Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

MOCHIO: a novel Multi-Objective Coronavirus Herd Immunity Optimization algorithm for solving brushless direct current wheel motor design optimization problem

C. Kumar, D. Magdalin Mary & T. Gunasekar

To cite this article: C. Kumar, D. Magdalin Mary & T. Gunasekar (2022) MOCHIO: a novel Multi-Objective Coronavirus Herd Immunity Optimization algorithm for solving brushless direct current wheel motor design optimization problem, Automatika, 63:1, 149-170, DOI: 10.1080/00051144.2021.2014035

To link to this article: https://doi.org/10.1080/00051144.2021.2014035

Published online: 10 Dec 2021.

AUTOMATIKA
2022, VOL. 63, NO. 1, 149–170
https://doi.org/10.1080/00051144.2021.2014035

MOCHIO: a novel Multi-Objective Coronavirus Herd Immunity Optimization algorithm for solving brushless direct current wheel motor design optimization problem

C. Kumar [a], D. Magdalin Mary[b] and T. Gunasekar[c]

[a]Department of Electrical and Electronics Engineering, M. Kumarasamy College of Engineering, Karur, Tamil Nadu, India; [b]Department of Electrical and Electronics Engineering, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India; [c]Department of Electrical and Electronics Engineering, Kongu Engineering College, Erode, Tamil Nadu, India

ABSTRACT

A prominent and realistic problem in magnetics is the optimal design of a brushless direct current (BLDC) motor. A key challenge is designing a BLDC motor to function efficiently with a minimum cost of materials to achieve maximum efficiency. Recently, a new metaheuristic optimization algorithm called the Coronavirus Herd Immunity Optimizer (CHIO) is reported for solving global optimization problems. The inspiration for this technique derives from the idea of herd immunity as a way of combating the coronavirus pandemic. A variant of CHIO called Multi-Objective Coronavirus Herd Immunity Optimizer (MOCHIO) is proposed in this paper, and it is applied to optimize the BLDC motor design optimization problem. A static penalty constraint handling is introduced to handle the constraints, and a fuzzy-based membership function has been introduced to find the best compromise results. The BLDC motor design problem has two main objectives: minimizing the motor mass and maximizing the efficiency with five constraints and five decision/design variables. First, MOCHIO is tested with benchmark functions and then applied to the BLDC motor design problem. The experimental results are compared with other competitors are presented to confirm the viability and dominance of the MOCHIO. Further, six performance metrics are calculated for all algorithms to assess the performances.

ARTICLE HISTORY
Received 19 February 2021
Accepted 29 November 2021

KEYWORDS
BLDC motor; magnetics; metaheuristic; Multi-Objective Coronavirus Herd Immunity Optimizer (MOCHIO); multi-objective optimization

1. Introduction

Electric vehicles (EVs) are seen as an exciting choice on the low-emission vehicle pathway that can allow the transportation sector to reduce greenhouse gas emissions dramatically [1,2]. As a result, the conventional direct-current (DC) motors can still deliver constant power during high-speed operation and high performance during the wide operating speed range, restricting EV applications [3]. The most preferred choice for EVs is currently the BLDC motor. In addition, the BLDC motors are now commonly used in modern industries due to their advantages, such as high performance, light-weight, and simple structure [4]. The BLDC motors are operated by an integrated switching power supply unit/inverter by a DC source, which converts an alternating-current (AC) signal to drive the motor. Furthermore, because of the potential in the speed and torque domain, the BLDC motors are more flexible [5]. Therefore, in current decades, a great deal of ongoing research has been dedicated to designing a BLDC motor. The involvement in BLDC motor design has recently gone up from both theoretical and experimental perspectives [6,7]. A BLDC motor that drives a vehicle with solar photovoltaic source during a competition is discussed in this study. The material and production expenses are not necessary, while the main focus is on axial bulk and motor efficiency [8]. An analytical model of the BLDC motor is used in this present study as a benchmark problem composed of 78 non-linear expressions with five inequality constraints and five design variables. Alternatively, five design variables are optimized to minimize the motor mass and maximize motor efficiency and meet five constraints of inequality at once. A multi-objective optimization approach is essential to accomplish these artifacts [9,10]. Few researchers have optimized the BLDC motor design optimization problem using various single-objective algorithms. The authors of [10] have discussed four different metaheuristic algorithms for the BLDC motor design problem. However, the authors have considered a multi-objective problem as a single-objective problem. The authors of [11] have applied the PSO algorithm for optimizing the torque density in the BLDC motor. The authors of [12] have compared the performance of three optimization algorithms, such as gradient-based, direct-search, and genetic algorithms for BLDC motor design. The authors have concluded that the genetic algorithm

CONTACT D. Magdalin Mary magdalinmary04@gmail.com Department of Electrical and Electronics Engineering, Sri Krishna College of Technology, Coimbatore, Tamil Nadu 641042, India

is suitable for the BLDC motor design problem. The authors of [13] have applied DE algorithm combined with MotorCAD and SPEED tools to design BLDC motors for ultralight aircraft propulsion systems. The authors of [14] have used an improved bees algorithm for the BLDC motor design problem. However, the performance of the improved bees algorithm is not compared with other algorithms. The authors of [15] have applied a genetic algorithm for the BLDC motor design optimization. The authors of [16] have applied a genetic algorithm and grey wolf algorithm for BLDC motor design with an efficient ratio of slots per pole.

Evolutionary algorithms (EA) have indeed been commonly used in several optimization problems in recent years. The techniques, such as differential evolution [17], genetic algorithm [18], and particle swarm optimization (PSO) [19], are few examples of EA utilized in electromagnetic field applications. In the meantime, Swarm Intelligence (SI) is involved with an attractive figure in the area of applied electromagnetics. Numerous bio-inspired SI algorithms, such as Multi-Objective Particle Swarm Optimization (MOPSO) [8], non-dominated sorting genetic algorithm Version-II [20], bat algorithm [21] and its variant Multi-Objective Bat Algorithm (MOBA) [17], krill herd optimizer [22] and it's variant Multi-objective Krill Herd Optimizer (MOKHO) [22], Multi-Objective Grey Wolf Optimizer (MOGWO) [23], Multi-Objective Whale Optimization Algorithm (MOWOA) [24], Multi-Objective Moth Flame Optimizer (MOMFO) [25,26], predator-prey biogeography-based optimization [27], imperialist competitive optimizer [28] and its variant called multi-objective modified imperialist competitive optimizer [28], pigeon-inspired optimizer [20], and it's variant multi-objective pigeon-inspired optimizer [29], and sequential quadratic programming [30] have been directly applied to the design problem of BLDC motor. The optimal solutions are a group of non-dominated Pareto with the best trade-off between two or more cost functions placed on the Pareto front in multi-objective optimization problems (MOOPs). Several real optimization engineering issues with science and technology advancement contribute to bio-inspired algorithms' boom. Five control variables must be optimized for the BLDC motor design problem to attain an satisfactory trade-off among the total mass and efficiency [31]. In addition, many variants of multi-objective algorithms are reported by various researchers for global multi-objective problems and real-world engineering optimization problems. For instance, multi-objective arithmetic optimizer [32], multi-objective slime mould algorithm [33], direction-based multi-objective evolutionary algorithm [34], multi-objective gradient-based optimizer [35], decomposition-based multiobjective evolutionary optimization [36], multi-objective plasma generation optimizer [37], improved multiobjective particle swarm optimization algorithm [38], etc., are applied for many multi-objective optimization problems. Nevertheless, none of those mentioned above algorithms are verified BLDC motor optimization problems.

The coronavirus herd-immunity optimizer's motivation derives from the idea of herd immunity as a strategy to fight the coronavirus pandemic [39]. The spread of infection with Coronavirus depends on how well the infected people directly encounter other community members. Health professionals recommend social distancing to shield other people in society from the infection. Herd immunity is a condition that the society enters while much of the community is resistant, contributing to the prevention of transmission of diseases. In terms of computation notions, these definitions are modeled. Three kinds of human cases for herd immunity are used: infected, susceptible, and immunized. It is to decide how well the newly formed solution with social distancing approaches updates its chromosomes. The performance of the coronavirus herd-immunity optimizer (CHIO) algorithm is tested for various benchmark functions and proved its superiority in handling the single-objective optimization problems. The CHIO algorithm performs better than other well-known competitors, such as grey wolf optimizer, bat algorithm, salp swarm optimization, harris hawk optimizer, GA, particle swarm optimizer, and artificial bee colony optimizer, as discussed in the original CHIO paper. In specific single-objective optimization problems, it is evident that the CHIO algorithm has certain advantages. So far, the CHIO algorithm is implemented for few engineering applications, such as vehicle routing problems [40] and non-hierarchical grouping of objects [41]. Still, the researchers are utilizing the CHIO algorithm in various applications. The CHIO algorithm still shrinks down, however, at the prospect of multi-objective problems. So far, the multi-objective variant of the CHIO algorithm is not developed and analyzed for real-world engineering optimization problems. As per the statement of the no-free-lunch theory, no single algorithm is suitable for all optimization problems. This statement motivated us to develop a new algorithm for the BLDC motor design optimization problem [42]. Inspired by [32,37,43], a multi-objective version called Multi-Objective Coronavirus Herd Immunity Optimizer (MOCHIO) is proposed to extend its scope in various fields. The crowding distance assignment and non-dominated procedure ensure that the selection process is elitist and efficient. For better solution accuracy, the MOCHIO is then implemented to optimize the BLDC motor design parameters. From the quantitative outcomes against MOGWO, MOBA, MOPSO, MOWOA, and MOMFO, in the view of MOOPs, MOCHIO seems to have superiority in homogeneity and degree of the Pareto front. The key contributions of this study are as follows.

- A new multi-objective algorithm called the MOCHIO algorithm is formulated using the crowding-distance and non-dominated sorting mechanism concepts.
- A fuzzy-based membership function is introduced to find the best compromise results
- A static penalty constraint handling approach is introduced to handle the equality and inequality constraints.
- The performance of the proposed algorithm is tested on unconstrained and constrained multi-objective benchmark test functions.
- Implementation of MOCHIO algorithm for the BLDC wheel motor design optimization problem and the performance of the proposed algorithms is compared with other competitive algorithms.
- The six performance metrics are calculated for all test problems and compared among selected optimization algorithms.

As follows, the remainder of the paper is organized. The principles of the CHIO algorithm and conversion of the single-objective CHIO algorithm to the multi-objective CHIO algorithm are discussed in Section 2. The problem formulation of the BLDC motor design and its constraints are discussed in section 3. The simulation results of MOCHIO for various benchmark functions and BLDC motor design problems are discussed in Section 4. In addition, the statistical analysis of all algorithms is also presented. Section 5 discusses the concluding remarks.

2. Single and multi-objective Coronavirus Herd Immunity Optimizer algorithm

2.1. Basic Coronavirus Herd Immunity Optimizer (CHIO) algorithm

The idea of herd immunity is mathematically modelled to construct the basic CHIO algorithm [39]. The methodology builds on the assumption of protecting society from the infection by transforming the bulk of the non-infected susceptible population to immunity. Consequently, even the remaining susceptible cases would not be affected since the immune population will no longer spread this infection. The population of individuals with herd immunity can be divided into susceptible, infected, and resistant. The formulation of CHIO is based on the herd immunity population, as depicted in Figure 1. For more details, please refer to [39]. As seen in the CHIO algorithm's implementation procedure, the improvement approach is extracted from susceptible, contaminated, and immunized individuals.

In the CHIO algorithm, social distance is defined by separating the current individual and a specified person from the community who could be susceptible, infected, or immunized. The herd immunity approach is modelled on the optimization technique in the CHIO algorithm. There are six primary steps to develop the algorithm. The implementation procedure is as follows.

Step-1: Initialize CHIO parameters and the problem formulation – The objective function of CHIO is as follows:

$$\min_{x} f(xx) xx \in [lblb, ubub] \tag{1}$$

where n signifies the number of genes in each individual, the objective function is formulated for all the individual, $(x_i = x_1, x_2, \ldots, x_n)$, in which x_i is the decision variable with index "i". The CHIO algorithm has two control parameters, such as maximum infected cases age (Max_{Age}) and basic reproduction rate (BR_r), and four algorithmic parameters, such as C_0 ($=1$), Max_Itr (Maximum number of iteration), HIS (Herd Immunity Size), and n (dimension).

Step-2: Produce Herd Immunity Population (HIP) – Originally, the algorithm produces a set of populations as much as HIS heuristically. The individuals produced are kept in the HIP as a two-dimensional matrix, as follows:

$$HIP = \begin{bmatrix} x_1^1 & \cdots & x_n^1 \\ \vdots & \ddots & \vdots \\ x_1^{HIS} & \cdots & x_n^{HIS} \end{bmatrix} \tag{2}$$

Using Eq. 1, the optimal solution for each individual is determined. In addition, for all individuals in the HIP, the status vector (S) is also established by either one or zero. Notice that the number of ones in S is originated at random as much as C_0.

Step-3: Herd Immunity Evolution – This is CHIO's primary enhancement loop. According to the BR_r, the individual gene maintains the same or is influenced by social distance using three following rules.

$$x_i^j(t+1) = \begin{cases} x_i^j(t), r \geq BR_r \\ C(x_i^j(t)), r < \dfrac{1}{3} \times BR_r \text{ (Infected)} \\ N(x_i^j(t)), r < \dfrac{2}{3} \times BR_r \text{ (Susceptible)} \\ R(x_i^j(t)), r < BR_r \text{ (Immuned)} \end{cases} \tag{3}$$

where r denotes random number between [0,1]. The infected case is in the range of 0 to $\frac{1}{3} \times BR_r$. The new gene's value is reduced by social distancing, and it is derived by finding the difference between the gene from the infected case and the current gene as follows.

$$x_i^j(t+1) = C(x_i^j(t)) \tag{4}$$

$$C(x_i^j(t)) = x_i^j(t) + r \times (x_i^j(t) - x_i^c(t)) \tag{5}$$

Note that $x_i^c(t)$ is selected randomly from the diseased person based on the S. The susceptible case is in the range of $\frac{1}{3} \times BR_r$ to $\frac{2}{3} \times BR_r$, the value of the new gene

