NMOPSO: AN IMPROVED MULTIOBJECTIVE PSO ALGORITHM FOR PERMANENT MAGNET MOTOR DESIGN

Lassaad ZAARAOUI[1], Ali MANSOURI[2], Nadia SMAIRI[3]

This work deals with the investigation of a new multiobjective particle swarm optimization (PSO) algorithm and the use of this algorithm for the optimization of the geometry of an in-wheel motor. The proposed algorithm introduces a new strategy to avoid the fall into local optima. This strategy proposes new parameters and adjusts the relationship between certain other parameters to increase the use of particle information and the convergence precision. Through a multiobjective optimization problem formulated from the analytical model of the studied motor, the proposed algorithm has been compared with two other multiobjective PSO algorithms and validated using the finite element analysis.

Keywords: multiobjective optimization, particle swarm optimization, permanent magnet motor, finite element analysis

1. Introduction

Particle Swarm Optimization (PSO) is a stochastic optimization algorithm for nonlinear problems developed by Eberhart and Kennedy in 1995 [1]. It is based on a set of particles modeled as vectors and originally arranged randomly. These particles move in the search space to find a global optimal solution. Each particle constitutes a potential solution, and it has a position vector inside the search space and a velocity vector which will be used to calculate the next position. Within the swarm, a neighborhood is determined for each particle as the subset of particles that can communicate. Moreover, each particle has a memory which contains the best position found by itself (personal leader), and the best position reached by the particles belonging to the neighborhood (global leader) [2]. The displacement of a particle is affected by three tendencies. Indeed, the particle tends either to follow its own path, or to return to its best position already found, or to follow the best position found by the neighborhood. Based on the

[1] PhD student, Laboratory of Computer, Electronics & Smart Engineering Systems Design, National Engineering School of Sfax, Tunisia. e-mail: lassaad.zaaraoui@gmail.com
[2] Lecturer, Higher School of Applied Sciences and Technology of Gafsa, University of Gafsa, Tunisia. e-mail: ali.mansouri@isetgf.rnu.tn
[3] Associate Professor, COSMOS Laboratory, National School of Computer Sciences, University of Manouba, Tunisia. e-mail: nadia.smairi@gmail.com

memory information, the displacement of the particle is a compromise between these three tendencies [3].

In a D-dimensional search space, we assume that N particles iterate T times. At each time step t, the i-th particle is determined in the d-dimensional position with x_{id}, the velocity is recorded by v_{id}, the best visited position of the particle (personal leader) is expressed by p_{id}, and the best solution found by its neighborhood (global leader) is recorded with g_d [2]. The general displacement of the particles is done by the following two equations [4]:

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (g_d - x_{id}) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2)$$

where ω is the inertia factor which allows to control the displacement of the particle between the exploitation and the exploration in the search space, c_1 and c_2 are learning factors, r_1 and r_2 are random numbers which follow a probabilistic law on [0..1].

The PSO algorithm is the most widely used in the optimization of electric machines. In [5] the PSO algorithm was used to determine the optimal values of the design variables of a reluctance machine. The optimization process was carried out according to two criteria, which are: the minimization of the torque ripple and the maximization of the output power. It has been shown that the algorithm makes it possible to find a compromise between the two conflicting objectives, and to obtain the optimal variables that meet the design requirements.

The optimization of a surface permanent magnet synchronous machine was performed in [6] using the PSO algorithm. The aim of this study was to optimize the parameters of the winding structure to decrease the copper losses and reduce the operating temperature. The authors were illustrated that the optimization results reduced the copper losses from 0.719 W to 0.468 W, thereby solving the problem of motor heating.

In [7], a detailed study of a permanent magnets machine for an in-wheel motor has been presented. Two multiobjective PSO algorithms were used to determine the optimal geometry that maximize the efficiency and minimize the mass of the machine. It has been shown that the two algorithms provide optimal and achievable solutions in a short execution time. With the final aim of confirming the efficiency of the used algorithms, the optimization results were validated by the finite element method.

