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, and T. Gunasekar

Department of Electrical and Electronics Engineering, M. Kumarsamy College of Engineering, Karur, Tamil Nadu, India; Department of Electrical and Electronics Engineering, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India

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 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.

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

© 2021 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group. This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (http://creativecommons.org/licenses/by-nc/4.0/), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
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 its 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 multi-objective evolutionary optimization [36], multi-objective plasma generation optimizer [37], improved multi-objective 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:

$$\min f(x) x \in [l_{lb}, u_{ub}]$$  \hspace{1cm} (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 (MaxAge) and basic reproduction rate (BRr), 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:

$$HIP = \begin{bmatrix} x^1_1 & \cdots & x^1_n \\ \vdots & \ddots & \vdots \\ x^N_{HIS} & \cdots & x^N_{HIS} \end{bmatrix}$$ \hspace{1cm} (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 BRr, the individual gene maintains the same or is influenced by social distance using three following rules.

$$x^i(t + 1) = \begin{cases} x^i(t), & r \geq BRr \\ C(x^i(t)), & r < \frac{1}{3} \times BRr \text{ (Infected) } \\ N(x^i(t)), & \frac{2}{3} \times BRr < r < BRr \text{ (Susceptible) } \\ R(x^i(t)), & r < BRr \text{ (Immunned)} \end{cases}$$ \hspace{1cm} (3)

where \( r \) denotes random number between [0,1]. The infected case is in the range of 0 to \( \frac{1}{3} \times BRr \). 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.

$$x^i(t + 1) = C(x^i(t))$$ \hspace{1cm} (4)

$$C(x^i(t)) = x^i(t) + r \times (x^i(t) - x^i(t))$$ \hspace{1cm} (5)

Note that \( x^i(t) \) is selected randomly from the diseased person based on the \( S \). The susceptible case is in the range of \( \frac{1}{3} \times BRr \) to \( \frac{2}{3} \times BRr \), 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.

$$x_j^i(t+1) = N(x_j^i(t))$$

$$N(x_j^i(t)) = x_j^i(t) + r \times (x_j^i(t) - x_j^m(t))$$

Note that $x_j^m(t)$ is selected randomly from the vulnerable case based on the $S_j$. 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.

$$x_j^i(t+1) = R(x_j^i(t))$$

$$R(x_j^i(t)) = x_j^i(t) + r \times (x_j^i(t) - x_j^f(t))$$

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

$$f(x^f) = \arg \min_{j \sim \{k | S_k = 2\}} f(x^k)$$

**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 $f(xx'(t+1)) < f(xx'(t))$. 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.

$$S_j = \begin{cases} 
1, & f(xx'(t+1)) < \frac{f(xx'(t+1))}{\Delta f(xx)} \\
0, & \text{otherwise} \\
\end{cases}$$

where $is\_Corona(xx'(t+1))$ 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 $x_j^i(t+1) = lb_i + (ub_i - lb_i) \times U(0, 1)$. 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 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.

\[ \text{Min/Max, } F(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x})\} \]

Subject to: \( p_i(\vec{x}) \geq 0, j = 1, 2, \ldots, g \)
\[ q_j(\vec{x}) = 0, j = 1, 2, \ldots, h \]
\[ lb_j \leq x_j \leq ub_j, j = 1, 2, \ldots, n \]

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 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].

\[
\begin{align*}
    f_i(X_i) \leq f_i(X_j), & \text{ for all } l = 0, 1, 2, \ldots, n \\
    f_l(X_i) \leq f_l(X_j), & \text{ for at least one } l \in \{0, 1, 2, \ldots, n\}
\end{align*}
\]

(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 \( S^X_X \).

**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.

\[ CD_j = \frac{f_l(X_{l+1}) - f_l(X_{l-1})}{f_l^{\text{max}} - f_l^{\text{min}}} \]

(14)

Where, \( f_l^{\text{min}} \) and \( f_l^{\text{max}} \) 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 \( IT_{\text{max}} \).
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 \( IT_{\text{max}} \), return to step-2. Else the current positions are evaluated and return the optimal Pareto front \( S^X_f \).
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.
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 out-runner 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.

\[
P_m = \sum_{i=1}^{p_h} e_i(t) \times i_i(t)
\]  

Where \( i_i \) denotes the phase current, \( e_i \) denotes the phase voltage, \( P_h \) denotes the number of phases, and \( N \) denotes the speed. In the BLDC motor, there are \( 3\phi \) 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 \( P_m \).

