag{8}
\]
\[ F_i = \text{rand}c(\mu_F, 0.1) \]  

Our parameter adaptation strategy for the mutation scale factor is depicted in Algorithm 1:

**Algorithm 1** Mutation parameter adaptation strategy

1. Input: Successful \( F \) parameters set denoted as \( S_F \)
2. Output: New \( \alpha \) for each individual
3. if \( S_F \) is not empty then
4. Compute a Lehmer mean value of \( S_F \) using (6)
5. Insert it into \( \text{memory}_\text{sf} \)
6. end if
7. Randomly select a value from \( \text{memory}_\text{sf} \) as a location parameter \( \mu_F \)
8. Generate \( F \) for each individual using (8)

C. **THE MUTATION FRAMEWORK**

As mentioned in [19], proposing suitable mutation strategies is an important task to improve the performance of the DE algorithm. Accordingly, a framework that switches between two mutation strategies is introduced to potentially achieve a balance between exploration and exploitation. The switching mechanism is based on a linearly reduced control parameter to favor the exploratory mutation strategy and gradually favors the exploitative strategy at the late stages of DADE. The exploratory mutation strategy is called DE/massed-current-to-the exploitative strategy at the late stages of DADE. The switching between two mutation strategies is introduced to potentially achieve a balance between exploration and exploitation. The switching parameter = 0.75. Computing the mass and updating CR parameters are detailed in Algorithm 3:

**Algorithm 2** The proposed mutation framework

1: Input: The parent population, \( \text{Switch}_\text{prob} \)
2: Output: The mutant population
3: if \( \text{Switch}_\text{prob} > \text{rand} \) then
4: Apply the mutation using (9)
5: else
6: Apply the mutation using (10)
7: end if

D. **DIMENSION-BASED ADAPTIVE STRATEGY TO ADAPT A "CR" PARAMETER**

In this subsection, we address the issue of how to efficiently adapt the CR parameter based on a novel yet simple learning technique. After crossing the offspring population with the parents, a binary-valued matrix called \( \text{Matrix}_{CR} \) is used to record whether a given dimension is taken from the parent or from the offspring for a given individual. Then, for each dimension, mass for the parent and offspring are computed based on how much the given individual is improved. If the parent mass is larger than the offspring mass, then the \( CR \) value for the given dimension is generated using a normal distribution with a location parameter = 0.25. This would produce a small value of \( CR \) forcing to share the value of the parent for the given dimension during the crossover procedure. Otherwise, the \( CR \) value is generated using a location parameter = 0.75. Computing the mass and updating \( CR \) parameters are detailed in Algorithm 3:

**Algorithm 3** Crossover parameter adaptation strategy

1: Input: \( \text{Matrix}_{CR} \), \( \text{NormReward} \), \( \text{dim} \), \( \text{PopSize} \), \( \text{parent}_\text{mass} \), \( \text{offspring}_\text{mass} \)
2: Output: New \( CR \) for each dimension
3: while \( i < \text{dim} \) do
4: while \( j < \text{PopSize} \) do
5: if \( \text{Matrix}_{CR}(i,j) = 0 \) then
6: \( \text{parent}_\text{mass}(i) + = \text{NormReward}(j) \)
7: else
8: \( \text{offspring}_\text{weight}(i) + = \text{NormReward}(j) \)
9: end if
10: end while
11: if \( \text{parent}_\text{mass}(i) < \text{offspring}_\text{weight}(i) \) then
12: \( CR(i) = \text{normrnd}(0.75, 0.1) \) // normal distribution
13: else
14: if \( \text{parent}_\text{mass}(i) < \text{offspring}_\text{weight}(i) \) then
15: \( CR(i) = \text{normrnd}(0.5, 0.1) \) // normal distribution
16: Else
17: \( CR(i) = \text{normrnd}(0.25, 0.1) \) // normal distribution
18: end if
19: end if
20: \( CR(i) = \text{min}(CR(i), 1) \)
21: \( CR(i) = \text{max}(CR(i), 0) \)
22 end while

After the evaluation phase, a reward/penalize value is updated as follows:

\[ \text{reward}(i) = f(i) - f^*(i) \]  

Due to the large difference between the values generated by (11), reward entries will be normalized to a common scale within the range [-1, 1] as follows:

\[ \text{NormReward}(i) = 2 \cdot \frac{\text{reward}(i) - \text{min}(\text{reward})}{\text{max}(\text{reward}) - \text{min}(\text{reward})} - 1 \]  

This proposition is considered a dimension-based parameter adaptation strategy. To the best of our knowledge, this is the first attempt to control the \( CR \) parameter for each dimension rather than for each individual. Indeed, having a unified \( CR \) value for all the dimensions might not always be successful.
because of the weak correlation that might potentially occur between dimensions. Moreover, although parameter adaptation strategies do not impose a serious computational burden [22], [15], other strategies may need an important computational time [23], as noted in [24], where DE parameters evolve using the optimization process of harmony search (HS). Therefore, adapting only the parameters for each dimension can imply a considerable reduction in the algorithm’s learning process.