In [8], the Optimizer Multiobjective PSO (OMOPSO) [9] and the Speed-constrained Multiobjective PSO (SMPSO) [10] algorithms were used to find the optimal design of an in-wheel permanent magnet motor for an electric vehicle. The optimization process has been carried out according to three objectives: The first is the efficiency maximization, the second is the mass minimization, and the third is the torque ripple minimization. After several executions, it has been shown that the two PSO algorithms can find a multitude of achievable solutions. However, the OMOPSO algorithm can generate a well-diversified Pareto front that contains more solutions than the SMPSO algorithm. Moreover, a comparison using the covariance metric showed that there are more solutions of the OMOPSO algorithm that dominate the SMPSO solutions. The same comparison was made in [11] using only the first two objective functions. The comparison showed that the OMOPSO algorithm is somewhat better than the SMPSO algorithm, according to the covariance metric.

Compared to the previous works [7] [8] [11] [12] [13] [14], the novelty of this work is to propose a new multiobjective particle swarm optimization algorithm. It is based on the OMOPSO algorithm. The ultimate goal is to improve the diversity and the convergence accuracy of the OMOPSO algorithm. The new algorithm will also be applied for the optimal design of an in-wheel motor. Indeed, this work also presents the analytical model of a new topology of the in-wheel motors intended for electric vehicles. This topology adopts the unequal stator teeth and the concentrated windings single layer to improve the motor performance.

The remainder of this paper is structured as follows: The OMOPSO algorithm is reviewed in section 2. In section 3, we present our new algorithm. The analytical model of the in-wheel motor and the optimization problem are described in the section 4. The optimization results of the new algorithm and two other multiobjective PSO algorithms are presented in section 5. The comparison of the algorithms and the results discussion are respectively presented in sections 6 and 7. Finally, we finish by the conclusion.

2. Multiobjective Particle Swarm Optimization (OMOPSO)

Based on Pareto dominance, crowding distance and two different mutation operators, the OMOPSO algorithm [9] was invented in 2005 by Sierra and Coello. This algorithm works as follows: first, after the initialization of the positions and velocities of the particles in the search space, the approach consists of storing the non-dominated particles in an external archive as leaders. Then, the crowding distance of each leader is calculated. After that, at each iteration and for each particle, the approach consists of randomly selecting a global leader from the archive, changing the particle position using equations (1) and (2), and applying a mutation operator that corresponds to this particle. Indeed, the OMOPSO algorithm subdivides the swarm into three sub-parts of equal size. The particles of the first part will have no mutation at all, and the particles of the second and third sub-parts will have a uniform mutation, and a non-uniform mutation, respectively. This mutation technique makes it possible to adjust the exploitation and exploration of the research space and to preserve the diversity. The process continues by evaluating the particle and replacing the best visited solution if it is dominated by the new particle. After updating all the particles, all the global leaders are also updated. The storage of leaders in the external archive depends on the crowding distance. This criterion is used to avoid the explosion of the external archive when its maximum size is exceeded. Finally, after repeating these steps for a given number of generations, the final archive is returned as the search result.

3. Improvement of OMOPSO algorithm

3.1 Limits of OMOPSO algorithm

In the OMOPSO algorithm [9], the particle displacement is updated with the same iterative formula using the inertia factor (ω) and the factors $\varphi_1 = r_1.c_1$ and $\varphi_2 = r_2.c_2$ to control the behavior of the particles. However, the evolution of the particle velocity with the same formula minimizes the difference between the particles, thereby minimizing the exploration capacity of the search space. Moreover, in some execution points, the velocity values of a particle change from a high value to a low value. Therefore, the particles come very close to their extreme values. These irregular movements affect the search performance of the algorithm because it is easy to get trapped in local optima. To remedy this problem, the OMOPSO algorithm uses a mutation technique to avoid the swarm stagnation in local solutions. Indeed, the general idea is to move the swarm away from a crowded location by the mutation of a set of particles according to a certain probability. If one of these mutated particles becomes the new best global guide g_d, it is therefore possible to escape the traps of local optima and the stagnation of the swarm. On the other hand, the authors of the SMPSO algorithm have been proposed a restriction coefficient to regulate the velocity of particles in OMOPSO and to achieve the high search performance of the algorithm [10]. However, the SMPSO algorithm does not allow to generate a very rich front of solutions [8].