\[
T_m \times N = 2 \times E \times I
\]  

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.

\[ E = \frac{z}{4} \times \frac{d\phi}{dt} \times \frac{d\theta}{dt} = \frac{z}{4} \times \frac{2\phi}{\pi} \times N \]  
(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 (\( \phi \)) is calculated using Eq. 18.

\[ \phi = B_c \times A_p \]  
(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.

\[ T_m = z \times I \times B_c \times \frac{A_e}{2 \times \pi} \]  
(19)

Total air gap area, \( A_e = 2 \times p \times A_p \)  
(20)

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

\[ A_e = \pi \times D_s \times L_m \]  
(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.

\[ E = \frac{z}{4} \times B_c \times D_s \times L_m \times N \]  
(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 to maximize the flux and that the EMF peak is now the maximum.

\[ \alpha = \beta = \frac{\pi}{p} \]  
(23)

The analytical technique revealed the same patterns as the sensitivity of the thickness of the EMF spike. For the variables \( \alpha \) and \( \beta \), 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 \( \alpha_l \) 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.

\[ \alpha_l = \alpha \frac{5}{\pi} \]  
(24)

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

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

\[ A_{nc} \times K_r = \frac{3}{\delta} \times \frac{z \times I}{5} \]  
(26)

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

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

\[ h_c = \frac{e_b}{\cos \frac{\varphi}{2}} - \frac{D_s}{2} \times \left( \frac{1}{\cos \frac{\varphi}{2}} - 1 \right) \]  
(29)

\[ h_i = \frac{D_s}{2 \times \left( 1 - \cos \frac{\alpha_l}{2} \right) + h_c \times \cos \frac{\alpha_l}{2} } \]  
(30)

where \( S_e \) denotes the total number of slots, \( D_{int} \) and \( D_{ext} \) denote the inner and outer diameters, \( \delta \) denotes the conductor current density, \( K_r \) denotes the factor due to slot filling (< 1), and \( A_{nc} \) denotes the total slot sections. The hold height \( h_i \) is found so that the elevation of a middle tooth is equal to one of a primary tooth \( h_{mc} \), 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.

\[ R_t \approx \frac{l_d - l_t}{4} + \left[ \frac{D_s}{2} - e_b - \frac{h_d}{2} \right] \times \frac{\pi}{2} \times S_e \] (31)

\[ L_d \approx \frac{L_m}{k_f} + \pi \times R_t \] (32)

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

\[ L_T \approx \frac{L_m}{k_f} + 2 \times \left[ \frac{D_s}{2} - e_b - \frac{h_d}{2} \right] \times \frac{\pi}{2} \times \frac{l_t}{2} \] (33)

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

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

\[ M_{cs} = d_i \times \pi \times h_{cs} \times \left[ 2 \times \left( \frac{D_s}{2} - e_b - h_d \right) \right] \times L_m \times r_{rs} \] (36)

\[ M_{ds} = d_i \times S_e \times \left[ \left( l_i + l_d \right) \right] \times h_d + \left( \frac{e_b + h_u}{2} + \frac{h_i + h_c}{2} \right) \times L_m \] (37)

\[ M_{cu} = d_{cu} \times \frac{3}{2} \times z \times \frac{l}{5} \times L_{ds} \] (38)

The total mass of all the active parts \( M_{Ti} \) is the sum of individual weights of all parts.

\[ M_{Ti} = M_a + M_{cr} + M_{cs} + M_{ds} + M_{cu} \] (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, rotor yoke, rotor yoke, and magnets, respectively, \( d_{cu}, d_i, d_{cr}, \) and \( d_a \) denote the density of copper, steel 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 \( B_{cr}, B_{br}, B_a, \) 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.

\[ B_d \times l_d = B_e \times \alpha \times \frac{D_s}{2} \] (40)

\[ B_d \times l_i = B_e \times \alpha_i \times \frac{D_s}{2} \] (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.