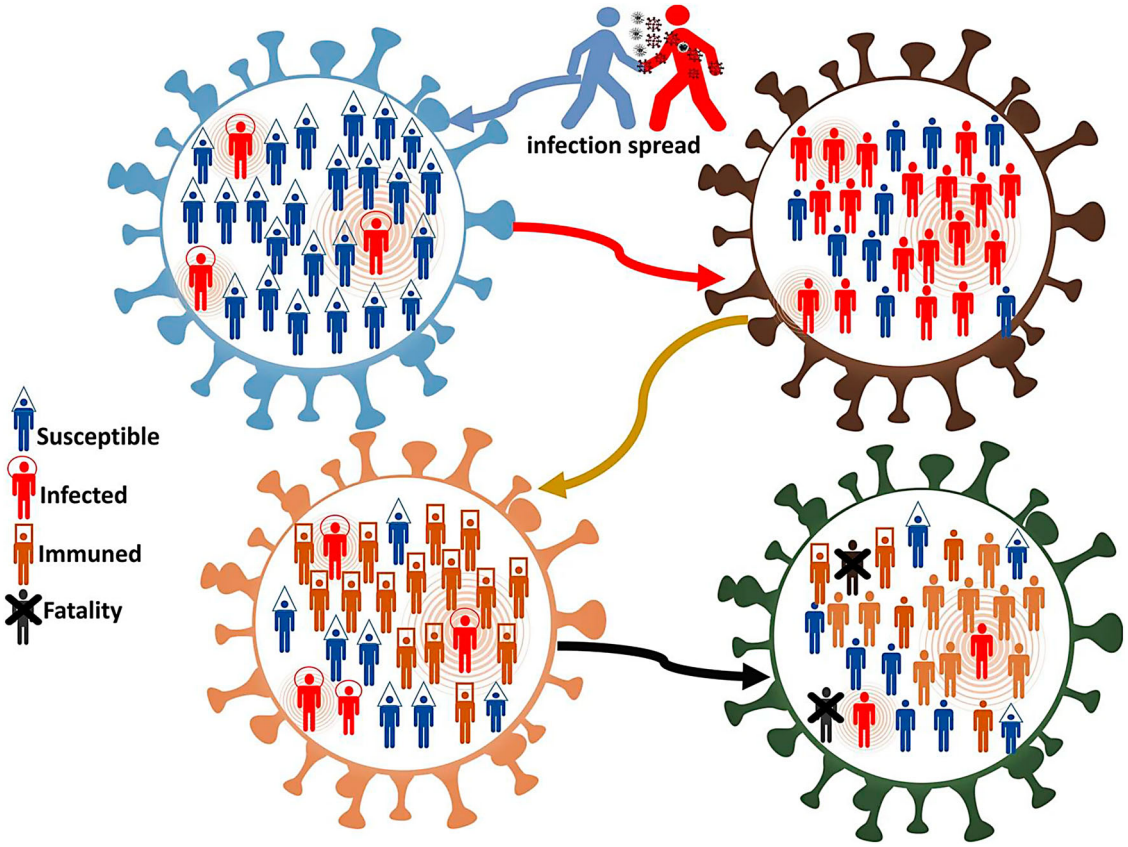

Figure 1. Herd immunity population [39]

is reduced by social distancing, and it is derived by finding the difference between the gene from the infected case and the current gene as follows.

$$x_i^j(t+1) = N(x_i^j(t)) \tag{6}$$

$$N(x_i^j(t)) = x_i^j(t) + r \times (x_i^j(t) - x_i^m(t)) \tag{7}$$

Note that $x_i^m(t)$ is selected randomly from the vulnerable case based on the S. The immune case is in the range of $\frac{2}{3} \times BR_r$ to BR_r, the value of the new gene is reduced by social distancing, and it is derived by finding the difference between the gene from the infected case and the current gene as follows.

$$x_i^j(t+1) = R(x_i^j(t)) \tag{8}$$

$$R(x_i^j(t)) = x_i^j(t) + r \times (x_i^j(t) - x_i^v(t)) \tag{9}$$

Note that $x_i^v(t)$ is feast from the best-immuned person based on the S such that,

$$f(x^v) = arg \min_{j \sim \{k|\mathcal{S}_k=2\}} f(x^j) \tag{10}$$

Step-4: Population Update – During each generated case, the immunity rate is determined, and the current solution is substituted by the generated person only if $f(xx^j(t+1)) < f(xx^j(t))$. The value of the age vector (A_j) is raised to one if the value of the status vector S_j is equal to one. During each iteration, the value of S_j is updated based on the threshold of herd immune using the following equation.

$$\mathcal{S}_j = \begin{cases} 1, f(xx^j(t+1)) < \dfrac{f(xx)^j(t+1)}{\Delta f(xx)} \\ \quad \wedge \mathcal{S}_j = 0 \wedge is\_Corona(xx^j(t+1)) \\ 2, f(xx^j(t+1)) > \dfrac{f(xx)^j(t+1)}{\Delta f(xx)} \wedge \mathcal{S}_j = 1 \end{cases} \tag{11}$$

where is_Corona$(xx^j(t+1))$ is equal to one, which is a binary value if a new case has inherited a benefit from any cases infected.

Step-5: Casualty Cases – If for a specified iteration, as defined by the *Max_Age* parameter, the immune rate of the present affected case could not increase, then this process is considered deceased. It is then regenerated from scratch using $x_i^j(t+1) = lb\_i + (ub\_i - lb\_i) \times U(0,1)$. Besides, the value of S_j and A_j is set as zero. It can be beneficial in expanding the present population and thereby avoiding local optimum solutions.

Step-6: Stopping condition – The CHIO algorithm performs steps 3–5 until the stopping criterion is met, usually depending on the maximum number of iterations (IT_{max}). In this situation, the population is

dominated by the total number of immune and susceptible cases. Also, the contaminated cases are removed.

2.2. Multi-Objective Coronavirus Herd Immunity Optimizer (MOCHIO) algorithm

In order to construct the MOCHIO algorithm, an archive was built into the algorithm, and it is used to store the optimal solution during each run. This process of archiving would be similar to the multi-objective variant of PSO. The search method in MOCHIO is very similar to the basic single-objective version of CHIO, in which solutions are built using the herd immunity process. The optimal solutions are stored in an archive, and the optimal solutions are selected from the Archive using the leader screening process. The MOCHIO algorithm can be used to minimize (or maximize) the objective functions as follows.

$$\begin{aligned} &\text{Min/Max}, F(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x})\} \\ &\textit{Subject to} : p_j(\vec{x}) \geq 0, j = 1, 2, \ldots, g \\ &q_j(\vec{x}) = 0, j = 1, 2, \ldots, h \\ &lb_j \leq x_j \leq ub_j, j = 1, 2, \ldots, n \end{aligned} \tag{12}$$

where p and q are the inequality and equality constraints of the given problem. Several algorithms to comply with MOOPs, such as NSGA-II [44–48] and MOBA [9], the Pareto sorting scheme has indeed been involved. In NSGA-II, the initial population transforms in order to achieve a genotype population and a set of genotypes [49]. The MOCHIO algorithm utilizes the elitist framework to determine the individual's efficiency because of its strong results in balancing exploration and exploitation. The following steps are the basis for implementing the sorting scheme.

Step-1: Non-dominated sorting operator – If and only if any of the following conditions are met, the position of the individual X_i is said to control the position of the individual X_j[8].

$$\begin{cases} f_l(X_i) \preceq f_l(X_j), \textit{ for all} l = 0, 1, 2, \ldots, n \\ f_{\vec{l}}(X_i) \preceq f_{\vec{l}}(X_j), \textit{ for at least one } \vec{l} \in \{0, 1, 2, \ldots, n\} \end{cases} \tag{13}$$

As shown in Figure 2, the individuals would be split by the non-dominated sorting operator, S_1^X, S_2^X and so on into various sets. The surface generated by solutions is known as the Pareto frontier in the best non-dominated set S_1^X.

Step-II: Crowded Comparison Operator (CCO) – After all the individuals have been grouped into m sets, by linking the Crowding Distance (CD) of individual locations, the CCO remains to rank in each set. The CD is defined as follows.

$$CD_j^i = \frac{f_l(X_{i+1}) - f_l(X_{i-1})}{f_l^{max} - f_l^{min}} \tag{14}$$

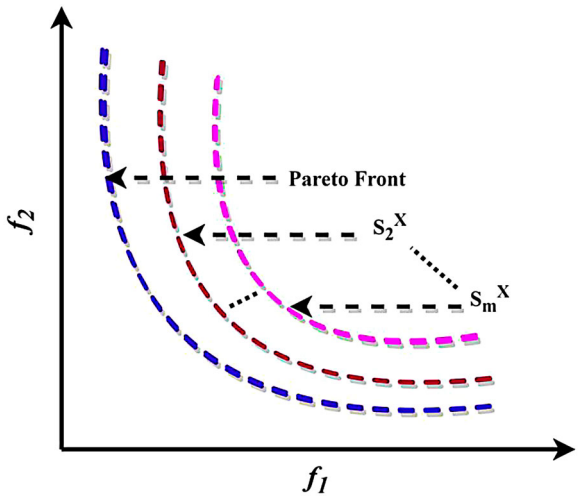

Figure 2. Different sets divided by CCO

Where, f_l^{min} and f_l^{max} are the minimum and maximum values of l^{th} fitness function. The higher CD is considered to be optimal in order to guarantee a variety of solutions. The MOOP has access to the range of Pareto sorting scheme solutions rather than contrasting fitness values in single-objective optimization. The individuals can go through two operations, as seen in Figure 3. First of all, they split into separate sets by the non-dominated sorting operator. In every set, the CCO proceeds to rate the individuals. As a consequence, Pareto's sorting scheme generates a series of individuals in descending order.

2.3. Implementation procedure of MOCHIO algorithm

The flowchart of the suggested MOCHIO algorithm is depicted in Figure 4. The following is the implementation of the proposed MOCHIO algorithm.

(1) Initialize the CHIO algorithm parameters, such as spreading rate BR_r, HIS, maximum age Max_{age}, and S_f. The maximum number of iterations is represented as IT_{max}.
(2) Assess the positions of the herd immune individual X via the Pareto sorting mechanism. Find the non-dominated solutions and store them in the Pareto Archive, and the crowding distance is calculated for each member of the Archive.
(3) The sorting scheme for Pareto is used to evaluate the best individual (only non-dominated solutions) in the Archive, and dominated solutions are deleted from the Archive.
(4) The position of the population is updated using Eq. 11 in the CHIO algorithm.
(5) Update the iteration loop t by $t = t+1$.
(6) If t is smaller than IT_{max}, return to step-2. Else the current positions are evaluated and return the optimal Pareto front $S_1{}^X$.

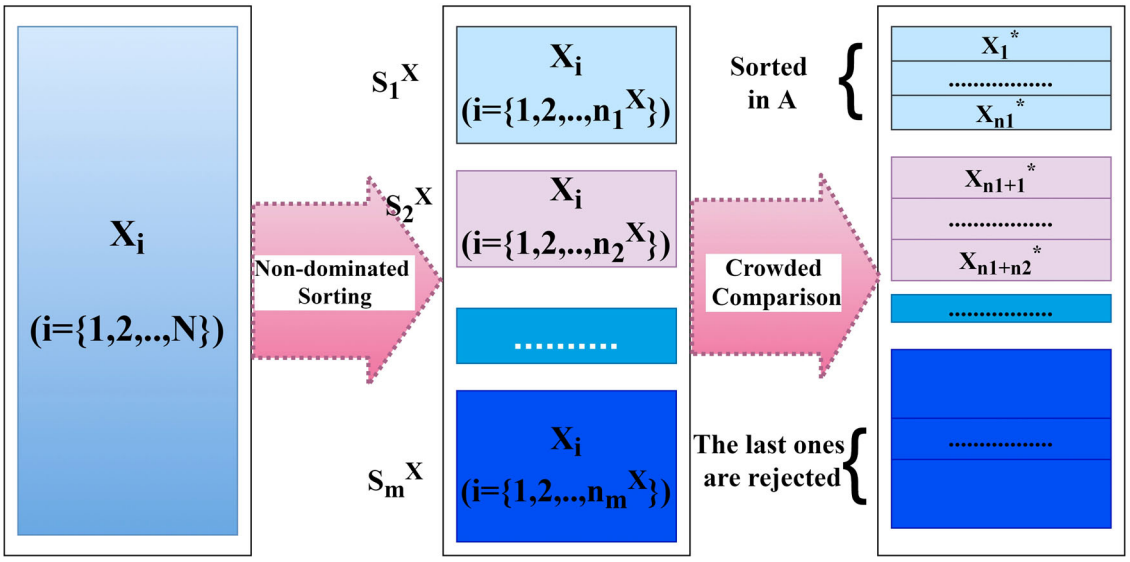

Figure 3. Non-dominated sorting scheme for MOCHIO algorithm

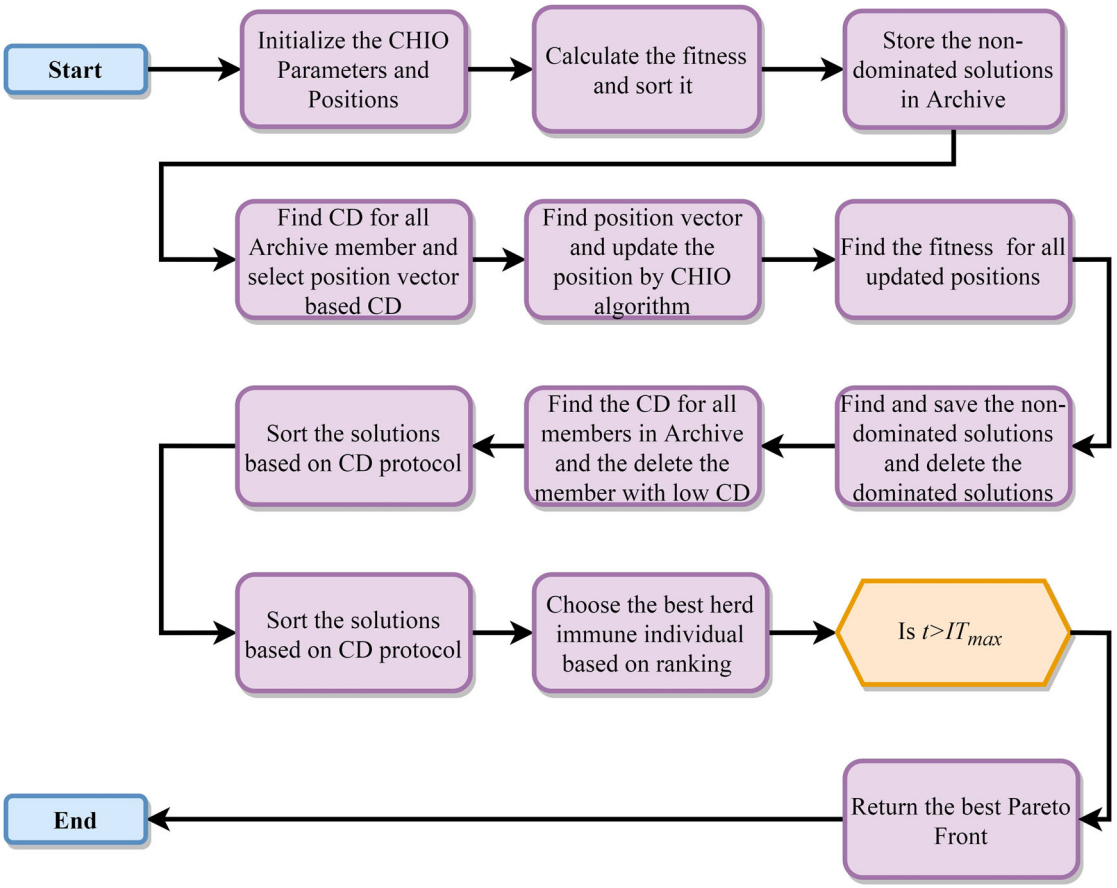

Figure 4. Flowchart of the multi-objective CHIO algorithm

3. BLDC motor design optimization problem

This segment of the paper presents the basic concepts of the BLDC wheel motor drive, mathematical modelling of the BLDC wheel motor, and problem formulation for the optimization process.