3.2 New Multi-objective Particle Swarm Optimization (NMOPSO)

To improve the convergence accuracy y, a third coefficient $\varphi_3 = r_3.c_3$ in the velocity update of the single-objective PSO algorithm were introduced in [15]. With this coefficient, the particle information can be used more effectively to avoid local optima and improve the diversity of particles. The new update equation is [15]:

$$v_{id}(t+1) = \begin{cases} \omega_d(t)v_{id}(t) + c_1r_1(p_{id} - x_{id}) + c_2r_2(g_d - x_{id}) & \text{if } rand() \geq \eta \\ \omega_d(t)v_{id}(t) + c_1r_1(p_{id} - x_{id}) + c_2r_2(g_d - x_{id}) \\ + c_3r_3(n_{id} - x_{id}) & \text{if } rand() < \eta \end{cases} \quad (3)$$

At each step time, and according to a certain probability, the update of the particle velocity is done using a contemporary leader n_{id}. In our new multiobjective algorithm, this contemporary leader is initialized by comparing two solutions of the external archive according to the crowding distance. The best solution between them will be the contemporary leader. After that, the updating of this contemporary leader is done by choosing a particle of the whole swarm. If this particle dominates the contemporary leader, the update is then performed.

In the OMOPSO algorithm, the values of ω, c_1 and c_2 are independently calculated. The inertia factor takes random values in the interval [0.1, 0.5]. Moreover, the self-cognition c_1 and the social guidance c_2 take random values in the interval [1.5, 2]. This random guidance can converge the particles towards local optima or slow down the convergence. For example, if the learning factors are very small, the particle memory information will not be fully used. Conversely, if c_1 is large, the particles tend to exploit their own range, which slows down the search speed. If c_2 is large, the particles may fall into local optima at an initial stage.

To overcome these problems, an evolutionary strategy must be used to adjust the values of these factors. In the early stages of research, this strategy adjusts the self-cognition factor to a great value, and the social guidance to a small value, which will encourage the particle to search around itself when the research space is very large, thereby improving the particles diversity. At later points of research, the self-cognition factor must be adjusted to small value and the social guidance to a great value, which will increase the particle displacement when the research space is small, thereby improving the precision of the algorithm and the convergence towards the global optimal solutions. Moreover, the adaptive inertia factor technique is used. The inertia factor ω and the learning factors c_1, c_2 and c_3 are expressed by the following equations [15]:

$$\omega_d(t) = \begin{cases} \omega_{\max} & \text{if } x_{id}(t) > x_{mean} \\ \omega_{\min} - \dfrac{(\omega_{\max} - \omega_{\min}) \times (x_{id}(t) - x_{\min})}{(x_{mean} - x_{\min})} & \text{if } x_{id}(t) \le x_{mean} \end{cases} \tag{4}$$

$$c1 = 0.5 + 2 \times \cos\left(\frac{\pi \times (t-1)}{2 \times (T-1)}\right) \tag{5}$$

$$c2 = 0.5 + 2 \times \sin\left(\frac{\pi \times (t-1)}{2 \times (T-1)}\right) \tag{6}$$

$$c3 = c1 \tag{7}$$

We can notice that the new algorithm pays more attention to the particle at an early stage. Then, NMOPSO highlights the intelligence of the whole swarm at a later stage. The third social orientation factor c_3 takes on the same value as the self-cognition of the particle. The principle of the new multiobjective PSO algorithm (NMOPSO) is presented by the following pseudocode:

Pseudocode of NMPSO Algorithm

```
Initialize the swarm (particles)
Evaluation (particles)
Initialize the personal leaders of particles
Initialize the global leaders in the external archive
Crowding distance (global leaders)
Initialize the contemporary leaders of particles
Repeat
    For each particle do
        Choose the global leader
        Update the speed and the position (Eqs. 3-7 and Eq. 2)
        Perform the mutation
        Evaluate the particle
        Update the personal leader
    End for
    Update the contemporary leaders of particles
    Update the global leaders
    Send global leaders to the external archive
    Crowding distance (global leaders)
Until (Maximum number of iterations reached)
Return the external archive
```

4. Formulation of the optimization problem

To test the effectiveness of the NMPSO proposed in this paper, this algorithm is applied to a multiobjective optimization problem formulated from the analytical model of the studied machine. Indeed, the optimization process consists to find the optimal geometry of an in-wheel permanent magnets motor intended for an electric vehicle. In this section, the dimensions parameters of this machine are given as they present the decision variables of our optimization problem. Then, the magnetic properties and the copper and iron losses, necessary for the optimization process are presented. Finally, the objective functions are defined.