E. COMBINATION OF THE ALGORITHMIC COMPONENTS

This subsection focuses on how the algorithmic components used are combined in our proposal. The proposition is depicted in Algorithm 4:

Algorithm 4 Dimension-based adaptive differential evolution (DADE)

1: Input: Population pop of PopSize, dim, MatrixCR, reward
2: Output: pop
3: Set CR = 0.5 for all dimensions
4: Set memory_sf of memory_size entries to 0.5
5: while Budget is not consumed do
6: popold ← pop
7: Apply Algorithm 1 to generate F value for each individual
8: Apply Algorithm 3 to generate mutant pop
9: for each individuals i in popold do
10: while j < dim do
11: if rand < CR(j) then
12: u\textsuperscript{G+1} \leftarrow {\text{popold}}_j
13: Matrix CR (i, j) ← 1
14: else
15: u\textsuperscript{G+1} \leftarrow {\text{mutant_pop}}_j // crossover procedure
16: Matrix CR (i, j) ← 0
17: end if
18: j ← j+1
19: end while
20: if u\textsuperscript{G+1} is better than {\text{popold}}_j then
21: {\text{pop}}_j \leftarrow u\textsuperscript{G+1}
22: end if
23: Update the corresponding entry in reward according to (11)
24: Normalize reward according to (12)
25: Apply Algorithm 2 to generate CR value for each dimension
26: Apply linear reduction of the population size to eliminate a fraction of the worst individuals using (13)
27: Apply linear reduction of m parameter according to (7)
28: Apply linear reduction of Switch\_prob parameter according to (14)
29: end for
30: end while

Initially, DADE starts with setting CR for all dimensions and memory_sf to 0.5. Afterward, the mutation parameter adaptation strategy takes place to generate the F value for each individual. In the third phase, the proposed mutation framework is applied to perform the switching mechanism to generate a mutant population with an initial value of Switch\_prob = 0.9.

Then, a binomial crossover is applied to generate trial vectors. During this phase, MatrixCR is used to record whether a given dimension in the trial to be generated is taken from the parent or the offspring. This procedure is performed as an initial step to adapt the CR parameter for the next generation. After applying one generation of the DE procedure, the proposed dimension-based adaptation strategy is performed to generate CR for each dimension for the next generation. Finally, a linear reduction of the population size, the m parameter to compute the Lehmer mean and Switch\_prob, takes place. The new population size is computed as follows:

\[ NP_{G+1} = \text{round}\left(\frac{N_{P_{\text{min}}}-N_{P_{\text{init}}}}{\text{MaxIT}} \cdot \text{IT} + N_{P_{\text{init}}}\right) \]  

(13)

The new value of Switch\_prob is computed as follows:

\[ \text{Switch}_{\text{prob}} = 0.1 + (0.9 \cdot \frac{\text{MaxIT} - \text{IT}}{\text{MaxIT}}) \]  

(14)

F. OPTIMIZATION RESULTS FOR THE CEC 2020 TEST SUITE

Our algorithm proposal was tested using the CEC 2020 test suite. This benchmark contains 10 unconstrained problems, for which, according to the benchmark rules, all the algorithms should be run 30 times for 50,000; 1,000,000; 3,000,000; and 10,000,000 fitness evaluations for problems with 5D, 10D, 15D, and 20D, respectively. For the statistical comparison, the nonparametric Kruskal–Wallis test was conducted (p – value = 0.05), which is a generalized version of the Mann–Whitney Wilcoxon test (specified for two unpaired samples). The Kruskal–Wallis test was followed by a Conover post hoc test for pairwise multiple comparisons.

1- Comparison of the 5D problems: The results obtained by all the algorithms are given in Table 3. This table shows the best and mean values of 30 runs of each algorithm for each problem. The best fitness found for each function is in bold. It can be stated from the results that our proposal performs well on the unimodal function F01. Similarly, optimal solutions were obtained for F04, F05, F06, F07, and F08. For F02, F03, and F09, solutions very close to the optimal solutions were achieved. However, the performance degrades when solving F10.

2- Comparison of the 10D problems: The computational results are summarized again in Table 3, where optimal solutions were obtained for F01 and F04. For F02, F03, F05, F06, and F07, very close solutions to the optimal solutions could be achieved. DADE performance slightly degrades when solving F08, F09, and F10. However, the results are still comparable, considering the performance of rival algorithms.
3- Comparison of the 15D problems: DADE performance for 15D problems can also be seen in Table 3, where optimal solutions could be obtained for F01 and F04. Moreover, promising solutions were achieved considering F02, F03, F05, F06, F07, and F08. The DADE performance degrades when F09 and F10 are solved.