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

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

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

\[ \frac{1}{2} \times B_a \times l_d = B_{cs} \times h_{cs} \] (44)

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

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

where \( k_\beta \) 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.

\[ \frac{B_{cr}}{\mu_{cr}(B_{cr})} \times \left[ \frac{h_{cr}}{2} + \frac{\pi}{2} \times \left( \frac{D_s}{2} + e + h_u + h_{cr} \right) \right] \]

\[ + \left[ \frac{B_{cr} - B_r(1 + \alpha_a \times T_a)}{\mu_a} \right] \times h_u + B_e \]

\[ \times e + \frac{B_{cr}}{\mu_{cr}(B_{cr})} \left( e_b + h_d \right) + \frac{B_{cs}}{\mu_{cs}(B_{cs})} \]

\[ \times \left[ \frac{h_{cs}}{2} + \frac{\pi}{2} \times \left( \frac{D_s}{2} - e_b - h_d - h_{cs} \right) \right] = 0 \] (46)

where \( \mu_{cr} \) and \( \mu_{t} \) denote the relative permeabilities of the rotor yoke and sheet metal, \( \alpha_a < 0, B_r, \) and \( \mu_a \) 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
in Eq. 51. Using Eq. 50, and the fundamental frequency is given
the thermal coefficient, and \( \rho \) denote the copper
where Eq. 46 is simplified as seen in Eq. 47
Sufficient current flows in the coils whenever the motor
The electrical loss can be found by defining the phase
\[
\frac{(B_e - B_r(1 + \alpha_c \times T_a))}{\mu_a} \times h_a + B_e \times e = 0
\]  \tag{47}
\[
\frac{(B_c - B_r(1 + \alpha_c \times T_a))}{\mu_a \times \mu_o} \times h_a + \frac{z \times I_{\text{max}}}{4\rho} + \frac{B_c}{\mu_o} \times \frac{\beta}{\alpha} \times \left(1 + \frac{2e}{D_r}\right) \times r_s \times k_f \times e = 0
\]  \tag{48}

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.
\[
T_a = T_{cu}
\]  \tag{54}

In the theory of an enclosed motor, the exterior surface
where convection occurs is equal to Eq. 55.
\[
A_{\text{ext}} = \frac{\pi}{2} \times D_{\text{ext}}^2 + \pi \times D_{\text{ext}} \times L_T
\]  \tag{55}

Therefore, the overall temperature of the motor is calculated
using Eq. 56.
\[
T_{cu} = T_{\text{ext}} + \frac{P_m + P_{cu} + P_{ir}}{h \times A_{\text{ext}}}
\]  \tag{56}


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


\[ P_{cu} = 2 \times R_{ph} \times I^2 \]  \tag{50}


\[ f = \frac{pN}{2\pi} \]  \tag{51}


\[ P_{ir} = q_l \times \left(\frac{f}{f_1}\right)^{1.5} \times \left[M_{cs} \times \left(\frac{B_a}{B_1}\right)^2 + M_{ds} \times \left(\frac{B_a}{B_1}\right)^2\right] \]  \tag{52}


\[ \eta = \frac{T_m \times N - P_{nec}}{T_m \times N + P_{cu} + P_{ir}} \]  \tag{53}


3.3. Problem formulation

In [53], a BLDC motor benchmark was presented, and
the basic code in MATLAB is publicly available to compu-
te the fitness functions. The studied problem has a
MATLAB model that can be used for research rec-
ommendations 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 prob-
lem. 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 vari-
ables, such as back iron \(B_{cs}\), magnetic induction both
in the teeth \(B_{t}\), the current density in the conductors
\(\delta\), magnetic induction in the air-gap \(B_a\), and bore sta-
tor diameter \(D_s\), and six inequality constraints, such as
temperature \(T_a\), magnetic maximum current \(I_m\), inner
radius \(D_i\), external diameter \(D_{ext}\), total mass \(M_{Tt}\),
and the determinant \((\text{Discr})\) 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 con-
sidered. The thermal model in this method is very
straightforward since the model is for pre-sizing con-
siderations only. It is also presumed that the thermal
conduction resistance is smaller than the thermal con-
vection 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

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

\[ \frac{(B_c - B_r(1 + \alpha_c \times T_a))}{\mu_a \times \mu_o} \times h_a + \frac{z \times I_{\text{max}}}{4\rho} + \frac{B_c}{\mu_o} \times \frac{\beta}{\alpha} \times \left(1 + \frac{2e}{D_r}\right) \times r_s \times k_f \times e = 0 \]  \tag{48}

\[ \text{The electrical loss can be found by defining the phase resistance and is given in Eq. 49.} \]