4.1 Decision variables

The studied machine of the in-wheel motor is a 3-phase, 18-slots, 16-poles, surface-mounted permanent magnets machine with an external rotor and unequal stator teeth. The sizing parameters of this machine consider the decision variables of our optimization problem. They are given in Table 1 with their symbols and ranges of values. The geometric dimensions are shown in Fig. 1.

Table 1

Sizing parameters

Parameters	Symbols	Bounds
The air gap length	δ	[1 ; 2]
The rotor inner diameter	D_{rint}	[225 ; 230]
The magnet height	h_m	[2 ; 4]
The wider stator tooth width	b_{ts1}	[21 ; 27]
The lower stator tooth width	b_{ts2}	[15 ; 17]
The stator slot height	h_s	[30 ; 40]
The axial machine length	l	[70 ; 85]
The half pole angle	α	[19 ; 21]
The stator inner diameter	D_{sint}	[115 ; 122]

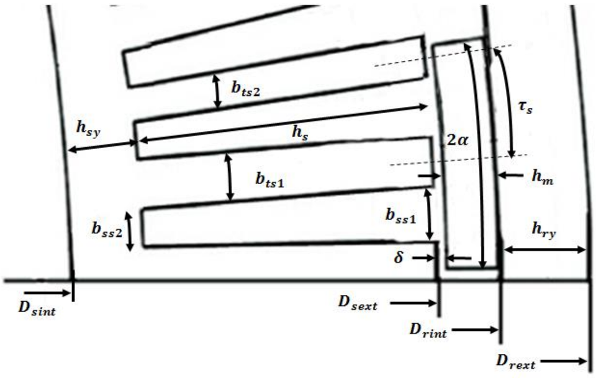

Fig. 1. Geometric dimensions of the machine

4.2 Magnetic proprieties

In the present section, we describe the major magnetic flux densities in all machine parts. Indeed, the analytical calculations of the magnetic proprieties are expressed as follows:

The amplitude of the fundamental component of the air gap flux density is given by [16]:

$$B_{d1} = \frac{4}{p} B_m \sin(a) \tag{8}$$

Where B_m is the mean air gap flux density, and α is the half pole angle.

The rotor yoke flux density is determined by [12]:

$$B_{ry} = \frac{B_m t_m}{2h_{ry} k_j} \tag{9}$$

Where τ_m is the slot pitch, h_{ry} is the rotor yoke height, and k_j is the lamination stacking factor.

The stator yoke flux density is expressed by [17]:

$$B_{sy} = \frac{B_m \tau_m}{2h_{sy}k_j} \tag{10}$$

Where h_{sy} is the stator yoke height.

The flux densities in the widest and lowest teeth are respectively given by the following equations [18] [19]:

$$B_{st1} = \frac{B_m \tau_s (l + 2\delta)}{b_{ts1}k_j l} \tag{11}$$

$$B_{st2} = \frac{B_m \tau_s (l + 2\delta)}{b_{ts2}k_j l} \tag{12}$$

Where τ_s is the slot pitch, l is the axial machine length, δ is the air gap length, b_{ts1} is the width of the widest stator tooth, and b_{ts2} is the width of the lowest stator tooth.

4.3 Copper and iron losses

The copper losses in the stator windings are expressed as [20]:

$$P_{co} = 3I_1^2 R_{ph} \tag{13}$$

Where I_1 is the rms stator current and R_{ph} is the winding resistance of a phase.