4- Comparison of the 20D problems: The results and comparison with rival algorithms for 20D problems can also be seen in Table 3. DADE obtains the optimal solutions when solving F01 and F04. Similarly, promising results can be noticed when solving F02, F03, F05, F06, and F07 compared to the state-of-the-art algorithms. DADE performance reasonably degrades on F08, F09, and F10. Nevertheless, DADE results remain comparable with rival algorithms. Moreover, the aggregate results for 5, 10, 15, and 20 dimensions using the statistical test in terms of mean results are shown in Tables 4. DADE clearly outperforms the rival algorithms on 5D and 10D problems. For instance, considering 5D problems, it has a better overall performance compared to EBOwithCMAR on five problems, as well as against AGSK on three problems. Besides, it could be outperformed on only one problem by j2020. Similarly, for 15D problems, DADE could exhibit slightly better performance compared to j2020 and AGSK (DADE is better on four problems).

G. COMPARISON IN TERMS OF CONVERGENCE RATE
To investigate the effect of the adopted components on the convergence rate of our proposed algorithm, a comparative study was conducted with AGSK and j2020; these are two of the best algorithms on the CEC 2020 unconstrained-optimization-benchmark competition. Four representative functions were chosen for comparison: F01, F02, F05, and F08, which represent unimodal, basic, hybrid, and composition functions, respectively.
1- **Convergence rate for 5D**: The convergence rate of our proposal is presented in Fig. 3 (where the X axis represents the number of iterations and the Y axis represents the fitness value). It can be stated that DADE converges quickly to the best solution compared to AGSK in all the problems. However, a slightly similar convergence rate can be seen in F05 compared to j2020.

2- **Convergence rate for 10D**: The convergence rate of DADE is shown in Fig. 4. It can be clearly noticed that DADE exhibits better convergence on all the representative problems with a similar performance to j2020 on F05 and F08.

3- **Convergence rate for 15D**: The comparison in terms of convergence rate for 15D problems is presented in Fig. 5, where a quicker convergence is stated on F02 and F05. However, a slightly better convergence of j2020 is noticed on F01 and F08 in the early iterations.

4- **Convergence rate for 20D**: The convergence rate of DADE is presented in Fig. 6 for 20D problems. Indeed, DADE exhibits a better convergence rate on F02, F05, and F08 with a similar performance on F01 compared to j2020.

**H. ALGORITHM COMPLEXITY**

The experiments were executed using MATLAB R2020a software installed on a PC with core i5–7440 (2.80 GHz) CPU and 8 GB RAM on Windows 10 operating system. To obtain a global overview of DADE complexity, the time values $T_0$, $T_1$, $T_2$, and $T_2 - T_1 / T_0$ are computed for the CEC 2020 test suite; their specific values are given in Table 5.

**TABLE 5** The computational load time for the proposed algorithm.

| Dimension | $T_0$  | $T_1$  | $T_2$  | $T_2 - T_1 / T_0$ |
|-----------|--------|--------|--------|-------------------|
| D=5       | 4.46E-02 | 1.84E-04 | 3.36E+00 | 8.39E+01          |
| D=10      | 3.21E-03 | 4.32E+00 | 9.38E+01 |                   |
| D=15      | 5.95E-03 | 4.74E+00 | 1.02E+02 |                   |
| D=20      | 7.51E-03 | 5.45E+00 | 1.18E+02 |                   |

It can be seen from Table 5 that the computational time increases each time the dimension $D$ increases; however, it remains reasonable. For instance, $T_2$ is only 5.45 s when running the proposal on f1 with $D=20$.

$T_0$ is the time needed to execute the following test problem:

**Test problem (for evaluating the algorithm’s load)**

1: $x = 0.55$
2: for $i = 1:1000000$
3: $x = x + x; x = x / 2; x = x * x; x = sqrt(x); x = log(x); x = exp(x); x = x / (x + 2);$
4: end

**I. OPTIMIZATION RESULTS ON THE STUDIED IMMG**

Employing the optimization approach on the problem at hand is justified by the need to increase the autonomy of the EV. Increasing the power density of the propulsion unit (i.e., the ratio of the mechanical power versus the mass of the machine) will have a direct and positive impact on the autonomy of the vehicle (which becomes lighter). Finding the optimal mechanical structure of the electromagnetic propulsion unit is an optimization problem, where the objective function consists of two steps:

- The first step is to reduce the active mass ($m_{atot}$) of the IMMG, which is defined as a function of the mass of each active part:

$$m_{atot} = m_{copper} + m_{steel} + m_{PM}$$  \(15\)
The second step concerns the output power ($P_{\text{out}}$), defined as a function of the input power and losses:

$$P_{\text{out}} = P_{\text{in}} - \sum \text{losses}$$

(16)

The optimization goal is to increase the power density of the IMMGG (decreasing its active mass, while maintaining the output power at the desired level by decreasing the losses).