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

\[ P_{cu} = 2 \times R_{ph} \times I^2 \]  \tag{50}

\[ f = \frac{pN}{2\pi} \]  \tag{51}

\[ P_{ir} = q_l \times \left(\frac{f}{f_1}\right)^{1.5} \times \left[M_{cs} \times \left(\frac{B_a}{B_1}\right)^2 + M_{ds} \times \left(\frac{B_a}{B_1}\right)^2\right] \]  \tag{52}

\[ \eta = \frac{T_m \times N - P_{nec}}{T_m \times N + P_{cu} + P_{ir}} \]  \tag{53}

\[ \text{The thermal model is straightforward. Conversely, it is assumed that thermal conduction resistances are much lesser than thermal convection resistances. As a result, the temperature of all stator parts in touches, such as the stator yoke, teeth, and coils, is the same. However, even if the rotor and stator are not in touch due to rolling, large areas are in touch with air space. The confined air is represented as a substance with low thermal conductivity under these circumstances. Despite this, it is low} \]
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 |
|                 | $B_{st}$ (T) | Mean flux density in stator back iron | 1.6 | 0.6 |
|                 | $B_{r}$ (T) | Airgap flux density | 0.76 | 0.5 |
|                 | $B_{t}$ (T) | Mean flux density in teeth | 1.8 | 0.9 |
|                 | $\delta$ (A/mm²) | Winding current density | 5 | 2 |
| Inequality Constraints | $T_o$ (°C) | 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 |
| Objective variables | $M_t$ (Kg) | Total motor mass | $\leq$ 15 |
|                 | $\eta$ (%) | Motor efficiency | - |

The control parameters of various algorithms are pre- 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.

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

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.

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

\[
\text{Spread} = \sqrt{\frac{1}{n} \sum_{i=1}^{a} \max(d(a_i, b_i))} \quad (59)
\]

\[
\text{Generational Distance (GD)} = \sqrt{\frac{\sum_{i=1}^{n_o} d_i^2}{n}} \quad (60)
\]

\[
\text{Inverted Generational Distance (IGD)} = \frac{\sqrt{\sum_{i=1}^{n} (d'_i)^2}}{n} \quad (61)
\]

\[
\text{Diversity Metric (DM)} = \left| \frac{d_i + d_m + \sum_{i=1}^{n} |d_i - d|}{d_i + d_m - (n-1)d} \right| \quad (62)
\]

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. $n_t$ denotes the number of true Pareto-optimal solutions, $n_o$ denotes the number of true Pareto set (PS), $d$ 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 $\min_j |f_j(x) - f_j(x)| + |f_j(x) - f_j(x)|$ for $i, j = 1, 2, \ldots, n$, $d$ denotes the average of $d_i$, $d_i$ and $d'_i$ specifies the Euclidean distance, and $b_l$ and $b_r$ 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 CONST, 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.

**CONST :**

\[
\begin{align*}
\text{Minimize } f_1(x, y) &= x \\
\text{Minimize } f_2(x, y) &= (1 + y)/x \\
g(1) &= 0.1 \leq x \leq 1 \\
g(2) &= 0 \leq y \leq 5 \\
g(3) &= 0 \leq 6 - y - 9x \\
g(4) &= 0 \geq 1 + y - 9x
\end{align*}
\]

**SRN :**

\[
\begin{align*}
\text{Minimize } f_1(x, y) &= (x - 2)^2 + (y - 1)^2 + 2 \\
\text{Minimize } f_2(x, y) &= 9x - (y - 1)^2 \\
g(1) &= -20 \leq x \\
g(2) &= y \leq 20 \\
g(3) &= 0 \geq x^2 + y^2 - 225 \\
g(4) &= 0 \geq x - 3y + 10
\end{align*}
\]

**BNH :**

\[
\begin{align*}
\text{Minimize } f_1(x, y) &= 4x^2 + 4y^2 \\
\text{Minimize } f_2(x, y) &= (x - 5)^2 + (y - 5)^2 \\
g(1) &= 0 \leq x \leq 5 \\
g(2) &= 0 \leq y \leq 3 \\
g(3) &= 0 \geq (x - 5)^2 + (y)^2 - 25 \\
g(4) &= 0 \geq - (x - 8)^2 + (y + 3)^2 + 7.7
\end{align*}
\]

**TNK :**

\[
\begin{align*}
\text{Minimize } f_1(x, y) &= x \\
\text{Minimize } f_2(x, y) &= y \\
g(1) &= 0 \leq x \\
g(2) &= y \leq \pi \\
g(3) &= 0 \geq -x^2 - y^2 + 1 + 0.1* \cos(16\arctan(\frac{x}{2})) \\
g(4) &= \frac{1}{2} \geq (x - \frac{1}{2})^2 + (y - \frac{1}{2})^2
\end{align*}
\]

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 CONST, 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-ZDT4 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| 0.0034089| 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| 0.0030618| 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 | 0.033671| 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
The MOCHIO algorithm acquired the best GD values for ZDT problems, demonstrating the MOEIO'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 is well-distributed.
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 ($\{x_1, x_2, \ldots, x_m\}$), $g_j(X) \leq 0$ denotes the inequality constraints ($j = 1, 2, \ldots, x$), $h_j(X) = 0$ denotes the equality constraints ($j = x + 1, \ldots, y$), and $\delta$ denotes the tolerance of equality constraint.