The iron losses in the rotor yoke (P_{ry}), the stator yoke (P_{sy}), the wider stator teeth (P_{st1}), and the lower stator teeth (P_{st2}) are expressed respectively by the following expressions [20]:

$$P_{ry} = \left(k_{hyst}B_{ry}^{\beta} f + k_{eddy}B_{ry}^{2} f^2 + 8.67k_{exc}B_{ry}^{1.5} f^{1.5}\right)V_{ry} \tag{14}$$

$$P_{sy} = \left(k_{hyst}B_{sy}^{\beta} f + k_{eddy}B_{sy}^{2} f^2 + 8.67k_{exc}B_{sy}^{1.5} f^{1.5}\right)V_{sy} \tag{15}$$

$$P_{st1} = \left(k_{hyst}B_{st1}^{\beta} f + k_{eddy}B_{st1}^{2} f^2 + 8.67k_{exc}B_{st1}^{1.5} f^{1.5}\right)V_{st1} \tag{16}$$

$$P_{st2} = \left(k_{hyst}B_{st2}^{\beta} f + k_{eddy}B_{st2}^{2} f^2 + 8.67k_{exc}B_{st2}^{1.5} f^{1.5}\right)V_{st2} \tag{17}$$

Where Where k_{hyst} is the hysteresis coefficient, β is the Steintmetz constant, f is the electrical frequency, k_{eddy} is the eddy current coefficient, k_{exc} is the excess eddy current loss coefficient, V_{ry} is the rotor yoke volume, V_{sy} is the stator yoke volume V_{st1} is the wider teeth volume and V_{st2} is the lower teeth volume.

4.4 Objective functions

The optimization procedure is performed according to three objective functions. Indeed, the maximization of the machine efficiency presents the first objective function, and the minimization of the mass and the torque ripple are the second and third objective functions, respectively. We can formulate the first objective to be minimized in order to decrease the complexity of the problem. With (η) is the machine efficiency, the minimization of (1-η) is identical to the maximization of (η). The three objectives' functions are determined by the following equations:

$$objective\_function1 = \text{minimize}(1-\eta) \tag{18}$$

Where (η) is the machine efficiency. It is determined according to the output power, (P_{out}), and the copper and iron losses as follow:

$$\eta = \frac{P_{out}}{P_{out} + P_{co} + P_{ry} + P_{sy} + P_{st1} + P_{st2}} \tag{19}$$

$$objective\_function2 = \text{minimize}(mass) \tag{20}$$

$$objective\_function3 = \text{minimize}(T_{rip}) \tag{21}$$

Where (T_{rip}) is the torque ripple, and given by:

$$T_{rip} = 2\frac{\sqrt{\left(\hat{E}_7 - \hat{E}_5\right)^2 - \left(\hat{E}_{13} - \hat{E}_{11}\right)^2 - \left(\hat{E}_{19} - \hat{E}_{17}\right)^2 - \left(\hat{E}_{25} - \hat{E}_{23}\right)^2}}{\hat{E}_1} \tag{22}$$

Where $\hat{E}_i$ is the i[th] harmonic of the induced electromotive force.

5. Optimization results

To show the effectiveness of the improved algorithm NMOPSO, it has been applied and compared with the OMOPSO and SMPSO algorithms on the considered machine sizing problem. For the computation, we have used, as environment, the "Eclipse" platform and the "jMetal" framework which is an object-oriented Java-based framework for multiobjective optimization. The Pareto fronts of these algorithms are shown in Fig. 2, Fig. 3, and Fig. 4.