The objective function is implemented by aggregating the two parts, and is depicted as follows:

$$\min J(x) = -\frac{P_{\text{out}}}{m_{\text{tot}}} + \text{penalty}$$

(17)

where

$$\text{penalty} = 10^4 \cdot \sum_{i=1}^{7} C_i$$

(18)

FIGURE 4 Comparison of the convergence of DADE with AGSK and j2020 on 10D problems.

FIGURE 5 Comparison of the convergence of DADE with AGSK and j2020 on 15D problems.
In (18), \( C_i = 0 \) if constraint \( i \) is satisfied, and 1 otherwise. The algorithm is compared with five state-of-the-art DE/non DE variants, which showed promising performance when solving recent optimization problems:
- Differential Evolution Algorithm for Single Objective Bound-Constrained Optimization Algorithm j2020 [25].
- Evaluating the Performance of Adaptive Gaining Sharing Knowledge Based Algorithm on CEC 2020 Benchmark Problems (AGSK) [26].
- Effective Butterfly Optimizer with Covariance Matrix Adapted Retreat Phase (EBO with CMAR) [27].
- LSHADE with Semi-Parameter Adaptation and Covariance Matrix Adaptation LSHADE-SPACMA [28].
- LSHADE with an Ensemble Sinusoidal Parameter Adaptation (LSHADE-cnEpSin) [29].

It should be mentioned that the proposed parameters were set using a trial-and-error procedure. The control parameters are set as follows:
- \( PopSize = 100; \)
- \( m_{init} = 15; \)
- \( Switch\_prob = 0.9; \)
- \( size\_of\_memory\_sf = 5. \)

To conduct a fair comparison, the parameters of the comparative algorithms were obtained from their relevant articles. The proposal was run 30 times with 1,000,000 fitness evaluations. The best mean and worst solutions of each algorithm were collected. Table 6 shows that the rival algorithms can obtain the best solution known thus far. However, our proposal obtains the best solution in terms of the mean and worst values. This advantage can be justified by the proposed dimension-based adaptation strategy, as well as the mutation framework, which tends to achieve a higher balance between exploration and exploitation.

| Algorithm      | Best        | Mean        | Worst        |
|----------------|-------------|-------------|--------------|
| AGSK           | -4.5546E+02 | -4.1550E+02 | -4.0980E+02 |
| j2020          | -4.5546E+02 | -3.8633E+02 | -2.7691E+02 |
| EBO with CMAR  | -4.5546E+02 | -3.7920E+02 | -2.7691E+02 |
| LSHADE-SPACMA  | -4.5546E+02 | -4.3682E+02 | -4.1580E+02 |
| LSHADE-cnEpSin | -4.5546E+02 | -4.1736E+02 | -4.0980E+02 |
| Our proposition| -4.5546E+02 | -4.5546E+02 | -4.5546E+02 |

Thus, we have verified and validated the quality of the proposed differential evolution approach. Next, the algorithm will be employed on the design process of the studied IMMG by taking into consideration the variation limits for the constraints and mass parameters, as depicted in Table 7 and Table 8.

| Parameter                      | Symbol | Unity | Variation limits |
|--------------------------------|--------|-------|------------------|
| Output power                   | \( P_{out} \) | \( W \) | [1490; 1510] |
| Current consumption            | \( I_S \) | \( A \) | [22; 30] |
| Motor torque (low speed)       | \( T_m \) | \( Nm \) | [33.8; 34.4] |
| Motor’s efficiency             | \( \eta \) | -     | [0.84; 0.99] |
| Motor’s power factor           | \( PF \) | -     | [0.9; 0.99] |
| Rotor inner diameter           | \( Dir \) | \( mm \) | [200; 300] |
| Slot filling factor            | \( \tau \) | \%    | [20; 50] |

**TABLE 6.** Comparison with State-of-the-Art algorithms.

**TABLE 7.** The problem constraints.
Based on a previous study, [30], we have found that these eight optimization parameters from Table 8 are the ones to affect the objective function, and, thus, they are used to establish the boundaries of the space of solutions. We recall that our objective is to improve the power density of the propulsion unit, since this will have a direct impact on the autonomy of the EV. This means that we are trying to decrease the iron losses of the IMMG’s active mass, while maintaining the output power at the desired level.

The obtained optimized results, in comparison with the initial solution, are shown in Table 9.