Figure 9. PF obtained by all algorithms for ZDT problems
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.

\[
\mu_j = \begin{cases} 
1, & f_j^i \leq f_{lb}^i \\
\frac{f_{ub}^j - f_j^i}{f_{ub}^i - f_{lb}^i}, & f_{lb}^i \leq f_j^i \leq f_{ub}^i \\
0, & f_j^i > f_{ub}^i 
\end{cases}
\]  

(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).

\[
\mu_j = \frac{\sum_{i=1}^{N} \mu_{ji}}{\sum_{j=1}^{M} \sum_{i=1}^{N} \mu_{ji}}
\]  

(69)

where \( M \) signifies the NDSs. The BC result is the one with the high value of \( \mu_j \).

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.
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 $B_{c3}$, $B_{d3}$, $\delta$, $B_e$, and $D_i$ 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 ($\Delta$), Epsilon ($\epsilon$), 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 $\Delta$, $\epsilon$, 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, MOBAT, MOWOA, MOPSO, MOGWO, MOMFO, MOWOA, MOPSO, MOGWO, MOMFO, MOBA.

![Figure 9. Continued.](image-url)
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) | $\delta$ (A/mm²) | $M_T$ (Kg) | $\eta$ (%) |
|---------|-----------|-----------|-------------|-----------|-----------------|------------|-----------|
| $f_1$   | 1.7999    | 0.6640    | 1.6000      | 0.1860    | 3670737.688     | 10.6213    | 93.60     |
| $f_2$   | 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) | $\delta$ (A/mm²) | $M_T$ (Kg) | $\eta$ (%) |
|---------|-----------|-----------|-------------|-----------|-----------------|------------|-----------|
| $f_1$   | 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) | $\delta$ (A/mm²) | $M_T$ (Kg) | $\eta$ (%) |
|---------|-----------|-----------|-------------|-----------|-----------------|------------|-----------|
| $f_1$   | 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     |
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) \( Bcs \), (d) \( D_s \), (e) \( \delta \)
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 $\Delta$, $\varepsilon$, 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
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/9780847358349.

[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/IREV.2011.4104.

[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 EL, 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/CPIE2014.0449.CP.2014.0449.

[8] dos Santos Coelho L, Barbosa LZ, Lebensztain 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.2010.204415.

[9] Bora TC, Coelho LDS, Lebensztain 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] Nagraha 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):100516–1–24. 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, CECF. 2006, p. 238, 2006, doi:10.1109/CECF.2006.1635028.

[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, ICIS. 2007, pp. 854–858, doi:10.1109/ICIS.2007.4685808.

[19] Sidwan 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, ISISITA 2017 - Proceeding, vol. 2017-January, pp. 193–198, Nov. 2017, doi:10.1109/ISISITA.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.ASSOC.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.EESWA.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
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/IAE-2007-913.

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.

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.

Premkumar M, Sowmya R, Jangir P, et al. MOSMABA: multi-objective slime mould algorithm based on elitist Non-dominated sorting. IEEE Access. 2021;9:3229–3248. doi:10.1109/ACCESS.2020.3047936.

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.

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.

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.

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.

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.

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.

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.

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.

Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput. 1997;1(1):67–82.

Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput. Apr. 2002;6(2):182–197. doi:10.1109/4235.996017.

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:10.1111/j.2175-2745.2013.04103.10.1109/S11431-013-5410-3.

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.

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.

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.

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.

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.

Bianchi N, Bolognani S. “Brushless DC motor design: An optimization procedure based on genetic algorithms,” IEEE Conference Publication, pp. 16–20, 1997, no. 444. doi:10.1049/CP:19971029.

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.

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:10.1108/033216405105610292.

Brisset S, Brochet P. “A Benchmark for a Mono and Multi-Objective Optimization of the Brushless DC Wheel Motor,” L2EP, Ecole Centrale de Lille, Villeneuve d’Ascq, France. http://l2ep.univ-lille1.fr/come/benchmarks/brushlessWheel-motor.htm (accessed Sep. 06, 2020).

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.

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/s10489-017-0741-1.

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.