3.1. Basics of BLDC motor

The structure and geometry of the BLDC motor, schematically shown in Figure 5, is commonly used when the application requires high torque-inertia and torque-volume ratios along with a high starting torque

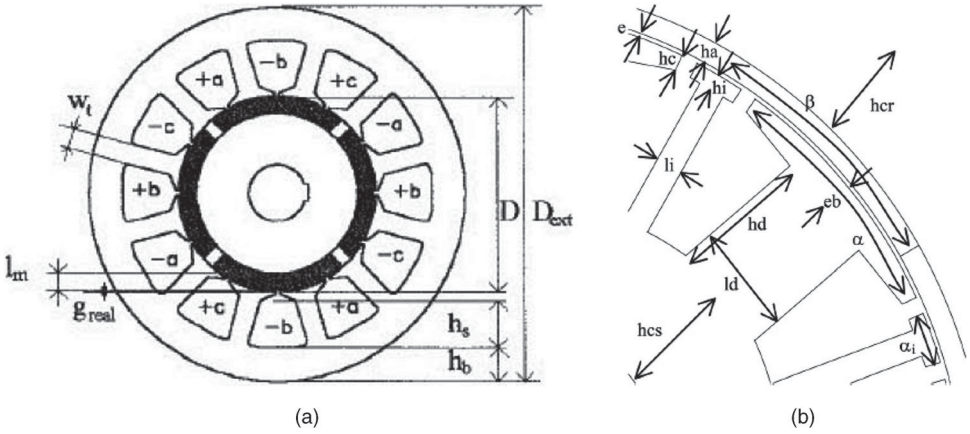

Figure 5. BLDC wheel motor; (a) Structure of the motor, (b) Geometry of the motor [27,50]

and frequent running. Its high versatility to be configured in many ways is another unique feature of the PM brushless motor. This work uses a combined method with thermal analysis to design a BLDC motor [50,51]. The design technique is assisted by a steady-state study of a BLDC motor. As a function of the working conditions and machine dimensions, it gives the equations of the magnetic, electrical, thermal, and mechanical quantities. The motor design is thus accomplished by solving a system of non-linear equations derived from the aforementioned operating study, where, as in [27], motor efficiency, material stress limits, and other restrictions are applied. An optimization method to configure the best motor configuration has been easily implemented, taking advantage of the fully analytical design process.

This paper analyzes the optimization of a BLDC wheel motor for a solar racecar [52]. Figure 3(b) depicts the geometry of the BLDC motor. Three benchmarks are suggested by the authors of [52], in which the first benchmark has five design variables, the second benchmark has 10 design variables, and the third benchmark has 11 design variables, with various constraints and a single objective function. The pole pair is related to the fundamental discrepancy between the second benchmark problem and the third benchmark problem proposed in [52]. The number of poles is fixed as six in the second benchmark problem, and the number of poles in the third benchmark problem varies between [4, 6]. An analytical model is then adopted for the benchmarks. In the design phase, this would be very effective. So, it neglects eddy current and saturation in the magnets and other effects.

3.2. Mathematical modelling of BLDC wheel motor

This section discusses the mathematical modelling of the BLDC wheel motor. A wheel drive that accelerates a car during racing is the motor that needs to be designed and analyzed. The analytical modelling of the BLDC wheel motor is carried out in this paper, as discussed by authors of [52]. This circumstance influences the design and implementation possibilities. However, materials and production costs aren't as important as motor performance and axial weight. As a result, a BLDC motor with surface Samarium Cobalt magnets, focused radial flux, windings, and an outrunner rotor is recommended. The inverter collects the data from Hall sensors with hysteresis-controlled current drives the motor. It is generally easier to evaluate parameters, such as temperature and efficiency, from their geometrical parameters, called direct model, and it necessitates reversing iterative optimization procedures, resulting in geometric parameters when the specifications are known. The dimensions can be computed non-iteratively by employing assumptions that result in a reverse model. It is simple and applicable for machine pre-sizing. Non-linear constitutive mathematical equations governing physical processes in different fields, including electrical, magnetic, mechanical, and thermal.

Eq. 15 presents the expression for electromagnetic power (P_m) from which the torque (T_m) equation can be derived.

$$P_m = T_m(t) \times N(t) = \sum_{i=1}^{Ph} e_i(t) \times i_i(t) \qquad (15)$$

Where i_i denotes the phase current, e_i denotes the phase voltage, Ph denotes the number of phases, and N denotes the speed. In the BLDC motor, there are 3φ with square-wave currents, and the value of electromotive force is equivalent to + E or − E and 0 or intermediate during 120 electrical degrees and 60 electrical degrees, respectively. Therefore, the expression for T_m is presented in Eq. 16 by assuming constant speed and P_m.

$$T_m \times N = 2 \times E \times I \qquad (16)$$

When the rotor displacement equals the pole pitch, a south magnet replaces a north (N) magnet, and the

induction converses. By applying Lenz's law, the expression for an electromotive force is computed, as shown in Eq. 17.

$$E = \frac{z}{4} \times \frac{d\varphi}{d\theta} \times \frac{d\theta}{dt} = \frac{z}{4} \times \frac{2\varphi}{\pi/p} \times N \tag{17}$$

where z denotes the number of supplied wires, and the total number of permanent-magnets is $2 \times p$. When an N-magnet is placed beside the winding, the flux (φ) is calculated using Eq. 18.

$$\varphi = B_e \times A_p \tag{18}$$

Where B_m denotes the maximum air-gap flux density, and A_p denotes the magnetic pole surface area. Equation (17) is based on the assumption of a linear flux fluctuation against the position of the rotor. This results in a constant that is proportional to the value of EMF. From Eqs. 16-18, the torque expression is rewritten as follows.

$$T_m = z \times I \times B_e \times \frac{A_e}{2 \times \pi} \tag{19}$$

$$\text{Total air gap area, } A_e = 2 \times p \times A_p \tag{20}$$

For a BLDC motor with radial flux, the expression for the overall air-gap zone is presented in Eq. 21.

$$A_e = \pi \times D_s \times L_m \tag{21}$$

where L_m denotes the length of the stack and D_s denotes the stator diameter. Now, Eq. 17 is rewritten as follows by considering Eqs. 20-21.

$$E = \frac{z}{4} \times B_e \times D_s \times L_m \times N \tag{22}$$

Figure 6 shows the magnetic portions in the area of the air gap in a simplified manner. The air-gap magnetic induction waveform is depicted in Figure 7 by assuming a constant air-gap width, completely radial air-gap flux, and magnets. Integrating magnetic induction on the area bounded by the winding, determined by the angle in the cross-section, yields the flux across the coil. Note that the EMF remains the same, but the magnets' thickness and the coils' openness can now be adjusted

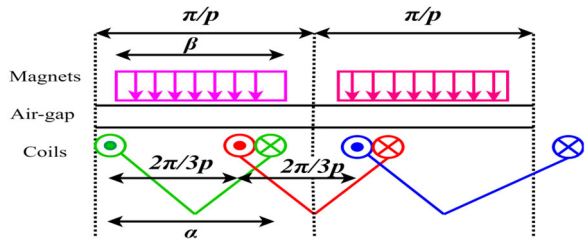

Figure 6. Area of the air-gap, stator coils, and magnets

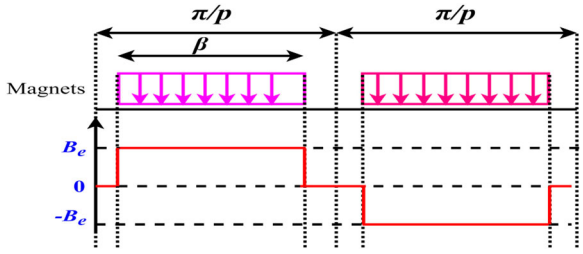

Figure 7. The air-gap flux waveform

to maximize the flux and that the EMF peak is now the maximum.

$$\alpha = \beta = \frac{\pi}{p} \tag{23}$$

The analytical technique revealed the same patterns as the sensitivity of the thickness of the EMF spike. For the variables α and β, nevertheless, the fluctuation is non-linear. Furthermore, it has been demonstrated that a trapezoidal EMF requires an intermediate tooth in the center of each slot and that the thickness of its expanding α_i has a significant impact. For a framework with a ratio of slots to magnets of 3/4, the succeeding relationship is established, presuming that the stator is adequately flat to facilitate the dispersion of flux in the air-gap is as per the assumptions, but not excessively to evade creating a magnetic obstruction between winding.

$$\alpha_i = \frac{\alpha}{5} \tag{24}$$

The geometric dimensions are well-defined in Figure 5. From Figure 5b, the following relations are derived.

$$A_{nc} = h_d \left(2\pi \times \left[\frac{D_s}{2} - e_b\right] - \pi \times h_d - S_e \times (l_d + l_i)\right) \tag{25}$$

$$A_{nc} \times K_r = \frac{3}{2} \times z \times \frac{I}{\delta} \tag{26}$$

$$D_{ext} = D_s + 2 \times (e + h_a + h_{cr}) \tag{27}$$

$$D_{int} = D_s - 2 \times (e_b + h_d + h_{cs}) \tag{28}$$

$$h_c = \frac{e_b}{\cos\frac{\alpha}{2}} - \frac{D_s}{2} \times \left(\frac{1}{\cos\frac{\alpha}{2}} - 1\right) \tag{29}$$

$$h_i = \frac{D_s}{2} \times \left(1 - \cos\frac{\alpha_i}{2}\right) + h_c \times \cos\frac{\alpha_i}{2} \tag{30}$$

where S_e denotes the total number of slots, D_{int} and D_{ext} denote the inner and outer diameters, δ denotes the conductor current density, K_r denotes the factor due to slot filling (< 1), and A_{nc} denotes the total slots sections. The hold height h_c is found to obtain a right angle between the pole shoe and the primary tooth. The width of the middle tooth widening h_i is found so that the elevation of a middle tooth is equal to one of a primary tooth h_c, and there is a right angle among the middle tooth and pole shoe. Eq. 31 presents the mean

radius of the winding terminal, and Eq. 32 shows the average span of a half turn.

$$R_t \approx \frac{l_d - l_i}{4} + \left[\frac{D_s}{2} - e_b - \frac{h_d}{2}\right] \times \frac{\pi}{2 \times S_e} \tag{31}$$

$$L_d \approx \frac{L_m}{k_f} + \pi \times R_t \tag{32}$$

The total motor axial length L_T is calculated using Eq. 33, and various active part weights are represented in Eqs. 34-39.

$$L_T \approx \frac{L_m}{k_f} + 2 \times \left[\frac{D_s}{2} - e_b - \frac{h_d}{2}\right] \times \frac{\pi}{S_e} - \frac{l_i}{2} \tag{33}$$

$$M_a = d_a \times p \times \beta \times h_a \times \left[h_a + \left(2 \times \frac{D_s}{2} + e\right)\right] \times L_m \times r_{rs} \tag{34}$$

$$M_{cr} = d_{cr} \times \pi \times h_{cr} \times \left[h_{cr} + \left(2 \times \frac{D_s}{2} + e + h_a\right)\right] \times L_m \times r_{rs} \tag{35}$$

$$M_{cs} = d_t \times \pi \times h_{cs} \times \left[2 \times \left(\frac{D_s}{2} - e_b - h_d\right) - h_{cs}\right] \times L_m \tag{36}$$

$$M_{ds} = d_t \times S_e \times \left[(l_i + l_d) \times h_d + \left(\alpha \frac{e_b + h_c}{2} + \alpha_i \frac{h_i + h_c}{2}\right) \times \frac{D_s}{2}\right] \times L_m \tag{37}$$

$$M_{cu} = d_{cu} \times \frac{3}{2} \times z \times \frac{I}{\delta} \times L_{ds} \tag{38}$$

The total mass of all the active parts M_{Tt} is the sum of individual weights of all parts.

$$M_{Tt} = M_a + M_{cr} + M_{cs} + M_{ds} + M_{cu} \tag{39}$$

where $k_f < 1$ denotes the metal sheets bulk factor, M_{cu}, M_{ds}, M_{cs}, M_{cr}, and M_a denote the mass of the copper, teeth of the stator, stator yoke, rotor yoke, and magnets, respectively, d_{cu}, d_t, d_{cr}, and d_a denote the density of copper, stator teeth, rotor yoke, and magnets, and $1.2 \geq r_{rs} \geq 1/k_f$ denotes the ratio of the rotor length on one of the stators.

The average flux density values in stator back iron, rotor yoke, magnets, and teeth are B_{cs}, B_{cr}, B_a, and B_d, respectively. The flux conservation between the pole shoe and the primary tooth and the pole shoe and the middle tooth is given in Eq. 40 and Eq. 41, respectively.

$$B_d \times l_d = B_e \times \alpha \times \frac{D_s}{2} \tag{40}$$

$$B_d \times l_i = B_e \times \alpha_i \times \frac{D_s}{2} \tag{41}$$

The pole shoe width is measured to allow the flux absorbed by the portion of the pole shoe traveling through the primary tooth to flow through, resulting in flux conservation, as given in Eq. 42.

$$B_d \times \left[e_b - \frac{D_s}{2}\left(1 - \cos\left(\sin^{-1}\frac{l_d}{D_s}\right)\right)\right] = B_e \times \left[\frac{\alpha}{2} - \sin^{-1}\frac{l_d}{D_s}\right] \times \frac{D_s}{2} \tag{42}$$

Because 50% of the flux traversing the magnet departs one side of the rotor yoke while the other 50% departs the other, flux conservation between the rotor yoke and the magnet, as given in Eq. 43.

$$\frac{1}{2} \times B_a \times \beta \left(\frac{D_s}{2} + e\right) = B_{cr} \times h_{cr} \tag{43}$$

Similarly, flux conservation between the stator yoke and the primary teeth, as given in Eq. 44.

$$\frac{1}{2} \times B_a \times l_d = B_{cs} \times h_{cs} \tag{44}$$

Eventually, the flux between a magnet and the primary tooth is also conserved, as presented in Eq. 45.

$$\left(B_a \times \beta \left(\frac{D_s}{2} + e\right) \times r_{rs}\right) \times k_{fi} = \frac{D_s}{2} \times \alpha \times B_e \tag{45}$$

where k_{fi} denotes the leakage coefficient calculated using FEM modeling and is always less than 1. It expresses how the leakage flux across magnets crosses a portion of the air-gap and the rotor but does not enter the stator.

By applying Ampere's law at no load condition, a relation between the magnet's geometric dimensions, magnetic circuit, the MMF, and the flux densities in various portions.

$$\frac{B_{cr}}{\mu_{cr}(B_{cr})} \times \left[\frac{h_{cr}}{2} + \frac{\pi}{2p} \times \left(\frac{D_s}{2} + e + h_a + \frac{h_{cr}}{2}\right)\right] + \left[\frac{B_a - B_r(1 + \alpha_a \times T_a)}{\mu_a}\right] \times h_a + B_e \times e + \frac{B_d}{\mu_t(B_d)}(e_b + h_d) + \frac{B_{cs}}{\mu_t(B_{cs})} \times \left[\frac{h_{cs}}{2} + \frac{\pi}{S_e}\left(\frac{D_s}{2} - e_b - h_d - \frac{h_{cs}}{2}\right)\right] = 0 \tag{46}$$

where μ_{cr} and μ_t denote the relative permeabilities of the rotor yoke and sheet metal, $\alpha_a < 0$, B_r, and μ_a denote the magnet temperature, thermal coefficient, the remnant flux density of the magnets, and relative permeability, respectively.