### TABLE 9. IMMG Results Comparison: Before and After Optimization.

| Parameter | Symbol | Original IMMG | Optimized IMMG | Units | Gain (%) |
|-----------|--------|---------------|----------------|-------|----------|
| m_{tot}   | 8.89   | 6.05          | kg             | +32   |
| P_{out} / m_{tot} | 168 | 248          | W/kg           | +32   |
| Efficiency | 0.85   | 0.92          | ---            | -1.1  |
| Power factor | 0.83   | 0.92          | ---            | 0     |
| Dir       | 207    | 211           | mm             |       |
| hjr       | 8      | 5             | mm             |       |
| histm     | 2      | 1             | mm             |       |
| hjs       | 6.5    | 5             | mm             |       |
| wt        | 9      | 7.6           | mm             |       |
| gap       | 1      | 0.85          | mm             |       |
| hmp       | 3      | 4.9           | mm             |       |
| Lm        | 50     | 30            | mm             |       |

By investigating the results presented in Table 9, the reader can see that, in terms of power density, the gain is 32% due to a lighter active mass part of the IMMG, while the desired output power is obtained. This important power density gain can be explained from two perspectives. One would expect that if the active mass of the IMMG is reduced, the iron losses will also decrease. However, for this IMMG, the mechanical loss component is more important on the MG part (due to the outer rotor configuration), and since the diameter of the PMSM has increased, the same happens for the MG. (To obtain the desired torque, we kept the same volume for the MG part.) On the other hand, to maximize the power density, we accepted an increased level of flux density within the active parts of the IMMG. Since the iron loss component is proportional to the flux density, the level of losses increased slightly. However, the overall efficiency of the IMMG was not affected, and thus, we obtained this important gain in power density without affecting the energetic performance of the propulsion unit. Since we obtained the desired results, we continued with the construction of the IMMG prototype.

### IV. VALIDATION OF THE STUDIED IMMG

The constructed IMMG prototype is presented in Fig. 7. Here, one can see the inner stator (Fig. 7a), the double-sided rotor (Fig. 7b), the static part of the MG (Fig. 7c), the outer rotor of the MG (Fig. 7d) and its passive elements (Fig. 7e). The assembly of the IMMG, having two shafts, two torque and speed levels, is shown in Fig. 7f.

For the validation of the study, we performed several tests. The electromechanical characteristics of the constructed prototype are shown in Fig. 8 for permanent regime operation. For the double rotor variant, two speeds are possible (see Fig. 8-top): 420 rot/min or 1428 rot/min (the outer rotor of the MG). The direct and quadrature axis currents are also plotted since the vector control technique was applied for the supply of the IMMG. With respect to the torque, a 34.1 Nm was obtained by the lower speed rotor, and a 10 Nm was obtained by the higher speed rotor (the outer rotor of the MG). Additionally, a torque ripple of 5% was obtained for the lower speed rotor and 20.57% for the high-speed rotor. The supply of the IMMG is assured via a three-phase inverter. By superimposing the voltage and the current from one phase, a 0.94 power factor was obtained, which is a very good achievement. Moreover, an efficiency of 0.8413 was obtained, which, regarding the fact that the prototype can offer two levels of speed and torque, is a satisfactory achievement.
Some dynamic characteristics are depicted in Fig. 9. Here, one can see that for a given speed ramp, the control follows the imposed value very well (Fig. 9-top), while the low-speed rotor torque, axis current, dc bus voltage and current have realistic values (Fig. 9-bottom). Based on the employed analysis on the studied IMMG, a summary of the results is presented in Table 10.

### TABLE 10. IMMG’s main Performances.

| Parameter                               | Achieved performances | Units |
|-----------------------------------------|-----------------------|-------|
| Output power                            | 1500                  | W     |
| Rated current                           | 28                    | A     |
| Torque ripples, at low-speed rotor      | 5                     | %     |
| Torque ripples, at high-speed rotor     | 20.57                 | %     |
| The power factor of the propulsion      | 0.94                  | ---   |
| The efficiency of the propulsion        | 84.1                  | %     |

According to the efficiency standard IEC 60034-30, we can situate our structure at the IE3 efficiency level, meaning a “premium efficient” electric motorization. However, since this standard reflects the efficiency of classic 50 Hz industrial machines, we can say that our propulsion unit, working at 119 Hz and chopped voltage, offers very good efficiency. Moreover, industrial electric motors at the same power level and at 50 Hz have a common 0.86–0.88 power factor. Our IMMG structure outperforms the standard motors in terms of energetic performance, having a power factor of 0.94. Finally, it can be concluded that the proposed multiport structure (with two levels of torque and speed) produces the application’s expected results.
V. CONCLUSIONS

This study is mainly focused on the optimization of a multipropulsion system containing an integrated motor-magnetic gear (IMMG) structure. This IMMG offers two shafts and two levels of torque and speed (420 r/min and 1428 r/min, respectively).