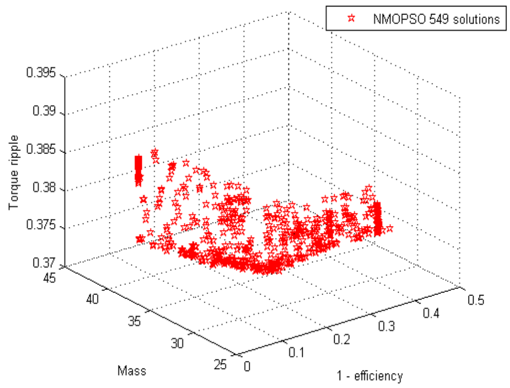

Fig. 2. NMOPSO Pareto front

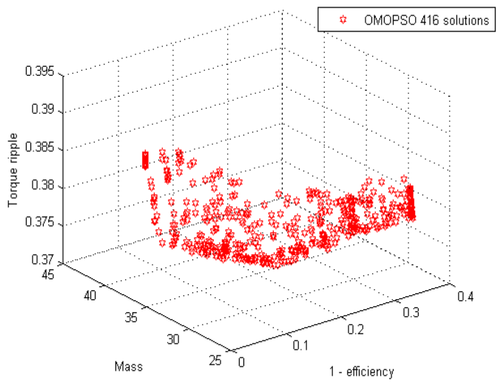

Fig. 3. OMOPSO Pareto front

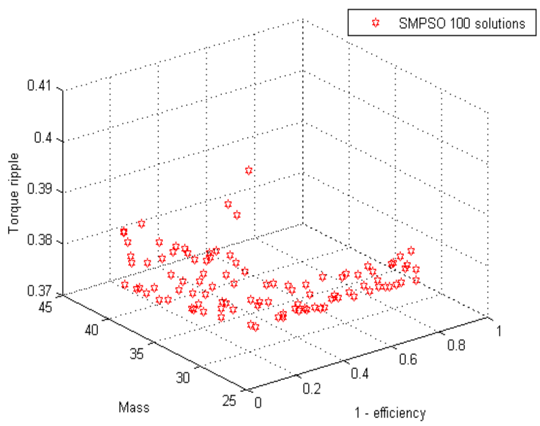

Fig. 4. SMPSO Pareto front

The objective functions used in this optimization procedure are contradictory since the increase in the efficiency of the machine also causes the increase in its mass and its torque ripple, and vice versa. Despite that, we can notice that all the applied algorithms are able to find a compromise between the contradictory objectives and thus generate optimal Pareto fronts. Indeed, all the produced solutions are feasible since they respect the machine requirements.

In this regard, the designer must make a compromise between the objective functions to determine the solution that meets the design requirements of the machine. The difference between these Pareto fronts can be noticed by the number of solutions generated by each algorithm. Indeed, by referring to these Pareto fronts, we can still report that the NMOPSO algorithm has the richest Pareto front since it allows to generate more than 500 solutions, well diversified on the research space. The OMOPSO algorithm generates only more than 400 solutions and the SMPSO algorithm loses its diversity and generates only 100 solutions at each execution. We have chosen the best optimal sizing of the studied machine among the solutions generated by each algorithm. The results are presented in Table 2.

Table 2

Optimization results

	NMOPSO	OMOPSO	SMPSO
δ	1	1	1
D_{rint}	230	227.66	229.85
h_m	2.02	2.45	2.38
b_{ts1}	21.05	21.08	21.01
b_{ts2}	17	15.37	16
h_s	35.75	36.53	36.16
l	81.92	75.70	77.50
α	19.0	19.0	19.0
D_{sint}	121.75	117.64	118.50
Efficiency	88.56%	87.31%	88.13%
Mass (kg)	31.98	32.41	32.63
Ripple torque	0.384	0.375	0.375
CPU (s)	4.960	4.424	5.252

The results show that the performance of the NMOPSO algorithm outperforms the other algorithms with regard to the efficiency and mass of the machine. Moreover, this algorithm has the best convergence accuracy with a fast execution time.

6. Performance Measurement

To measure the performance of the NMOPSO algorithm, the "measurement instruments" must be available. In this paper, we adopted two comparison metrics which are the covariance metric and the hypervolume metric. The covariance metric is a relative metric that makes it possible to compare two fronts A and B according to the dominance of Pareto. Indeed, the value C (A, B) allows to calculate which portion of the front A dominates the front B. the calculation of this ratio is carried out according to the following equation:

$$C(A,B) = \frac{card\left(\{y \in B; \exists x \in A \mid x \text{ dominates } y\}\right)}{card\left(B\right)} \tag{23}$$

Thus, C (A, B) =1 implies that the front B is totally dominated by A. Conversely, C (A, B) = 0 implies that none of the points of B is dominated by a point of A. Therefore, the more the value of C (A, B) is closer to 1, the more the front A is better with respect to B. Since this metric is not symmetrical, it is necessary to consider C (A, B) and C (B, A) to obtain a more reliable measure of the two compared fronts. Each algorithm has been executed ten times. At each execution, the three algorithms have been compared. The average values of the covariance metric are illustrated in Table 3.