While assuming that the permeabilities of rotor yoke and sheet metal are greater than vacuum, and thus that the magneto-motive force used in the circuit is

insignificant compared to that expended in the air-gap. Therefore, Eq. 46 is simplified as seen in Eq. 47

$$\left(\frac{B_a - B_r(1+\alpha_a \times T_a)}{\mu_a}\right) \times h_a + B_e \times e = 0 \quad (47)$$

Sufficient current flows in the coils whenever the motor is energized. Magnetization in the magnets can exceed the threshold for an excessively large current, resulting in demagnetization. As a result of the low induction, the magnetic permeability is large, and the MMF expended in the magnetic circuit can be ignored. The phase current reaches its maximum tolerable values at a critical magnetic induction in the magnets B_c.

$$\left(\frac{B_c - B_r(1+\alpha_a \times T_a)}{\mu_a \times \mu_o}\right) \times h_a + \frac{z \times I_{max}}{4p} + \frac{B_c}{\mu_o} \times \frac{\beta}{\alpha} \times \left(1 + \frac{2e}{D_s}\right) \times r_{rs} \times k_{fi} \times e = 0 \quad (48)$$

The electrical loss can be found by defining the phase resistance and is given in Eq. 49.

$$R_{ph} = \rho \times (1 + \alpha_{cu} \times T_{cu}) \times \frac{z}{2} \times L_{ds} \times \frac{\delta}{I} \quad (49)$$

where T_{cu} denotes the coil temperature, $\alpha_{cu} > 0$ denotes thermal coefficient, and ρ denote the copper resistivity. The electrical copper loss is calculated using Eq. 50, and the fundamental frequency is given in Eq. 51.

$$P_{cu} = 2 \times R_{ph} \times I^2 \quad (50)$$

$$f = \frac{pN}{2\pi} \quad (51)$$

The core or iron loss is calculated using Eq. 52.

$$P_{ir} = q_t \times \left(\frac{f}{f_t}\right)^{1.5} \times \left[M_{cs} \times \left(\frac{B_{cs}}{B_t}\right)^2 + M_{ds} \times \left(\frac{B_d}{B_t}\right)^2\right] \quad (52)$$

where q_t denote the specific loss for flux density B_t and a frequency f_t. Therefore, the motor efficiency is calculated using Eq. 40, in which P_{mec} denotes the mechanical losses.

$$\eta = \frac{T_m \times N - P_{mec}}{T_m \times N + P_{cu} + P_{ir}} \quad (53)$$

The thermal model is straightforward. Conversely, it is assumed that thermal conduction resistances are much lesser than thermal convection resistances. As a result, the temperature of all stator parts in touches, such as the stator yoke, teeth, and coils, is the same. However, even if the rotor and stator are not in touch due to rolling, large areas are in touch with air space. The confined air is represented as a substance with low thermal conductivity under these circumstances. Despite this, it is low because the thermal resistance between the stator and the rotor is equivalent to the depth of air separated by the thermal conduction and the face-to-face surfaces. As a result, the temperature gradient within the motor is minimal, and all active components are assumed to be at the same temperature.

$$T_a = T_{cu} \quad (54)$$

In the theory of an enclosed motor, the exterior surface where convection occurs is equal to Eq. 55.

$$A_{ext} = \frac{\pi}{2} \times {D_{ext}}^2 + \pi \times D_{ext} \times L_T \quad (55)$$

Therefore, the overall temperature of the motor is calculated using Eq. 56.

$$T_{cu} = T_{ext} + \frac{P_m + P_{cu} + P_{ir}}{h \times A_{ext}} \quad (56)$$

where T_{ext} denotes the ambient temperature and h denotes the convection coefficient in air.

3.3. Problem formulation

In [53], a BLDC motor benchmark was presented, and the basic code in MATLAB is publicly available to compute the fitness functions. The studied problem has a MATLAB model that can be used for research recommendations in [52], where 78 non-linear equations are introduced with five optimization variables and six inequality constraints in the single-objective problem or with five constraints in the multi-objective problem. In this paper, two objective functions, such as minimization of motor mass (f_1) and maximization of motor efficiency (f_2) with five design/optimization variables, such as back iron B_{cs}, magnetic induction both in the teeth B_d, the current density in the conductors δ, magnetic induction in the air-gap B_e, and bore stator diameter D_s, and six inequality constraints, such as temperature T_a, magnetics maximum current I_m, inner diameter D_{in}, external diameter D_{ex}, total mass M_{Tt}, and the determinant (*Discr*) used for the slot height calculation based on five optimization variables. This research focuses on the multi-objective situation, thus adopting electromagnetic fields with the new concept of MOCHIO. The optimization aims to achieve minimum total mass and maximum efficiency within the acceptable range of optimization parameters under the constraint variables' restriction, as shown in Table 1.

In this paper, a benchmark (five constraints with pole pair constant) with two objective functions is considered. The thermal model in this method is very straightforward since the model is for pre-sizing considerations only. It is also presumed that the thermal conduction resistance is smaller than the thermal convection resistance and that the temperature gradient within the motor might be ignored because of the motor geometry. So, it is presumed that the temperature of the

Table 1. BLDC wheel motor's design variables and its constraints

Type	Variables	Definition	Upper limit	Lower limit
Design variables	D_s (mm)	Diameter of the stator	300	150
	B_{cs} (T)	Mean flux density in stator back iron	1.6	0.6
	B_e (T)	Airgap flux density	0.76	0.5
	B_d (T)	Mean flux density in teeth	1.8	0.9
	δ (A/mm^2)	Winding current density	5	2
Inequality Constraints	T_a (^oC)	Motor temperature	$\leq$ 120	
	I_m (A)	Maximum phase current	$\geq$ 125	
	D_{in} (mm)	Internal diameter	$\geq$ 76	
	D_{ex} (mm)	External diameter	$\leq$ 340	
	M_{Tt} (Kg)	Total motor mass	$\leq$ 15	
	Discr	Determinant for the slot height calculation	Discr (δ, D_s, B_e, B_d) $\geq$ 0	
Objective variables	M_{Tt} (Kg)	Total motor mass	-	
	η (%)	Motor efficiency	-	

copper exists in the active parts. Therefore, the biggest change in heat is between the surrounding temperature and the motor. Generally, this problem is a multidisciplinary optimization challenge. Two objective functions are considered, such as the motor mass (f_1) and motor efficiency (f_2). It is possible to write the MOOPs as follows.

$$
\begin{aligned}
&\textit{Minimize, } [f_1, f_2] = [M_{Tt}, 1-\eta] = f(D_s, B_{cs}, B_e, B_d, \delta)\\
&\qquad\qquad D_{in} \geq 76mm\\
&\qquad\qquad D_{ex} \geq 340mm\\
&\text{Subjected to constraints :}\quad T_a \leq 120^oC\\
&\qquad\qquad Im \geq 125A\\
&\qquad\qquad Discr(\delta, D_s, B_e, B_d) \geq 0\\
&\qquad\qquad 150mm < D_s < 300mm\\
&\qquad\qquad 0.6T < B_{cs} < 1.6T\\
&\text{Upper and Lower Bounds:}\quad 0.5T < B_e < 0.76T\\
&\qquad\qquad 0.9T < B_d < 1.8T\\
&\qquad\qquad 2\,A/mm2 < \delta < 5\,A/mm^2
\end{aligned}
$$

4. Simulation results and discussions

This section discusses the simulation analysis of the proposed MOCHIO algorithm and performance comparison between the other competitive algorithms, such as MOPSO, MOBA, MOGWO, MOMFO, and MOWOA. The BLDC wheel motor design problem has two objective functions and six inequality constraints. Therefore, it is fair enough to test the MOCHIO algorithm on several standard benchmark test functions. Therefore, the proposed MOCHIO algorithm is validated on standard benchmark test functions, such as CONSTR, SRN, BNH, TNK, ZDT1-ZDT4, and ZDT6. Later, the proposed algorithm and other competitive algorithms are applied to the BLDC motor design problem, and the performance is also analyzed. The control parameters of various algorithms are presented in Table 2. The population size and maximum iterations for all selected algorithms are 100 and 1000, respectively. For a fair comparison, each algorithm is executed 30 times individually.

Table 2. Control parameters of various algorithms

Algorithm	Parameters	Definition	Range
MOPSO	nArc	Size of Archive	200
	C_1 and C_2	Constants	1.5
	w	Inertia weight	0.75
MOBA	nArc	Size of Archive	200
	α and γ	Constants	0.9
MOGWO	nArc	Size of Archive	200
	a	Linear decrease	2
MOMFO	nArc	Size of Archive	200
MOWOA	nArc	Size of Archive	200
	a	Linear decrease	2
MOCHIO	nArc	Size of Archive	200
	BR_r	Basic reproduction rate	0.5
	Max_{Age}	Maximum infected cases age	100

4.1. Performance metrics

The performance metrics are very important for a fair comparison of the MOCHIO with other state-of-the-art multi-objective algorithms. Therefore, the performance indicators, such as Delta, Epsilon, Spread, Generational Distance (GD), Inverted Generational Distance (IGD), and Spacing, are calculated and discussed in this paper [33,35,54]. The performance metrics are calculated as follows.

$$\text{Spacing} \triangleq \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d}-d_i)^2} \tag{58}$$

$$\text{Spread} = \sqrt{\sum_{i=1}^{o}\max(d(a_i,b_i))} \tag{59}$$

$$\text{Generational Distance (GD)} = \frac{\sqrt{\sum_{i=1}^{no} d_i^2}}{n} \tag{60}$$

$$\text{Inverted Generational Distance (IGD)} = \frac{\sqrt{\sum_{i=1}^{nt}(d_i')^2}}{n} \tag{61}$$

$$\text{Diversity Metric(DM)} = \left|\frac{d_l + d_m + \int_{i=1}^{nt}|d_i - d|}{d_l + d_m - (n-1)d}\right| \tag{62}$$

where d_m and d_l are Euclidean distances between attained Pareto front and true Pareto front, and d denotes the number of attained Pareto optimal solutions and averaged all solutions distance. nt denotes the number of true Pareto-optimal solutions, no denotes the number of true Pareto set (PS), o denotes the number of objectives, n denotes the number of attained PS, d_i is Euclidean distance between attained Pareto front and true Pareto front in each point and is equal to $\min_j(|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})|)$ for $i, j = 1,2,\ldots,n.$, $\bar{d}$ denotes the average of d_i, d_i and d_i' specifies the Euclidean distance, and b_i and a_i denotes

minimum and maximum values in the i^{th} objective. For validation, the performance of the MOCHIO algorithm and other competitive algorithms, such as MOPSO, MOBA, MOGWO, MOMFO, and MOWOA in terms of faster convergence metrics, such as Spread, GD, combined diversity-spread, such as Epsilon and Spacing, and combined uniformity-convergence-spread, such as DM, IGD, are discussed.

4.2. Classical constrained benchmark functions

First of all, the performance of the suggested MOCHIO algorithm is confirmed using the well-known constraint multi-objective benchmark problems, such as CONSTR, SRN, BNH, and TNK, the benchmark optimization test functions presented in [55], before applying to the design and optimization problem of a BLDC motor. All four problems have two objective functions and four constraints. The objective functions and their constraints are presented in Eqs. 63-66.

$$\text{CONSTR}: \begin{cases} Minimize\, f_1(x,y) = x \\ Minimize\, f_2(x,y) = (1+y)/x \\ g(1) = 0.1 \le x \le 1 \\ g(2) = 0 \le y \le 5 \\ g(3) = 0 \ge 6 - y - 9x \\ g(4) = 0 \ge 1 + y - 9x \end{cases} \tag{63}$$

$$\text{SRN}: \begin{cases} Minimize\, f_1(x,y) = (x-2)^2 + (y-1)^2 + 2 \\ Minimize\, f_2(x,y) = 9x - (y-1)^2 \\ g(1) = -20 \le x \\ g(2) = y \le 20 \\ g(3) = 0 \ge x^2 + y^2 - 225 \\ g(4) = 0 \ge x - 3y + 10 \end{cases} \tag{64}$$

$$\text{BNH}: \begin{cases} Minimize\, f_1(x,y) = 4x^2 + 4y^2 \\ Minimize\, f_2(x,y) = (x-5)^2 + (y-5)^2 \\ g(1) = 0 \le x \le 5 \\ g(2) = 0 \le y \le 3 \\ g(3) = 0 \ge (x-5)^2 + (y)^2 - 25 \\ g(4) = 0 \ge -(x-8)^2 + (y+3)^2 + 7.7 \end{cases} \tag{65}$$

$$\text{TNK}: \begin{cases} Minimize\, f_1(x,y) = x \\ Minimize\, f_2(x,y) = y \\ g(1) = 0 \le x \\ g(2) = y \le \pi \\ g(3) = 0 \ge -x^2 - y^2 + 1 + 0.1 * \\ \quad cos\left(16 arctan \frac{x}{y}\right) \\ g(4) = \frac{1}{2} \ge \left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2 \end{cases} \tag{66}$$

The above four multi-objective benchmark problems are optimized using the proposed MOCHIO algorithm to test the ability of the proposed MOCHIO in handling the multi-objective problem with inequality constraints. The proposed MOCHIO algorithm's performance is compared with the other well-known and recent multi-objective algorithms, such as MOPSO, MOBA, MOGWO, MOMFO, and MOWOA for the classical constrained multi-objective benchmark test functions. The obtained Pareto front (PF) for all four test functions, with the optimal solutions, are plotted and illustrated in Figure 8.

Tables 3–6 lists all the performance metrics values by all selected algorithms on various benchmark test functions, such as CONSTR, SRN, BNH, and TNK. The performance indicators enumerate the convergence and the coverage of Pareto optimal solutions assessed by various algorithms. The bold letters in all the tables indicate the best results. As discussed earlier, the results are taken after simulating 30 individual runs. From Tables 3–6, it is observed that the proposed MOCHIO is outperforming the other algorithms in terms of all performance indicators, followed by MOGWO, MOMFO, MOPSO, MOWOA, and MOBA.

4.3. ZDT benchmark test functions

Secondly, the proposed algorithm and competitors are validated on unconstrained ZDT benchmark functions (ZDT1-ZD4 and ZDT6). All ZDT functions have 2-objectives with 30-dimensions (ZDT1-ZDT3) and 10-dimensions (ZDT4 and ZDT6). The ZDT1-ZDT2 functions have unimodal characteristics, ZDT3-ZDT4 functions have unimodal/multimodal characteristics, and ZDT6 has multimodal characteristics [35,54]. The unimodal characteristics are suitable for assessing the exploitation stage, while the multimodal characteristics are suitable for assessing the exploring stage. The MOCHIO algorithm's exploitation and exploration capabilities are analyzed and compared to other state-of-the-art algorithms in this sense. The results obtained by all algorithms on ZDT benchmark functions are listed in Tables 7–12. It is obvious that the MOCHIO produces the best results for ZDT benchmark test functions. In order to assess the performance of the MOCHIO algorithm, the values of various performance metrics -are obtained by all selected algorithms and presented in Tables 7–12. The MOCHIO algorithm shows the faster convergence (GD and Runtime (RT)), the best uniform convergence (DM, IGD, and Spacing), and the combined diversity (Spread and Epsilon). The PF obtained by all algorithms is plotted and illustrated in Figure 9. The findings demonstrate that the Pareto optimal solution on the computed PF is distributed evenly for all problems.