A dimension-based adaptive differential evolution (DE) algorithm was proposed for the optimization of the IMMG. Specifically, a simple yet efficient approach was introduced to control the mutation scale factor. In addition, a novel dimension-based technique was integrated to control the crossover parameter for each dimension instead of each individual. By comparing our optimization algorithm with some employed approaches found in the literature, a competitive performance was observed for our proposal – it has significantly outperformed recent state-of-the-art DE/non DE algorithms. By applying this algorithm to the proposed IMMG, an important gain of 32 % in power density was obtained.

Next, we constructed and tested the studied IMMG. The desired output power of 1.5 kW was obtained, while the energetic performances were at good levels: 0.841 for the efficiency and 0.94 for the power factor. Additionally, the torque ripples were 5% for the low-speed rotor and 20.57 % for the high-speed rotor. It is worth mentioning that the increased speed option for the proposed IMMG was obtained for the high-speed rotor. It is worth mentioning that the torque ripples were 5% for the low-speed rotor and 20.57 % for the high-speed rotor.

To emphasize the findings of this work, we would like to recall the following:

- The studied IMMG has two shafts, offering two levels of torque and speed.
- This propulsion system permits an increase in the speed without increasing the supplying frequency.
- Since the iron loss component of the electromechanical converter is proportional to the square of the frequency, by keeping the supplying frequency at a lower level, we avoided a decrease in the efficiency of the propulsion unit.
- Very good energetic performances have been obtained, which classifies this IMMG structure at the “premium efficiency” level.
- A novel dimension-based optimization technique was proposed for a differential evolutionary algorithm applied for the optimization of the designed IMMG.
- An important power density gain was obtained after the optimization of the proposed IMMG was employed.

In future work, regarding the optimization algorithm, we are interested in integrating multiple parameter adaptation strategies. Then, switching between them can be possible using reward-based techniques such as historical performance or Q-learning techniques.

APPENDIX - NOMENCLATURE AND ABBREVIATIONS

CR - cross-over parameter.

D - number of variables within the solution vector.

\( f(i) \) and \( f'(i) \) - fitness of the parent and the new individual.

F - mutation parameter.

\( F_{m, in} \) - the magnetomotive force of the MG’s inner part.

\( F_{m, out} \) - the magnetomotive force of the MG’s outer rotor.

\( G_{max} \) - number (maximum) of generations.

\( HS \) - Harmony Search.

i - constraints.

IT - current iteration.

\( l_x \) - a randomly generated number within the range [0, 1].

\( l_m \) - the field length path for specific portion of the MEC.

m - parameter set to calibrate the exploration and the exploitation capabilities of the approach.

\( m_{act} \) - mass of the entire active part of the IMMG.

\( MaxIT \) - maximum number of iterations.

\( m_{cooper} \) - the mass of copper within the IMMG.

\( m_{min} \) - the smallest possible (starting) value for \( m \).

\( m_{steel} \) - the mass of iron within the IMMG.

\( m_{magnetics} \) - the mass of magnets within the IMMG.

\( n_g \) - number of elements of the MEC on the rotor span.

\( NP \) - population size.

\( NP_{init} \) - starting population size.

\( NP_{min} \) - the smallest possible population size.

\( P_{in} \) - the input power of the IMMG.

\( P_{out} \) - the output power of the IMMG.

\( r_1, r_2, r_3 \) - different randomly generated numbers within the range [0, 1].

\( R_1 \) - a randomly generated number within the range [0, 1].

\( R_{m, x} \) - magnetic reluctance on the \( x \) portion of the magnetic circuit (while \( x \) stands for air, iron yoke & teeth, magnet).

\( R_{m, eq} \) - the equivalent magnetic reluctance of the MEC.

\( S_f \) - successful set of F parameters.

\( S_x \) - area of the \( x \) portion of the MEC.

\( T_0 \) - the time needed to execute the test problem.

\( T_{test} \) - the time to execute 200,000 evaluations of the function \( f \) by itself with \( D \) dimensions.

\( T_{in} \) - average time to execute DADE with 200,000 evaluations of \( f \) in \( D \) dimensions for 5 times.

\( T_{out} \) - the torque of the outer part of the integrated MG.

\( u_{G+1} \) - trial vector.

\( u_{G+1} \) - mutant vector.

\( x_{best} \) - best individual in the population at generation \( G \).

\( x^G \) - solution vector, or solution \( i \) at generation \( G \).

\( x_{pbest} \) and \( x_{pworst} \) - randomly selected from the best and the worst 25% solutions in the current population.

\( x_{r1...r4} \) - a randomly selected individuals from the current population.

\( \sum \text{losses} \) - amount of losses within the IMMG.

\( \Phi_{m, in} \) - the magnetic flux of the inner part of the MG.

\( \mu_f \) - location parameter randomly selected within the optimization process.