Table 3

Covariance metric

C(column, row)	NMPSO	OMOPSO	SMPSO
NMPSO	*	0,2449	0,0804
OMOPSO	0,3549	*	0,0917
SMPSO	0,5836	0,5666	*

The results show that NMOPSO is the best algorithm according to the Pareto dominance. Indeed, the NMOPSO and OMOPSO algorithms are better than the SMPSO algorithm, and there are more portions of NMOPSO that dominate the OMOPSO algorithm.

The second metric is the hypervolume indicator. It measures the hypervolume of the portion of the objectives space that is dominated by the found Pareto front. It is an indicator to be maximized. The average values of ten comparisons are shown in Table 4.

Table 4

Hypervolume indicator (HI)

	NMOPSO	OMOPSO	SMPSO
HI	7,75403313495	7,73851478683	7,72883901804

The results show that the NMOPSO algorithm is better than the other two algorithms, since it has the highest value of the hypervolume indicator.

7. Validation of NMOPSO results

To validate the NMOPSO optimization results, the studied machine has been analyzed using the finite element method. The FEA results are presented in Fig.5, Fig. 6 and Table 5.

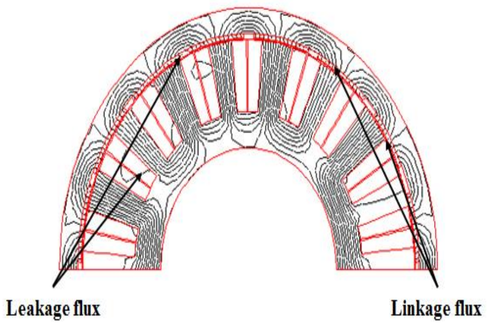

Fig. 5. 2-D Magnetic flux lines

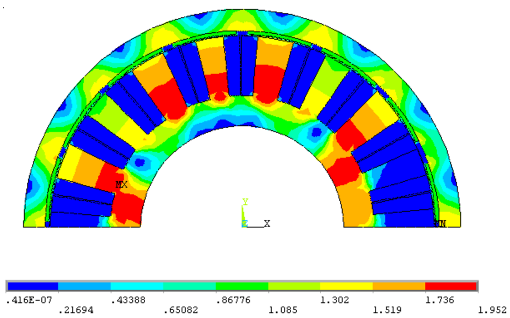

Fig. 6. Magnetic flux density distribution

By referring to Fig. 5, we clearly notice the flux lines which do not follow the intended path and do not cross the air gap. These magnetic flux lines are called the leakage flux. Fig. 6 shows the distribution of the magnetic flux density of the studied machine.

Table 5

FEA results

Parameter	NMOPSO results	FEA results	Error
The fundamental air gap flux density $B_{\delta,1}$	0.91	0.96	5%
The rotor yoke flux density B_{ry}	1.17	1.31	10%
The stator yoke flux density B_{sy}	0.95	1.02	10%
The widest stator tooth flux density B_{st1}	1.48	1.4	5%
The lowest stator tooth flux density B_{st2}	1.84	1.54	16%

We can clearly identify some magnetic saturations in the stator teeth and some small areas in the stator yoke. According to Table 5, we can notice that there is a good agreement between the FEA and the NMPSO results, which make the improved multiobjective algorithm (NMOPSO) a valuable tool for the optimization of complex real problems, in particular for the optimal design of permanent magnet machines.

8. Conclusion

In the present work, a new swarm particle optimization technique (NMOPSO) has been proposed. This improved multiobjective PSO algorithm introduces new parameters and adjusts the relationship between various parameters. The new algorithm has been evaluated using multiobjective optimization problem for the design of an in-wheel motor. Then it has been compared with the standard OMOPSO and SMPSO algorithms using the covariance metric and the hypervolume indicator. The results shown that the NMOPSO algorithm overcomes the performance in both comparison metrics. Indeed, it has the best convergence accuracy and the richest Pareto front. The optimization results of the NMOPSO algorithm have been also compared with the FEA results of the studied machine. This comparison shows that the proposed NMOPSO is a valuable tool for the design of permanent magnet motors.

R E F E R E N C E S

[1]. *R.C. Eberhart, J. Kennedy,* “A new optimizer using particle swarm theory,” Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan, 1995, pp. 39–43.