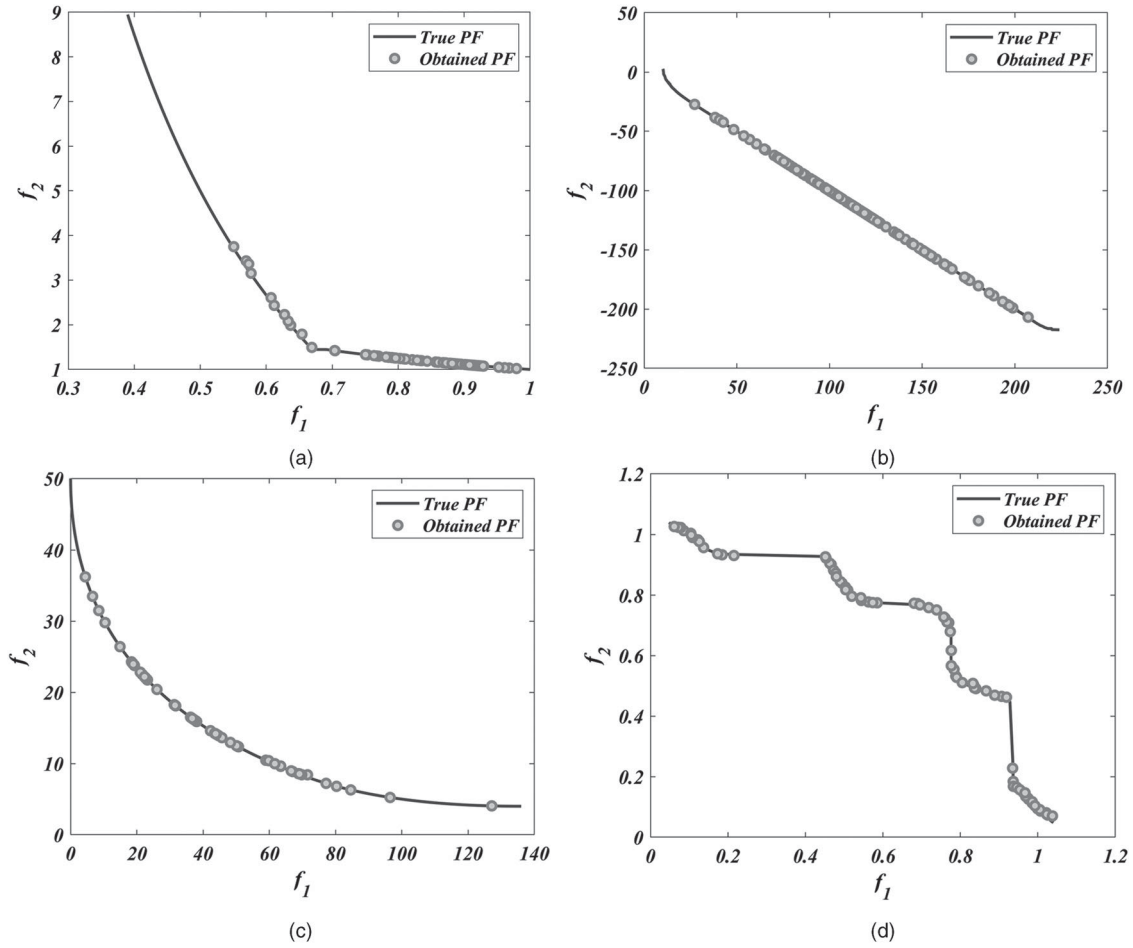

Figure 8. Pareto fronts obtained by the proposed MOCHIO algorithm; (a) CONSTR, (b) SRN, (c) BNH, (d) TNK

Table 3. Performance comparison of various algorithms for CONSTR

Algorithm	DM	Epsilon	Spread	GD	IGD	Spacing
MOPSO	1.3528	0.087284	1.1788	0.0002941	0.0034098	0.07439
MOBA	1.5967	0.034578	1.2368	0.00019807	0.00096741	0.11333
MOGWO	1.3773	0.085223	1.1979	0.00013277	0.0035998	0.057799
MOMFO	1.4943	0.053943	1.2042	0.00019598	0.0017192	0.08068
MOWOA	1.4858	0.044312	1.1834	0.00025891	0.0011774	0.094263
MOCHIO	1.1252	0.050385	1.2971	0.00012991	0.0009199	0.067902

Table 4. Performance comparison of various algorithms for SRN

Algorithm	DM	Epsilon	Spread	GD	IGD	Spacing
MOPSO	1.088	9.3888	1.0863	0.015203	0.00030618	1.7198
MOBA	1.0961	6.2194	1.0945	0.011729	0.00022835	1.3137
MOGWO	1.1367	6.8601	1.1344	0.024488	0.00033117	2.2011
MOMFO	1.1551	11.3608	1.1524	0.019977	0.00063907	3.2687
MOWOA	1.1758	11.3532	1.1732	0.013941	0.00067671	3.3165
MOCHIO	1.0216	4.9672	1.0191	0.011134	0.00013241	1.5873

Table 5. Performance comparison of various algorithms for BNH

Algorithm	DM	Epsilon	Spread	GD	IGD	Spacing
MOPSO	1.2984	4.3221	1.2456	0.032969	0.0058033	3.9771
MOBA	1.3332	5.9898	1.3131	0.024976	0.0074386	3.6298
MOGWO	1.2755	**2.9590**	1.2769	0.02828	0.0027545	2.6517
MOMFO	1.3222	4.7917	1.3033	0.038368	0.00327	2.8668
MOWOA	1.3628	5.1142	1.3420	0.023671	0.0039312	2.6719
MOCHIO	**1.2032**	3.4975	**1.1891**	**0.022469**	**0.0027112**	**1.6717**

Tables 7–12 presents the Mean and STD values obtained by all optimizers for ZDT problems. Table 7 gives the DM values to verify the diversity maintenance capability. The MOCHIO delivers the best outcomes in four of the five problems. Table 8 provides the Spread values to show how the non-dominated solutions are

Table 6. Performance comparison of various algorithms for TNK

Algorithm	DM	Epsilon	Spread	GD	IGD	Spacing
MOPSO	NaN	0.027803	NaN	NaN	0.0012339	NaN
MOBA	NaN	0.025042	NaN	NaN	0.0014032	NaN
MOGWO	NaN	0.024603	NaN	NaN	0.0017123	NaN
MOMFO	NaN	0.031469	NaN	NaN	0.001569	NaN
MOWOA	NaN	0.050145	NaN	NaN	0.0015854	NaN
MOCHIO	NaN	**0.01806**	NaN	NaN	**0.0010303**	NaN

NaN indicates that the values are not available

Table 7. Mean and standard deviation (STD) values of DM obtained by all algorithms

Problem	MOMFO	MOBA	MOPSO	MOMFO	MOWOA	MOCHIO
ZDT1	8.1207e-1 (1.43e-2)	8.3753e-1 (1.08e-2)	7.0757e-1 (1.00e-2)	4.2717e-1 (7.41e-2)	9.0220e-1 (0.00e+0)	**9.0220e-1 (0.00e+0)**
ZDT2	8.3740e-1 (1.35e-2)	8.6610e-1 (4.51e-3)	2.5447e-1 (4.12e-1)	4.9800e-1 (3.50e-2)	8.5893e-1 (1.32e-2)	**8.8777e-1 (5.25e-3)**
ZDT3	9.7425e-1 (8.92e-3)	9.3815e-1 (1.05e-2)	9.2985e-1 (4.86e-2)	2.6555e-1 (1.78e-2)	9.1087e-1 (1.98e-2)	**9.7950e-1 (1.79e-2)**
ZDT4	8.2677e-1 (1.32e-2)	8.9477e-1 (9.55e-3)	4.7650e-1 (3.98e-1)	3.6320e-1 (3.56e-2)	**9.0220e-1 (0.00e+0)**	8.2897e-1 (1.71e-2)
ZDT6	8.4390e-1 (6.80e-3)	8.5867e-1 (1.74e-2)	6.9067e-1 (3.34e-2)	8.1417e-1 (1.96e-2)	8.7093e-1 (2.89e-3)	**8.7093e-1 (2.89e-3)**

Table 8. Mean and STD values of Spread obtained by all algorithms

Problem	MOMFO	MOBA	MOPSO	MOMFO	MOWOA	MOCHIO
ZDT1	3.7956e-1 (4.03e-2)	2.7413e-1 (1.32e-3)	8.5086e-1 (4.98e-2)	1.3645e+0 (1.38e-1)	2.7364e-1 (4.28e-4)	2.5600e-1 (2.55e-2)
ZDT2	3.7130e-1 (7.97e-2)	1.4764e-1 (1.69e-3)	9.6816e-1 (5.51e-2)	1.1861e+0 (3.24e-2)	2.5354e-1 (2.15e-2)	1.4276e-1 (9.37e-4)
ZDT3	4.3111e-1 (5.95e-2)	5.6005e-1 (5.01e-3)	9.0538e-1 (6.82e-2)	1.7667e+0 (6.47e-3)	5.3771e-1 (1.39e-2)	2.7635e-1 (1.32e-2)
ZDT4	4.0346e-1 (3.14e-2)	2.7749e-1 (6.05e-3)	9.8778e-1 (1.27e-2)	1.4173e+0 (6.65e-2)	2.7260e-1 (1.99e-3)	2.5194e-1 (2.10e-2)
ZDT6	4.3435e-1 (4.06e-2)	1.4109e-1 (4.96e-3)	9.5171e-1 (8.29e-2)	4.2545e-1 (3.81e-2)	1.3753e-1 (6.49e-4)	2.7427e-1 (3.83e-2)

Table 9. Mean and STD values of GD obtained by all algorithms

Problem	MOMFO	MOBA	MOPSO	MOMFO	MOWOA	MOCHIO
ZDT1	1.5741e-4 (4.94e-5)	3.7630e-5 (3.64e-5)	5.4570e-2 (1.73e-2)	1.9203e-5 (2.11e-5)	1.0415e-4 (1.33e-5)	1.6852e-5 (2.40e-6)
ZDT2	1.2738e-4 (5.48e-5)	1.5410e-4 (6.64e-5)	1.1095e-1 (3.82e-2)	1.0663e-4 (2.62e-5)	6.3508e-5 (3.90e-5)	6.2060e-6 (2.30e-6)
ZDT3	7.5873e-5 (2.54e-5)	2.0217e-4 (3.88e-5)	9.0643e-2 (2.74e-2)	8.0794e-5 (1.19e-5)	7.9505e-5 (8.41e-6)	9.4683e-6 (4.81e-6)
ZDT4	**3.1700e-5 (9.70e-6)**	3.1927e-4 (1.99e-4)	1.6345e+0 (9.20e-1)	7.0693e-5 (2.95e-5)	1.4941e-4 (1.31e-4)	3.2860e-5 (6.05e-6)
ZDT6	3.7838e-6 (4.09e-7)	1.7000e-4 (5.32e-5)	1.5788e-2 (1.44e-2)	3.6256e-6 (1.01e-7)	6.4800e-5 (2.16e-5)	3.5448e-6 (1.05e-7)

Table 10. Mean and STD values of IGD obtained by all algorithms

Problem	MOMFO	MOBA	MOPSO	MOMFO	MOWOA	MOCHIO
ZDT1	4.7829e-3 (3.05e-4)	3.9776e-3 (1.57e-5)	5.3280e-1 (1.50e-1)	2.1872e-2 (3.77e-3)	3.9619e-3 (2.05e-6)	4.4153e-3 (1.10e-4)
ZDT2	4.8755e-3 (4.87e-4)	4.0097e-3 (8.06e-5)	1.4105e+0 (5.83e-1)	1.8745e-2 (2.83e-3)	4.5271e-3 (1.41e-4)	3.8586e-3 (4.11e-5)
ZDT3	5.3882e-3 (1.98e-4)	1.1984e-2 (1.99e-4)	6.3830e-1 (1.68e-1)	1.0086e-1 (6.93e-4)	1.1425e-2 (1.07e-4)	5.1009e-3 (1.32e-4)
ZDT4	4.4649e-3 (4.13e-5)	5.3851e-3 (1.06e-3)	1.5822e+1 (8.74e+0)	3.1551e-2 (2.03e-3)	4.4705e-3 (6.66e-4)	4.4004e-3 (2.77e-4)
ZDT6	3.7427e-3 (5.64e-5)	3.5418e-3 (2.25e-4)	6.8336e-2 (1.07e-1)	4.2217e-3 (4.45e-4)	3.4182e-3 (7.88e-5)	3.1982e-3 (5.44e-5)

Table 11. Mean and STD values of Spacing obtained by all algorithms

Problem	MOMFO	MOBA	MOPSO	MOMFO	MOWOA	MOCHIO
ZDT1	6.8348e-3 (7.57e-4)	4.9625e-3 (2.91e-5)	1.2392e-2 (2.83e-3)	3.5684e-2 (7.43e-3)	5.6978e-3 (6.61e-4)	4.9043e-3 (4.47e-6)
ZDT2	6.4892e-3 (4.05e-4)	4.4510e-3 (9.16e-5)	3.0291e-3 (5.25e-3)	3.2567e-2 (2.59e-3)	4.2702e-3 (4.57e-5)	5.6507e-3 (5.35e-4)
ZDT3	8.1461e-3 (9.30e-4)	1.7475e-2 (3.35e-4)	1.7114e-2 (3.94e-3)	1.0760e-1 (3.72e-4)	1.6249e-2 (2.24e-4)	5.6939e-3 (6.10e-5)
ZDT4	7.0331e-3 (5.48e-4)	4.9546e-3 (7.47e-5)	2.0275e-2 (1.80e-2)	4.3742e-2 (5.54e-3)	5.5806e-3 (5.31e-4)	4.7864e-3 (1.39e-4)
ZDT6	6.1500e-3 (4.80e-4)	3.0309e-3 (1.36e-4)	1.4177e-2 (3.70e-3)	6.7458e-3 (5.75e-4)	4.7176e-3 (3.82e-4)	2.9289e-3 (2.40e-5)

Table 12. Mean and STD values of RT obtained by all algorithms

Problem	NSGAII	MOEAD	MOPSO	NMPSO	CMOEAD	CAMOEA
ZDT1	1.3297e+1 (6.48e-2)	1.9241e+0 (3.94e-2)	3.4788e+1 (3.91e-1)	1.6480e+1 (1.88e-1)	2.7731e+0 (1.03e-1)	1.5120e+0 (8.94e-2)
ZDT2	1.3319e+1 (6.44e-2)	1.6869e+0 (1.08e-1)	3.4207e+1 (1.81e-1)	1.6169e+1 (7.97e-2)	2.6354e+0 (1.30e-2)	1.4229e+0 (3.17e-3)
ZDT3	1.4153e+1 (8.70e-1)	1.8093e+0 (1.27e-2)	2.3751e+1 (3.40e-1)	1.6531e+1 (3.48e-1)	2.7009e+0 (4.19e-2)	1.4271e+0 (7.66e-3)
ZDT4	1.3355e+1 (9.97e-2)	1.8043e+0 (1.45e-1)	1.8406e+1 (1.56e+0)	1.6481e+1 (3.31e-1)	2.4223e+0 (7.54e-2)	1.3397e+0 (6.09e-3)
ZDT6	1.3315e+1 (1.01e-1)	1.9021e+0 (1.76e-1)	3.3866e+1 (9.41e-1)	1.5852e+1 (2.03e-3)	2.4317e+0 (1.13e-2)	1.4011e+0 (1.97e-2)

well-distributed. Table 9 shows that the MOCHIO algorithm acquired the best GD values for ZDT problems, demonstrating the MOEO's superior convergence rate than other approaches. Table 10 provides the IGD values to measure the quality of estimations to the PF acquired by the proposed algorithm, and it shows that the MOCHIO has the best outcomes for four out of the five problems. Table 11 includes the Spacing values to indicate the non-dominated solutions spacing accuracy. For all problems, the proposed algorithm has the optimal parameters, implying that the spacing accuracy of the non-dominated solutions provided by the proposed algorithm is superior to all algorithms. The RT values are listed in Table 12 to analyze the computational cost. The MOCHIO achieves the best results in all ZDT problems, demonstrating that MOCHIO

has lower computational complexity than all other optimizers.