\( \mu_s \) - the magnetic permeability of the \( x \) portion of the MEC.
\( \sigma_i \) - a randomly generated number within the range \([1, D]\) within the optimization process.

\( AGSK \) = Adaptive Gaining Sharing Knowledge Based Algorithm.

\( DADE \) = Dimension-based adaptive differential evolution.

\( DE \) = Differential Evolution.

\( DR \) = dual rotor.

\( DR\text{-}PMSM \) = DR Permanent Magnet Synchronous Machine.

\( DR\text{-}SynRM \) = DR Synchronous Reluctance Machine.

\( DR\text{-}PMSG \) = DR Permanent Magnet Synchronous Generator.

\( DR\text{-}HESM \) = DR Hybrid Excited Synchronous Machine.

\( DR\text{-}FSSM \) = DR Flux Switching Synchronous Machine.

\( EBO \) with \( CMAR \) = Effective Butterfly Optimizer with Covariance Matrix Adapted Retreat Phase.

\( EVs \) = electric vehicles.

\( IMMG \) = integrated motor-magnetic gear.

\( j2020 \) = Differential Evolution Algorithm for Single Objective Bound-Constrained Optimization Algorithm.

\( JADE \) = Joint Approximate Diagonalization of Eiden matrices.

\( LSHADE\text{-}SPACMA \) = LSHADE with Semi-Parameter Adaptation and Covariance Matrix Adaptation.

\( LSHADE\text{-}enEpSpM \) = LSHADE with an Ensemble Sinusoidal Parameter Adaptation.

\( MEC \) = magnetic equivalent circuit.

\( MGs \) = magnetic gears.

\( PMs \) = permanent magnets.

\( PMSM \) = Permanent Magnet Synchronous Machine.

\( SHADE \) = Success-History based Adaptive Differential Evolution.

**REFERENCES**

[1] D. Fodorean, *State of the art of magnetic gears, their design, and characteristics with respect to EV application*, book chapter (in Modeling and Simulation for EV Applications, Intech (Croatia) 2016).

[2] L. Jing, J. Gong, Z. Huang, T. Ben and Y. Huang, “A New Structure for the Magnetic Gear”, *IEEE Access*, vol. 7, pp. 75550-75555, 2019, DOI: 10.1109/ACCESS.2019.2919679.

[3] M. Bilal, J. Ikram, A. Fida, S. S. H. Bukhari, N. Haider and J. -S. Ro, “Performance Improvement of Dual Stator Axial Flux Spoke Type Permanent Magnet Vernier Machine”, *IEEE Access*, vol. 9, pp. 64179-64188, 2021, DOI: 10.1109/ACCESS.2021.3076111.

[4] S. Xie et al., “A Consequent-Pole Magnetic-Geared Machine with Axially Embedded Permanent Magnets for Hybrid Electric Vehicle”, *IEEE Access*, vol. 9, pp. 14905-14917, 2021, DOI: 10.1109/ACCESS.2021.3052914.

[5] G. Schüssler, and J. Lindner, “Magnetic parallel shaft drive for contactless transmission of torque - uses wheels with magnets at circumference at least two of which are rotationally located on shafts”, DE2423826 DE Patent, 01 27, 1994.

[6] J. K. Henderson, and B. B. Bushman, “Variable ratio angled magnetic drive”, US Patent, 7 25, 2002.

[7] X. Liu, H. Feng and X. Wang, “Study on the Harmonic Loss of Permanent Magnet Toroidal Space Motor for Hybrid Electric Vehicles”, *IEEE Access*, vol. 9, pp. 51475-51483, 2021, DOI: 10.1109/ACCESS.2021.3069851.

[8] M. A. Dranca, M. Chirca, S. I. Cosman, S. Breban, “Design Analysis of a Permanent Magnet Brushless Generator with Two Counter Rotating Rotors for Small-Wind Turbine”, *IEEE International Conference on Applied and Theoretical Electricity (ICATE'18)*, 4-6 October 2018, DOI: 10.1109/ICATE.2018.8551431.

[9] J. Bai, P. Zheng, B. Yu, L. Cheng, S. Zhang and Z. Liu, “Investigation of a Magnetic-Field Modulated Brushless Double-Rotor Machine With the Same Polarity of PM Rotor”, *IEEE Transactions on Magnetics*, vol. 51, no. 11, pp. 1-4, NoV. 2015, Art no. 8110004, DOI: 10.1109/TMAG.2015.2441376.

[10] M. Al-Ani, H. Mahmoud, M. Degano and C. Gerada, “Double Rotor Synchronous Reluctance Machine: Analysis towards Torque Capability”, 2018 XIII International Conference on Electrical Machines (ICEM), 2018, pp. 2144-2149, DOI: 10.1109/ICELMACH.2018.8506898.