4.4. BLDC motor design optimization problem

After test cases analysis, the BLDC motor design optimization problem is optimized using six selected multi-objective algorithms, including the proposed MOCHIO. All the selected algorithms are executed in 30 trials. The control parameters of all algorithms are selected based on Table 2. Since the BLDC wheel motor design problem has six constraints, the proposed algorithm is employed with a constraint handling mechanism. To transform the constraint problem into an unconstraint problem, this study uses a static penalty method. If at all constraint is breached, a significant penalty P_i is imposed on the fitness function. The constraint handling approach is modeled and represented in Eq. 67.

$$f_i(X) = f_i(X) + \sum_{j=1}^{x} P_j \max\{g_j(X),0\} + \sum_{j=x+1}^{y} P_j \max\{|h_j(X)| - \delta,0\} \quad (67)$$

where $f_i(X)$ denotes the objective function ($i = 1, 2, \ldots, N$), N denotes the number of objective functions, X denotes decision variables ($\{x_1,x_2, \ldots, x_m\}$), $g_j(X) \leqslant 0$ denotes the inequality constraints ($j = 1,2, \ldots, x$), $h_i(X) = 0$ denotes the equality constraints ($j = x + 1, \ldots, y$), and δ denotes the tolerance of equality constraint.

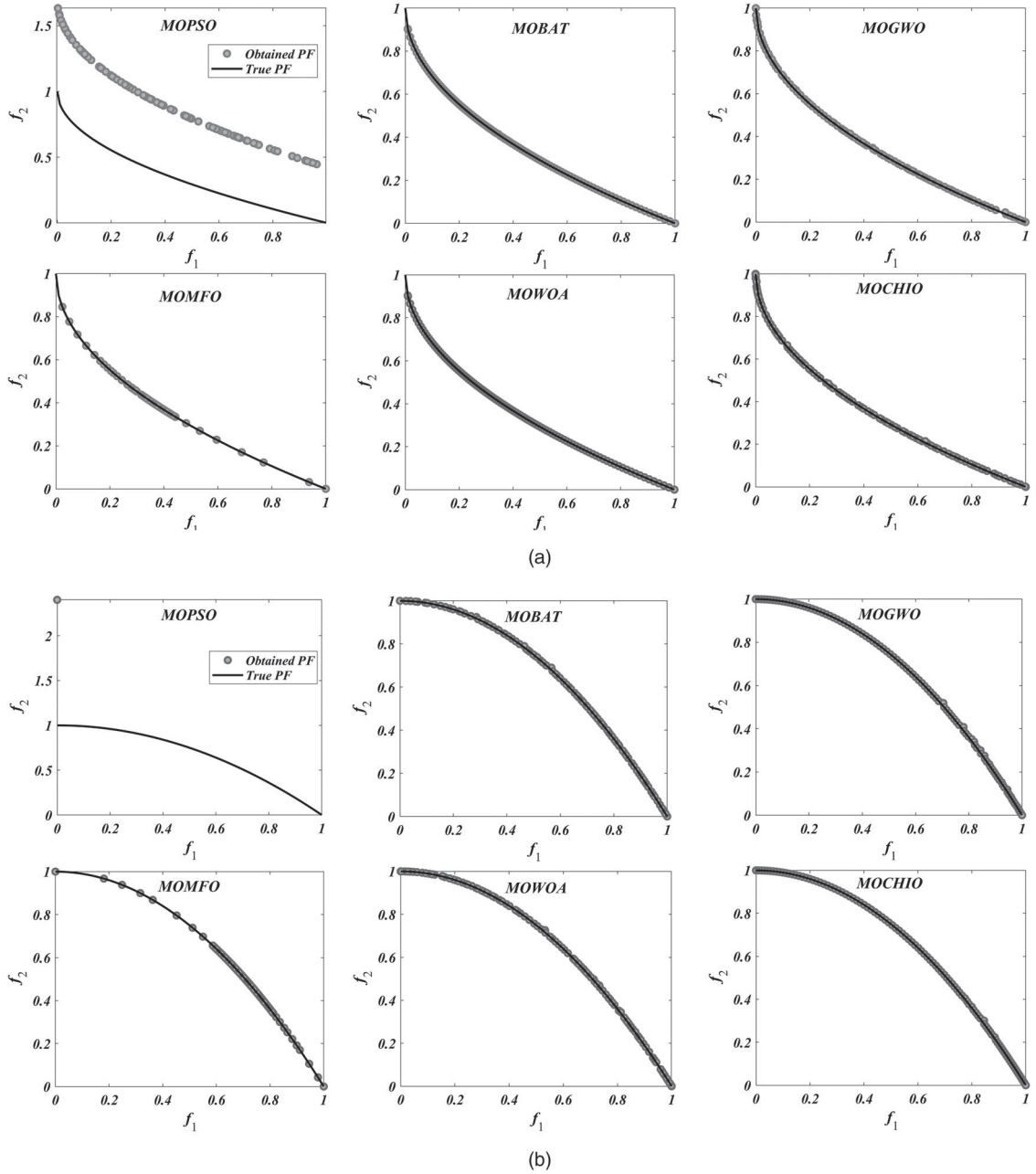

Figure 9. PF obtained by all algorithms for ZDT problems

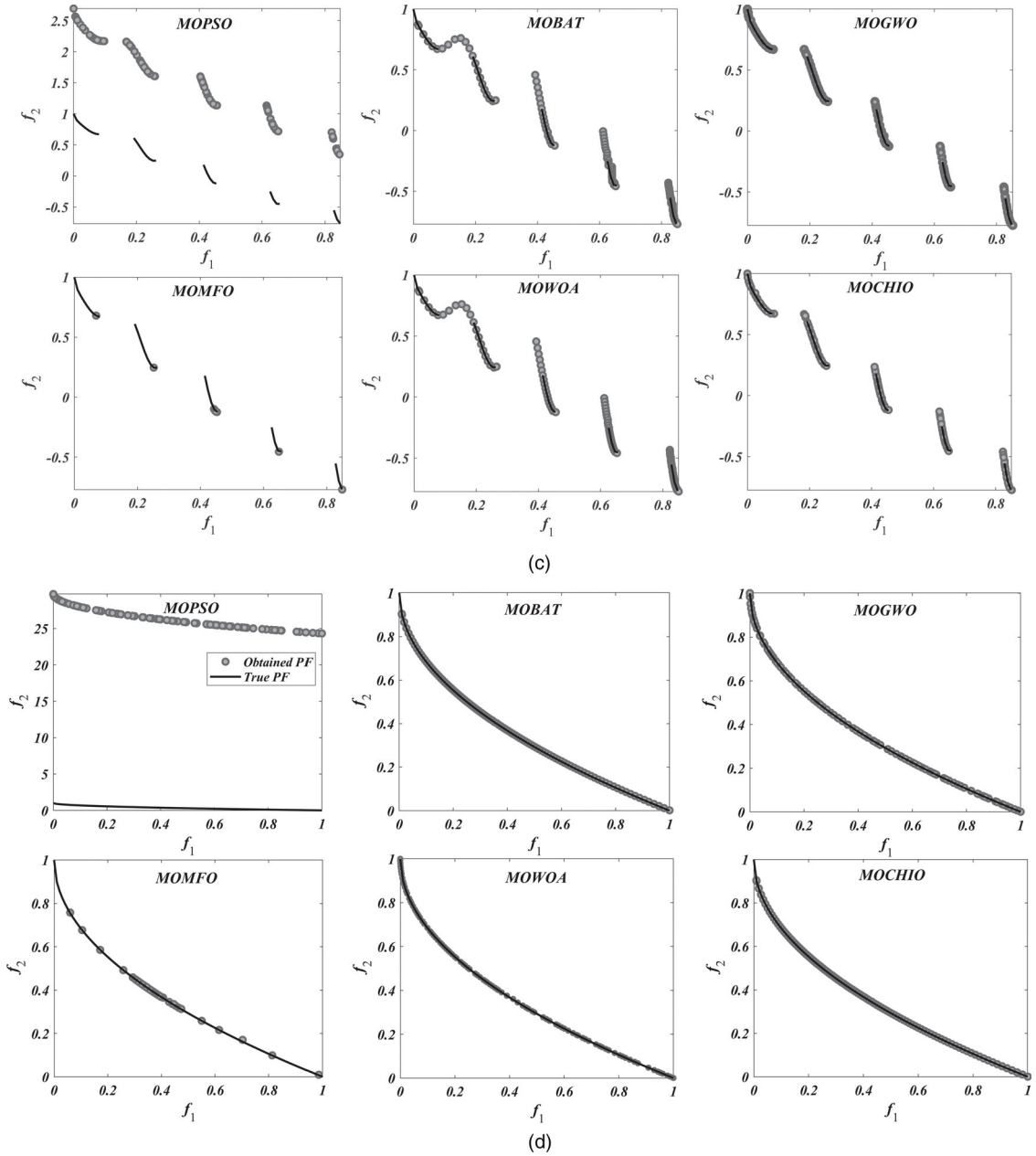

Figure 9. Continued.

In order to find the best compromise (BC) results, this study uses the fuzzy membership approach is employed with the proposed algorithm. It is used to establish BC results over the trade-off features after getting the Pareto solutions and finding the BC among the non-dominated solutions that are dynamically in the decision stage [56]. The solution to each j^{th} problem is represented by a membership function given in Eq. 68.

$$\mu_j^i = \begin{cases} 1, & f_j^i \leqslant f_{\text{lb}}^i \\ \dfrac{f_{ub}^i - f_j^i}{f_{ub}^i - f_{\text{lb}}^i}, & f_{\text{lb}}^i \leqslant f_j^i \leqslant f_{ub}^i \\ 0, & f_j^i \geqslant f_{ub}^i \end{cases} \tag{68}$$

where f_{lb}^i and f_{ub}^i represent the lower bounds and upper bounds of the fitness function. If the value of the membership function is high, then the solution is optimal. As shown in Eq. 69, the normalized membership function can be generated at each non-dominated solution (NDS).

$$\mu_j = \frac{\sum_{i=1}^{\text{N}} \mu_{ji}}{\sum_{j=1}^{M} \sum_{i=1}^{\text{N}} \mu_{ji}} \tag{69}$$

where M signifies the NDSs. The BC result is the one with the high value of μ_j.

The proposed algorithm and all other selected algorithms are applied to the BLDC wheel motor design problem, and obtained results are discussed in this section. The best optimal solutions are generated by all algorithms, and obtained PFs are plotted. Figure 10 shows the Pareto optimal solutions generated by all algorithms.

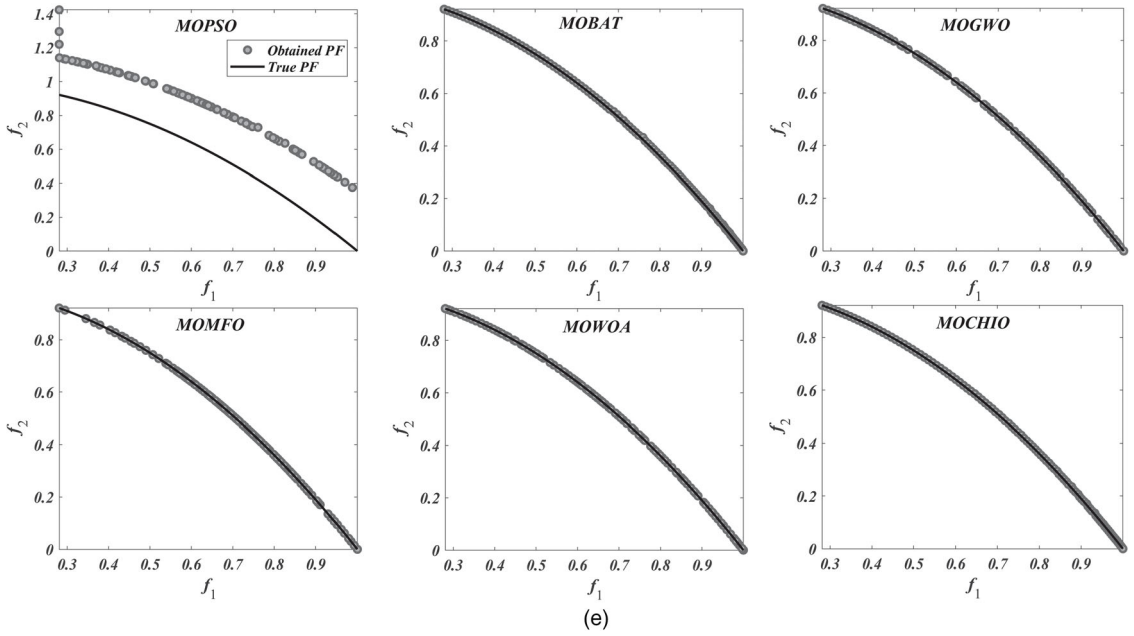

Figure 9. Continued.

In this study, in the optimization problem, a third objective function f_3 to be minimized is introduced in relation to the number of infeasible constraints. The population that meets all the given constraints, i.e. the value of f_3 equal to zero, is referred to as feasible solutions, whereas the population that does not meet at least one of the constraints is infeasible solutions. The following assumption was addressed in this context when the third objective function was adopted in this paper. Throughout the early phases of the search process, leading to local optima or conditional Pareto front, the satisfactory convergence could be lost if the infeasible solutions are fully ignored. In Figure 10, all algorithms' compromised result is illustrated to readers for their better understanding. From Figure 10, it is observed that the proposed MOCHIO algorithm is superior in handling all the constraints and produces good results compared to all other selected algorithms. In addition, the five design/optimization variables, such as B_{cs}, B_d, δ, B_e, and D_s optimized by using all six algorithms and the corresponding objective functions values, such as the motor efficiency, the motor mass, and the compromised (by balancing the motor mass and the motor efficiency) are listed in Tables 13–18.

Tables 13–18 shows that the proposed MOCHIO algorithm performs better than the other algorithms in all aspects. Thus, it is observed that the MOCHIO performs better in handling the BLDC motor design problem. The decision space and objective during the optimization process by the individual population of the MOCHIO algorithm are illustrated in Figure 11 and Figure 12, respectively. From Figure 11 and Figure 12, it can be seen how the decision variables are optimized to get the optimal Pareto solutions (objective space). It also illustrates how each population in the algorithm looks for the optimal solution by optimizing the decision variables. It is observed that none of the decision variables are not crossing the boundary limits. This is due to the fact that the proposed algorithm is provided with an effective constraint handling mechanism called a static penalty scheme. Figure 12 shows how the proposed algorithm is moving towards optimal efficiency and motor mass. Out of 100 possible population solutions, one of the optimal solutions is returned by the algorithm.

To further prove the MOCHIO algorithm's efficiency, the statistical analysis that includes minimum (Min), Mean, and standard deviation (STD) of the proposed algorithm and other algorithms are presented in Table 19. Similar to the test benchmark problems, the performance indicators, such as Delta (Δ), Epsilon (ϵ), Spread, GD, Inverted General Distance IGD, and Spacing, are listed for the BLDC motor design problem.

The proposed algorithm produces the lowest STD value, showing that the proposed algorithm's reliability is better than other competitive algorithms. The lowest value of GD and IGD indicates the fast convergence characteristics of the proposed algorithm. It can be seen from Tables 13–18 that the best efficiency (i.e. 95.36%) refers to the suggested MOCHIO, thus achieving an optimal total mass (i.e. 14.8867 Kg). The proposed MOCHIO algorithm also achieved the lowest mass value (i.e. 10.6509 Kg) in terms of total motor mass without disturbing the second objective (i.e. 93.92%). Furthermore, it can be noticed that in terms of Δ, ϵ, Spread, GD, IGD, and Spacing outperformed MOPSO, MOBA, MOGWO, MOMFO, MOWOA algorithms by having 0.8589, 0.0234, 0.7009, 3.1869e-04, 0.0071, and 0.0152. Therefore, the dominance of the MOCHIO algorithm over MOPSO, MOGWO, MOMFO, MOBA,

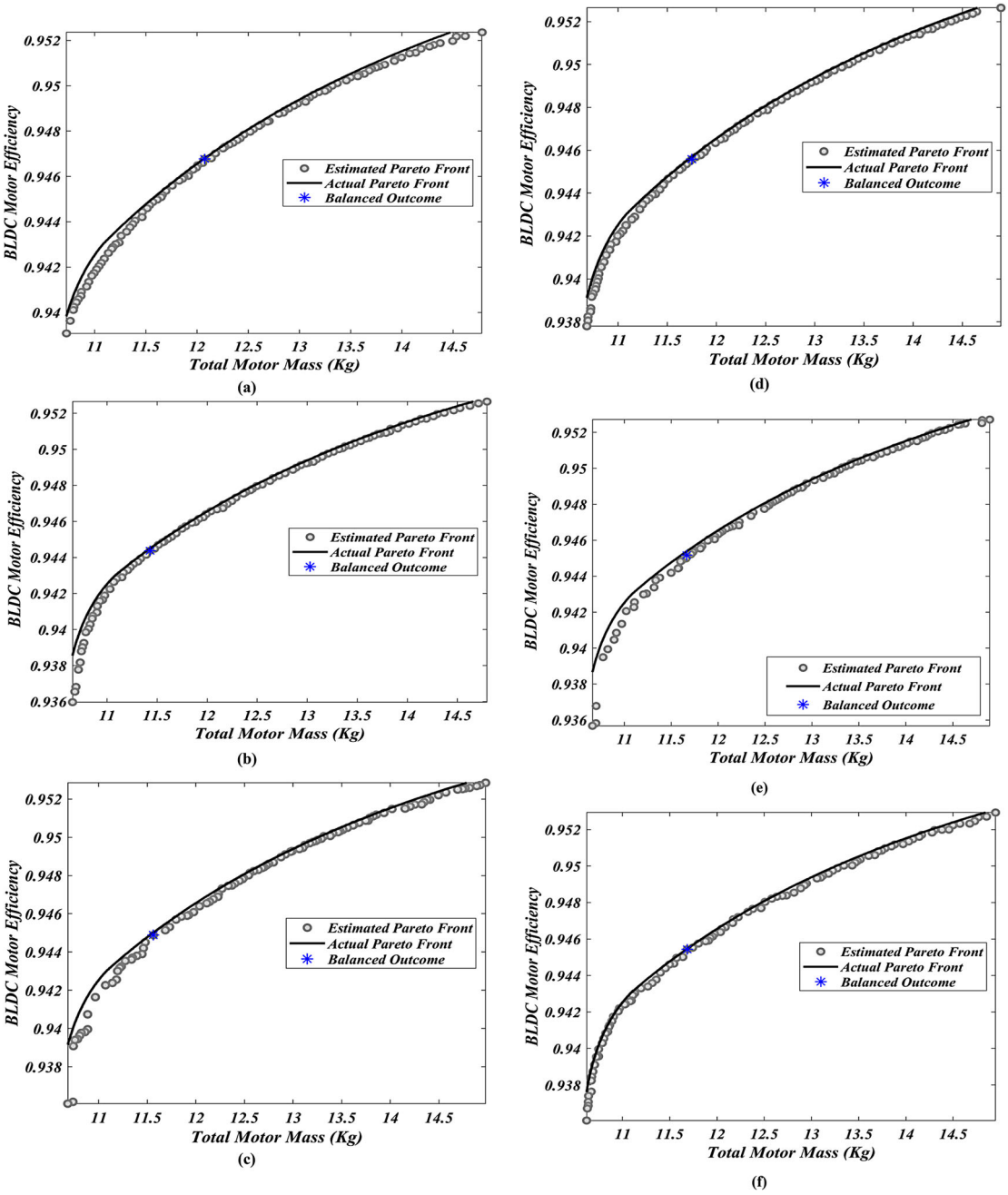

Figure 10. Pareto optimal solutions obtained by various algorithms; (a) MOPSO, (b) MOBA, (c) MOGWO, (d) MOMFO, (e) MOWOA, (f) MOCHIO

Table 13. Best results obtained by the MOPSO algorithm

Indices	B_d (T)	B_e (T)	B_{cs} (T)	D_s (m)	δ (A/mm^2)	M_{Tt} (Kg)	η (%)
$\boldsymbol{f_1}$	1.7999	0.6640	1.6000	0.1860	3670737.688	10.6213	93.60
$\boldsymbol{f_2}$	1.7896	0.6608	0.8854	0.2012	2137080.216	14.9478	95.29
BC values	1.8000	0.6617	1.5581	0.1804	2720384.343	12.0744	94.54

Table 14. Best results obtained by the MOBA

Indices	B_d (T)	B_e (T)	B_{cs} (T)	D_s (m)	δ (A/mm^2)	M_{Tt} (Kg)	η (%)
f_1	1.7926	0.6651	1.6000	0.1865	3687048.881	10.6555	93.59
f_2	1.7855	0.6620	0.8642	0.1999	2186808.398	14.8957	95.26
BC values	1.7928	0.6620	1.5153	0.1784	2832724.661	11.4291	94.43

Table 15. Best results obtained by the MOGWO algorithm

Indices	B_d (T)	B_e (T)	B_{cs} (T)	D_s (m)	δ (A/mm^2)	M_{Tt} (Kg)	η (%)
f_1	1.7738	0.6649	1.5750	0.1861	3684518.517	10.6835	93.60
f_2	1.7641	0.6645	0.8769	0.2011	2175044.191	14.9799	95.28
BC values	1.7927	0.6636	1.5508	0.1795	2783871.123	11.5635	94.48

Table 16. Best results obtained by the MOMFO algorithm

Indices	B_d (T)	B_e (T)	B_{cs} (T)	D_s (m)	δ (A/mm^2)	M_{Tt} (Kg)	η (%)
$\boldsymbol{f_1}$	1.78768	0.65938	1.57208	0.18398	3490639.719	10.6823	93.78
$\boldsymbol{f_2}$	1.7584	0.6596	1.0422	0.2037	2149179.188	14.8911	95.26
BC values	1.7879	0.6592	1.4854	0.1807	2720537.401	11.7505	94.56

Table 17. Best results obtained by the MOWOA

Indices	B_d (T)	B_e (T)	B_{cs} (T)	D_s (m)	δ (A/mm^2)	M_{Tt} (Kg)	η (%)
$\boldsymbol{f_1}$	1.7849	0.6591	1.5850	0.1852	3583296.644	10.6519	93.68
$\boldsymbol{f_2}$	1.7761	0.6587	0.9975	0.2006	2057684.395	14.9433	95.29
BC values	1.7874	0.6631	1.5276	0.1810	2714549.090	11.7904	94.56

Table 18. Best results obtained by the proposed MOCHIO algorithm

Indices	B_d (T)	B_e (T)	B_{cs} (T)	D_s (m)	δ (A/mm^2)	M_{Tt} (Kg)	η (%)
$\boldsymbol{f_1}$	1.7946	0.6593	1.5432	0.1818	3366034.636	10.6509	93.92
$\boldsymbol{f_2}$	1.7881	0.6597	0.9365	0.2005	2075616.953	14.8867	95.36
BC values	1.7989	0.6595	1.4536	0.1829	2608762.151	11.6852	94.68

and MOWOA in the aforementioned performance metrics can be concluded. The results of the design specifications attained by the MOCHIO are presented in Table 18 when the efficiency is maximum, or the total mass is minimum, or compromise between the best efficiency and best mass.

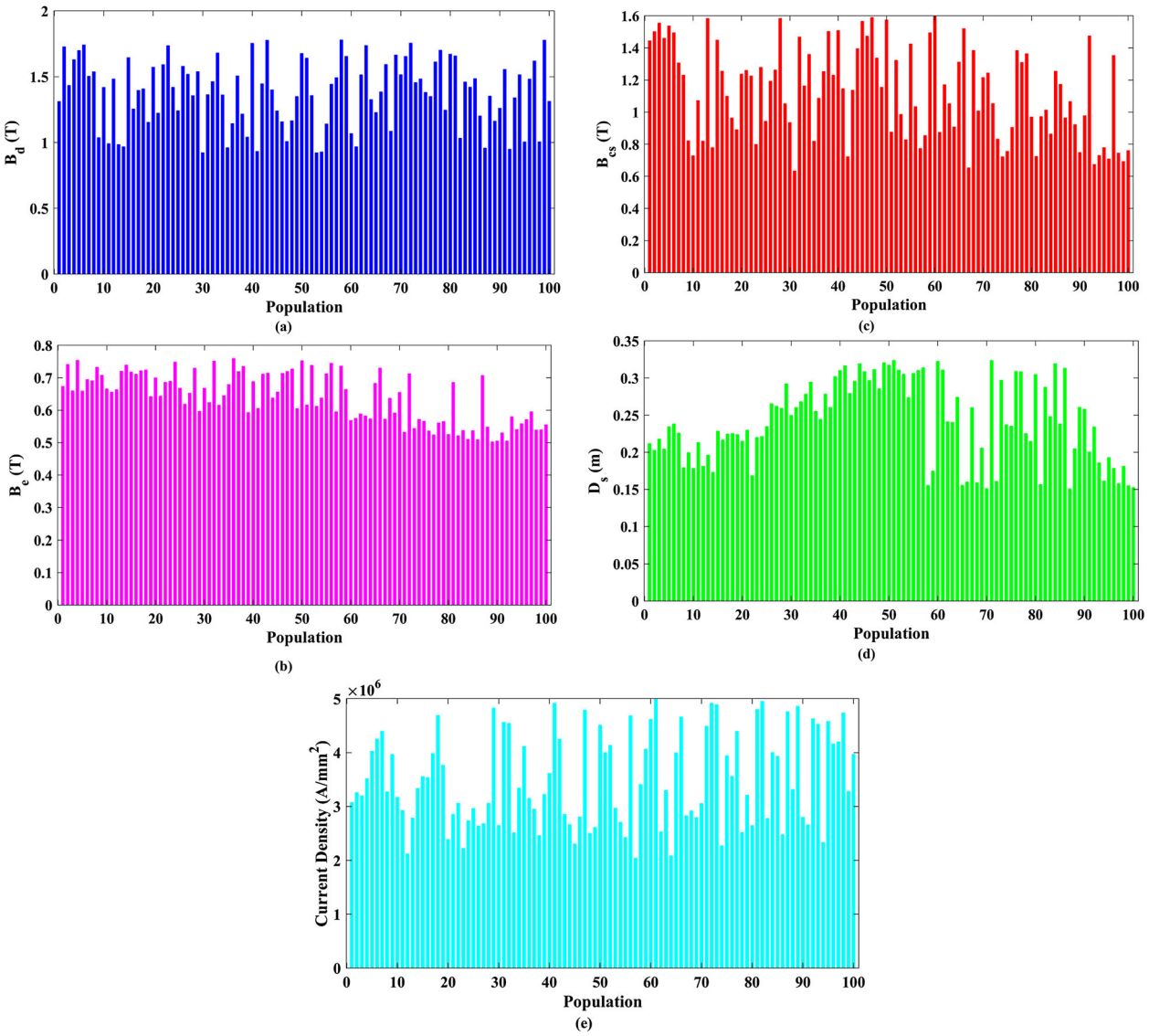

Figure 11. Decision variables optimized in each population; (a) B_d, (b) B_e, (c) B_{cs}, (d) D_s, (e) δ

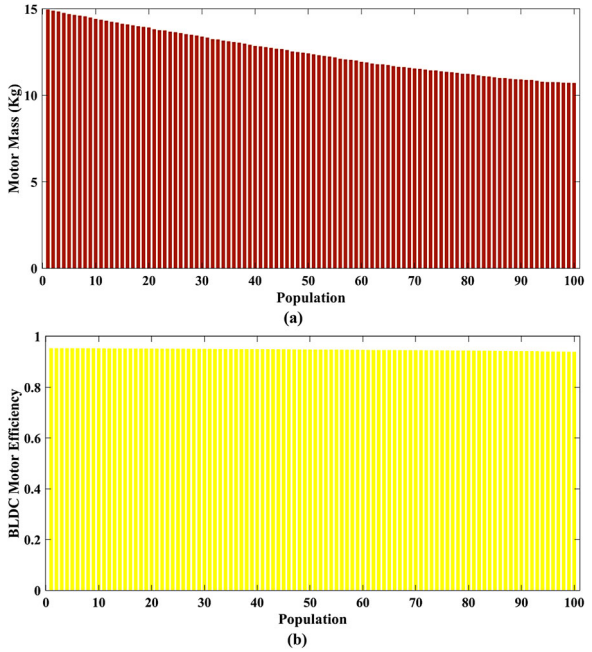

Figure 12. Objective functions optimized in each population; (a) Motor total mass, (b) Motor efficiency

5. Conclusions

In general, swarm intelligence and evolutionary algorithms have proved to be robust and effective methods for finding the optimal Pareto front for MOOPs. In this situation, the CHIO algorithm is a recent optimization algorithm that imitates the herd immunity behavior of the infected individual, ultimately leading the individual to the best chance of survival. The effectiveness of the MOCHIO algorithm on the BLDC motor design problem was validated in this paper. The non-dominated sorting scheme and CD strategies were used to handle with the multi-objective BLDC motor design problem and achieve an acceptable tradeoff between the total mass and motor efficiency. Considering total mass and efficiency as two fitness functions, the PFs observed compared to MOPSO, MOBA, MOGWO, MOMFO, and MOWOA demonstrate that even without spoiling the other one, the suggested MOCHIO algorithm can achieve the minimum values for both objectives respectively. Compared to other selected algorithms, the proposed MOCHIO algorithm also achieves the minimum values in terms of Δ, ϵ, Spread, GD, IGD, and Spacing. Therefore, it can be inferred that MOCHIO is an efficient algorithm applied to the BLDC motor design problem.