[11] L. Jian, K. T. Chau and J. Z. Jiang, “A Magnetic-Geared Outer-Rotor Permanent-Magnet Brushless Machine for Wind Power Generation”, *IEEE Transactions on Industry Applications*, vol. 45, no. 3, pp. 954-962, May-June 2009, DOI: 10.1109/TIA.2009.2018974.

[12] J. Druant, H. Vansompel, F. De Belie, J. Melkebeek and P. Sergeant, “Torque Analysis on a Double Rotor Electrical Variable Transmission With Hybrid Excitation”, *IEEE Transactions on Industrial Electronics*, vol. 64, n.1, pp. 60-68, Jan. 2017, DOI: 10.1109/TIE.2016.2608786.

[13] J. Kwon, N. Baloch and B. Kwon, “High Gear Ratio Flux Switching Permanent Magnet Machine for High Torque Performance”, *IEEE Access*, vol. 8, pp. 121630-121636, 2020, DOI: 10.1109/ACCESS.2020.3006196.

[14] Claudia V. Pop, and D. Fodorean, “Modelling of an in-wheel motor with integrated magnetic gear for electric vehicle applications”, *IEEE International Conference and Exposition on Electrical and Power Engineering (EPE)*, Oct. 2016, DOI: 10.1109/ICEPE.2016.7781340.

[15] A Kai Qin, Vicky Ling Huang, and Ponnuthurai N Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization”, *IEEE Transactions on Evolutionary Computation*, vol.13, nr.2, pp.398-417, February 2009.

[16] R. Storn, “On the usage of differential evolution for function optimization”, *IEEE Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, pp.519-523, 1996.

[17] R. Tanabe, and Alex S Fukunaga. Improving the search performance of SHADE using linear population size reduction, *IEEE Transactions on Evolutionary Computation (CEC 2014)*, pages 1658–1665, 2014.

[18] V. Stanovov, S. Akhmedova, and E. Semenkin. “Differential evolution with linear bias reduction in parameter adaptation”, *Algorithms*, vol.13, nr.11, November 2020.

[19] A. W. Mohamed, A. A. Hadi, and K. M. Jambi, “Novel mutation strategy for enhancing shade and lshade algorithms for global numerical optimization”, *Swarm and Evolutionary Computation*, vol.50, paper number 100455,2019.

[20] J. Zhang, and A. C. Sanderson, “JADE: adaptive differential evolution with optional external archive”, *IEEE Transactions on Evolutionary Computation*, vol.13, nr.5, pp.945-958, May 2009.

[21] R. Tanabe, and A. Fukunaga, “Success-history based parameter adaptation for differential evolution”, *IEEE Evolutionary Computation (CEC 2013)*, June 2013, Cancun, Mexie, pp.71-78.

[22] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, “Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems”, *IEEE Transactions on Evolutionary Computation*, vol.10, nr.6, pp.646-657, June 2006.

[23] S. Das, S. S. Mullick, and P. N. Suganthan,”Recent advances in differential evolution—an updated survey”, *Swarm and Evolutionary Computation*, vol.27, pp.1-30, 2016.

[24] R. Mallipeddi. “Harmony search based parameter ensemble adaptation for differential evolution”, *Journal of Applied Mathematics*, 2013.

[25] J. Brest, M. S. Maucec, and B. Boskovic, “Differential evolution algorithm for single objective bound-constrained optimization: Algorithm 2020”, *IEEE Congress on Evolutionary Computation (CEC 2020)*, pp.1–8, 2020.

[26] A. W. Mohamed, A. A. Hadi, A. K. Mohamed, and N. H. Awad, “Evaluating the performance of adaptive gaining sharing knowledge based algorithm on CEC 2020 benchmark problems”, *IEEE Congress on Evolutionary Computation (CEC 2020)*, pp.1–8, 2020.
[27] A. Kumar, R. Kumar Misra, and D. Singh, “Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase”, *IEEE congress on Evolutionary Computation* (CEC 2017), pp.1835–1842. 2017.

[28] A. W. Mohamed, A. A. Hadi, A. M. Fattouh, and K. M. Jambi, “Lshade with semi-parameter adaptation hybrid with cma-es for solving CEC 2017 benchmark problems”, *IEEE Congress on Evolutionary computation* (CEC 2017), pages 145–152. IEEE, 2017.

[29] N. H. Awad, M. Z. Ali, and P. N. Suganthan. “Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC 2017 benchmark problems”, *IEEE Congress on Evolutionary Computation* (CEC 2017), pp.372–379. 2017.

[30] S. Giurgea, D. Fodoran, G. Cirrincione, A. Miraoui and M. Cirrincione: “Multimodel Optimization Based on the Response Surface of the Reduced FEM Simulation Model With Application to a PMSM”, IEEE Transactions on Magnetics, vol.44, n°9, pp.2153-2157, September 2008.