For future research, we examine the proposed MOCHIO algorithm's implementation in other electromagnetic optimization-related case studies. Moreover, the effectiveness of various robust MOCHIO handling techniques will be examined to enable MOCHIO to deal with different kinds of uncertainties, which are essential to solving real-world engineering optimization problems. The proposed algorithm can be

Table 19. Statistical analysis of all algorithms for BLDC motor design problem

	DM			Epsilon			Spread			GD			IGD			Spacing		
Algorithms	Min	Mean	STD	Min	Mean	STD	Min	Mean	STD	Min	Mean	STD	Min	Mean	STD	Min	Mean	STD
MOPSO	0.8780	0.8855	0.0091	0.0713	0.1374	0.0432	0.7013	0.7262	0.0229	3.9049e-04	7.3133e-04	4.3823e-04	0.0071	0.0074	1.0679e-04	0.0178	0.0197	0.0014
MOBAT	0.8624	0.8656	0.0041	0.0728	0.1015	0.0390	0.7171	0.7508	0.0299	3.2097e-04	4.3418e-04	1.2952e-04	0.0072	0.0074	6.5631e-05	0.0138	0.0184	0.0030
MOGWO	0.8758	0.8862	0.0091	0.0672	0.1109	0.0296	0.8078	0.8436	0.0252	5.1731e-04	5.9350e-04	1.1929e-04	0.0071	0.0074	1.0754e-04	0.0185	0.0199	0.0014
MOMFO	0.8594	0.8729	0.0124	0.0996	0.1300	0.0308	0.7085	0.7582	0.0584	3.4691e-04	5.9059e-04	3.9110e-04	0.0072	0.0073	1.4446e-04	0.0139	0.0214	0.0062
MOWOA	0.8757	0.8953	0.0204	0.0692	0.0977	0.0272	0.8309	0.8566	0.0313	4.5703e-04	5.17e-04	5.4205e-05	0.0071	0.0074	7.0194e-05	0.0178	0.0194	0.0014
MOCHIO	0.8589	0.8738	0.0080	0.0234	0.0370	0.0154	0.7009	0.7377	0.0164	3.1869e-04	4.1729e-04	5.0394e-05	0.0071	0.0072	6.4870e-05	0.0152	0.0172	0.0013

applied to other multi-objective engineering optimization problems, such as vehicle routing, feature selection, controller tuning, optimal capacitor placements, distributed generation units, FACTS devices, economic emission dispatch, economic load dispatch, optimal power flow problems, etc.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

C. Kumar http://orcid.org/0000-0002-1132-4794

References

[1] Ramu K. Permanent Magnet Synchronous and Brushless DC Motor Drives, 1st ed. UK: CRC Press, 2009. Accessed: Sep. 06, 2021. [Online]. Available: https://www.routledge.com/Permanent-Magnet-Synchronous-and-Brushless-DC-Motor-Drives/Krishnan/p/book/9780824753849.

[2] Zhu X, Cheng M. Design, analysis and control of hybrid excited doubly salient stator-permanent-magnet motor. Sci China Ser E Technol Sci. Jan. 2010;53(1):188–199. doi:10.1007/S11431-009-0357-0.

[3] Mansouri A, Msaddek H, Trabelsi H, et al. Optimum design of a surface mounted fractional slots Permanent Magnet motor. Int Rev Elect Eng. Feb. 2015;10(1):28–35. doi:10.15866/IREE.V10I1.4104.

[4] Mansouri A, Njeh A, Makni Z, et al. On the torque production capability of a claw pole TFPM. Int J Appl Electromagnet Mech. Jan. 2004;19(1–4):391–394. doi:10.3233/JAE-2004-596

[5] Yadav P, Kumar R, Panda SK, et al. Improved harmony search algorithm based optimal design of the brushless DC wheel motor, 2010 IEEE International Conference on Sustainable Energy Technologies, ICSET 2010, 2010, doi: 10.1109/ICSET.2010.5684426.

[6] Mbadiwe EI, bin Sulaiman E. Design and optimization of outer-rotor permanent magnet flux switching motor using transverse segmental rotor shape for automotive applications. Ain Shams Engineering Journal. Mar. 2021;12(1):507–516. doi:10.1016/J.ASEJ.2020.08.007

[7] Luk PCK, Xia B, Wu D, et al. Design considerations of outer-rotor Permanent Magnet Synchronous machines for In-wheel Electric drivetrain using particle swarm optimization). 7th IET International Conference on power electronics, Machines and Drives (PEMD 2014), pp. 0393–0393, 2014, doi:10.1049/CP.2014.044910.1049/CP.2014.0449.

[8] dos Santos Coelho L, Barbosa LZ, Lebensztajn L. Multiobjective particle swarm approach for the design of a brushless DC wheel motor. IEEE Trans Magn. Aug. 2010;46(8):2994–2997. doi:10.1109/TMAG.2010.2044145.

[9] Bora TC, Coelho LDS, Lebensztajn L. Bat-inspired optimization approach for the brushless DC wheel motor problem. IEEE Trans Magn. Feb. 2012;48(2):947–950. doi:10.1109/TMAG.2011.2176108.

[10] Premkumar M, Sowmya R, Jangir P, et al. "A New metaheuristic optimization algorithms for brushless Direct Current Wheel Motor Design problem. Computers, Materials and Continua. 2021;67(2):2227–2242. doi:10.32604/CMC.2021.015565.

[11] Nugraha YU, Cahyadi A, Yuniarto MN, et al. Design optimization for torque density in brushless DC motor with IPM V-type using PSO method. IOP Conference Series: Materials Science and Engineering. Nov. 2019;694(1):012009, doi:10.1088/1757-899X/694/1/012009.

[12] Markovic M, Ragot P, Perriard Y. Design optimization of a BLDC motor: A comparative analysis. Proceedings of IEEE International Electric Machines and Drives Conference, IEMDC 2007. 2007;2:1520–1523. doi:10.1109/IEMDC.2007.383653.

[13] Jurković M, Žarko D. Optimized design of a brushless DC Permanent Magnet motor for propulsion of an ultra light aircraft. Automatika. 2012;53(3):244–254. doi:10.7305/AUTOMATIKA.53-3.114.

[14] Shabanian A, Tousiwas AAP, Pourmandi M, et al. Optimization of brushless direct current motor design using an intelligent technique. ISA Trans. Jul. 2015;57:311–321. doi:10.1016/J.ISATRA.2015.03.005.

[15] Upadhyay PR, Rajagopal KR. Genetic algorithm based design optimization of a permanent magnet brushless dc motor. J Appl Phys. May 2005;97(10):10Q516-1–24. doi:10.1063/1.1860891.

[16] Potnuru UK, Rao PM. Design optimization for permanent magnet machine with efficient slot per pole ratio. AIP Conf Proc. Apr. 2018;1952(1):020117-1–020117-8. doi:10.1063/1.5032079.

[17] Kim HK, Chong JK, Lowther DA. Differential evolution strategy for constrained global optimization and application to practical engineering problems. 12th biennial IEEE Conference on electromagnetic field computation, CEFC. 2006, p. 238, 2006, doi:10.1109/CEFC-06.2006.1633028.

[18] Rao KSR, bin Othman AH. Design optimization of a bldc motor by genetic algorithm and simulated annealing). 2007 International Conference on Intelligent and advanced systems, ICIAS. 2007, pp. 854–858, doi:10.1109/ICIAS.2007.4658508.

[19] Ridwan M, Riawan DC, Suryoatmojo H. "Particle swarm optimization-based BLDC motor speed controller with response speed consideration," 2017 International Seminar on Intelligent Technology and Its Application: Strengthening the Link Between University Research and Industry to Support ASEAN Energy Sector, ISITIA 2017 - Proceeding, vol. 2017-January, pp. 193–198, Nov. 2017, doi: 10.1109/ISITIA.2017.8124079.

[20] Chakkarapani K, Thangavelu T, Dharmalingam K. Thermal analysis of brushless DC motor using multiobjective optimization. Int Trans Elect Ener Sys. Oct. 2020;30(10):e12546-1–18. doi:10.1002/2050-7038.12546.

[21] Premkumar K, Manikandan Bv. Speed control of brushless DC motor using bat algorithm optimized Adaptive neuro-fuzzy inference system. Appl Soft Comput. Jul. 2015;32:403–419. doi:10.1016/J.ASOC.2015.04.014.

[22] Ayala HVH, Segundo EHV, Mariani VC, et al. Multiobjective Krill herd algorithm for electromagnetic optimization. IEEE Trans Magn. Mar. 2016;52(3):1–4. doi:10.1109/TMAG.2015.2483060.

[23] Mirjalili S, Saremi S, Mirjalili SM, et al. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. Expert Syst Appl. Apr. 2016;47:106–119. doi:10.1016/J.ESWA.2015.10.039.

[24] Ghasemi M, Bagherifard K, Parvin H, et al. Multi-objective whale optimization algorithm and multi-objective grey wolf optimizer for solving next release problem with developing fairness and uncertainty quality indicators. Appl Intell 51:8. Jan. 2021;51(8):5358–5387. doi:10.1007/S10489-020-02018-2.

[25] Jangir P. Non-Dominated sorting Moth Flame Optimizer: A Novel Multi-Objective Optimization Algorithm for Solving Engineering design problems. Eng Techn

Open Access Journ. 2018;2(1):17–31. doi:10.19080/etoaj.2018.02.555579.
[26] Zhang Z, et al. Improved multi-objective moth-flame Optimization Algorithm based on R-domination for cascade reservoirs operation. J Hydrol (Amst). Feb. 2020;581:124431-1–32. doi:10.1016/J.JHYDROL.2019.124431.
[27] A MD, Silva CE, Coelho LDS, et al. Multiobjective biogeography-based optimization based on predator-prey approach. IEEE Trans Magn. Feb. 2012;48(2):951–954. doi:10.1109/TMAG.2011.2174205.
[28] Sharifi M, Mojallali H. Multi-Objective modified imperialist competitive algorithm for brushless DC motor optimization. IETE J Res. Jan. 2017;65(1):96–103. doi:10.1080/03772063.2017.1391130.
[29] Qiu H, Duan H. Multi-objective pigeon-inspired optimization for brushless direct current motor parameter design. Science China Technological Sciences. Jun. 2015;58(11):1915–1923. doi:10.1007/S11431-015-5860-X.
[30] Moussouni F, Brisset S, Brochet P. Some results on the design of brushless DC wheel motor using SQP and GA. Int J Appl Electromagnet Mech. Jan. 2007;26(3–4):233–241. doi:10.3233/JAE-2007-913.
[31] Duan H, Li S, Shi Y. Predator-prey brain storm optimization for DC brushless motor. IEEE Trans Magn. 2013;49(10):5336–5340. doi:10.1109/TMAG.2013.2262296.
[32] Premkumar M, et al. A New arithmetic Optimization Algorithm for solving real-world Multiobjective CEC-2021 constrained optimization problems: diversity analysis and validations. IEEE Access. 2021;9:84263–84295. doi:10.1109/ACCESS.2021.3085529.
[33] Premkumar M, Sowmya R, Jangir P, et al. MOSMAâĂŕ: multi-objective slime mould algorithm based on elitist Non-dominated sorting. IEEE Access. 2021;9:3229–3248. doi:10.1109/ACCESS.2020.3047936.
[34] Bui LT, Liu J, Bender A, et al. DMEA: a direction-based multiobjective evolutionary algorithm. Memetic Computing. 2011;3(4):271–285. doi:10.1007/s12293-011-0072-9.
[35] Premkumar M, Jangir P, Sowmya R. MOGBO: A new Multiobjective gradient-based optimizer for real-world structural optimization problems. Knowl Based Syst. 2021;218:106856-1–40. doi:10.1016/j.knosys.2021.106856.
[36] Wu X, Zhang S, Gong Z, et al. Decomposition-Based Multiobjective evolutionary optimization with Adaptive Multiple Gaussian process models. Complexity. 2020;2020:9643273-1–22. doi:10.1155/2020/9643273.
[37] Kumar S, Jangir P, Tejani GG, et al. MOPGO: A New physics-based multi-objective plasma Generation optimizer for solving structural optimization problems. IEEE Access. 2021;9:84982–85016. doi:10.1109/ACCESS.2021.3087739.
[38] Fan Z, Wang T, Cheng Z, et al. An improved Multiobjective particle swarm optimization algorithm using minimum distance of point to line. Shock Vib. 2017;2017:8204867-1–16. doi:10.1155/2017/8204867.
[39] Al-Betar MA, Alyasseri ZAA, Awadallah MA, et al. Coronavirus herd immunity optimizer (CHIO). Neural Computing and Applications 2020 33:10. Aug. 2020;33(10):5011–5042. doi:10.1007/S00521-020-05296-6.
[40] Dalbah LM, Al-Betar MA, Awadallah MA, et al. A modified coronavirus herd immunity optimizer for capacitated vehicle routing problem. Journal of King Saud University - Computer and Information Sciences. Jun. 2021: 1–14. doi:10.1016/J.JKSUCI.2021.06.013.
[41] Neto AA, Neto ASA. Optimization inspired on herd immunity applied to Non-hierarchical grouping of objects. Revista de Informática Teórica e Aplicada. Aug. 2021;28(2):50–65. doi:10.22456/2175-2745.107478.
[42] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput. 1997;1(1):67–82.
[43] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput. Apr. 2002;6(2):182–197. doi:10.1109/4235.996017.
[44] Tian Y, Liu P, Li Z. Multi-objective optimization of shock control bump on a supercritical wing. Science China Technological Sciences 2013 57:1. Nov. 2013;57(1):192–202. doi:10.1007/S11431-013-5410-3.
[45] Fan L, et al. Multi-Objective optimal configuration of Multiple switchgear Considering distribution network fault reconfiguration. IEEE Access. 2021;9:69905–69912. doi:10.1109/ACCESS.2021.3051338.
[46] Li H, Zhang Q. Multiobjective optimization problems With complicated Pareto sets, MOEA/D and NSGA-II. IEEE Trans Evol Comput. 2009;13(2):284–302. doi:10.1109/TEVC.2008.925798.
[47] Bi X, Wang C. A niche-elimination operation based NSGA-III algorithm for many-objective optimization. Applied Intelligence. 2018;48(1):118–141. doi:10.1007/s10489-017-0958-4.
[48] Zhang J, Wang S, Tang Q, et al. An improved NSGA-III integrating adaptive elimination strategy to solution of many-objective optimal power flow problems. Energy. 2019;172:945–957. doi:10.1016/j.energy.2019.02.009.
[49] Abello MB, Michalewicz Z. Implicit memory-based technique in solving dynamic scheduling problems through response surface methodology – part II: experiments and analysis. Int J Intell Comp Cybern. 2014;7(2):143–174. doi:10.1108/IJICC-12-2013-0054.
[50] Bianchi N, Bolognani S. "Brushless DC motor design: An optimization procedure based on genetic algorithms," IEE Conference Publication, pp. 16–20, 1997, no. 444 doi: 10.1049/CP:19971029.
[51] Rahideh A, Korakianitis T, Ruiz P, et al. Optimal brushless DC motor design using genetic algorithms. J Magn Magn Mater. Nov. 2010;322(22):3680–3687. doi:10.1016/J.JMMM.2010.07.025.
[52] Brisset S, Brochet P. Analytical model for the optimal design of a brushless DC wheel motor. COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering. 2005;24(3):829–848. doi:10.1108/03321640510612952.
[53] Brisset S, Brochet P. "A Benchmark for a Mono and Multi-Objective Optimization of the Brushless DC Wheel Motor," *L2EP, École Centrale de Lille, Villeneuve-d'Ascq, France*. http://l2ep.univ-lille1.fr/come/benchmarkwheel-motor.htm (accessed Sep. 06, 2020).
[54] Jangir P, Buch H, Mirjalili S, et al. MOMPA: multi-objective marine predator algorithm for solving multi-objective optimization problems. Evol Intell. 2021:1–27. doi:10.1007/S12065-021-00649-Z.
[55] Tharwat A, Houssein EH, Ahmed MM, et al. MOGOA algorithm for constrained and unconstrained multi-objective optimization problems. Applied Intelligence. Nov. 2017;48(8):2268–2283. doi:10.1007/S10489-017-1074-1.
[56] Opricovic S. A fuzzy compromise solution for multi-criteria problems. Int J Uncertainty Fuzziness Knowledge Based Syst. 2007;15(3):363–380. doi:10.1142/S0218488507004728.