[2]. *D. Bratton, J. Kennedy,* “Defining a Standard for Particle Swarm Optimization,” Proceedings of the 2007 IEEE Swarm Intelligence Symposium, pp.120–127.

[3]. *M. Clerc, P. Siarry,* “Une nouvelle métaheuristique pour l'optimisation difficile : la méthode des essaims particulaires ”, 2004. http://www.particleswarm.info

[4]. *M.R. Sierra, C.A.C. Coello,* “Multiobjective Particle Swarm Optimizers: A survey of the State-Of-the-Art,” International Journal of Computational Intelligence Research, 2006, pp. 287–308.

[5]. *R. Banerjee, P. Sensarma,* "Design Optimization of a Switched Reluctance Machine using Analytical model and PSO," Proceedings in the 45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, 2019.
[6]. *O. Dal, M. Yildirim, H. Kurum,* "Optimization of Permanent Magnet Synchronous Motor Design by Using PSO," Proceedings in 4th International Conference on Power Electronics and their Applications, Elazig, Turkey, 2019.
[7]. *L. Zaaraoui, A. Mansouri,* "Optimization and finite element analysis of an in-wheel permanent magnet motor" Malaysian Journal of Fundamental and Applied Sciences, **vol. 17**, 2021, pp. 104–108.
[8]. *L. Zaaraoui, A. Mansouri,* "Multiobjective optimal design of an in-wheel permanent magnet synchronous motor for an electric vehicle" Journal of Automation & Systems Engineering, **vol. 13**, 2019, pp. 29–43.
[9]. *M.R. Sierra, C.A.C. Coello,* "Improving PSO-based multi-objective optimization using crowding, mutation and ε-dominance," Proceedings In third International Conference on Evolutionary Multi-Criterion Optimization, EMO, Guanajuata, 2005, pp. 505–519.
[10]. *A.J. Nebro, et al.,* "SMPSO: A new PSO-based metaheuristic for multi-objective optimization," Proceedings of the IEEE Symposium on Computational Intelligence in Multi-criteria Decision-Making, 2009, pp. 66–73.
[11]. *L. Zaaraoui, A. Mansouri, H. Trabelsi,* "PSO-Based Optimal Design of In-Wheel Permanent Magnet Motor," The 2017 International Conference on Internet of Things, Embedded Systems and Communications, Gafsa, Tunisia, 2017.
[12]. *L. Zaaraoui, A. Mansouri, H. Trabelsi,* "Particle swarm-based optimization of an in wheel permanent magnet motor," Proceedings in the 14th International Multi-Conference on Systems, Signals & Devices (SSD), Marrakech, Morocco, 2017.
[13]. *L. Zaaraoui, A. Mansouri,* "Design improvement of an in-Wheel machine for electric vehicle based on multiobjective algorithms", Proceedings of the International Conference on Recent Advances in Electrical Systems, Hammamet, Tunisia, 2018.
[14]. *Zaaraoui Lassaad, Mansouri Ali,* "Conception optimale d'un moteur-roue destiné à un véhicule électrique", Editions universitaires européennes, 2019.
[15]. *H. Liu, D. Yue, L. Zhang, Z. Li, D. Jiang,* "A New Improved Simplified Particle Swarm Optimization Algorithm," Journal of Physics Conference Series, **vol. 1187**, 2019.
[16]. *A. Mansouri, N. Smairi, H. Trabelsi,* "Multi-objective optimization of an in-wheel electric vehicle motor," International Journal of Applied Electromagnetics and Mechanics **vol. 50**, 2016, pp. 449–465.
[17]. *D. Martinez,* "Design of a permanent magnet synchronous machine with non-overlapping concentrated windings for the shell eco marathon urban prototype," Master's thesis, Royal Institute of Technology, Stockholm, 2012.
[18]. *Y. L. Feng, C. N. Zhang,* "Analytical calculation for predicting the air gap flux density in surface-mounted permanent magnet synchronous machine," Journal of Electrical Engineering and Technology, **vol. 12**, pp.769-777, 2017.
[19]. *A. Mansouri, N. Smairi, H. Trabelsi,* "Multi-objective optimization of an in-wheel electric vehicle motor," International Journal of Applied Electromagnetics and Mechanics, **vol. 50**, 2016, pp. 449–465.
[20]. *T. D. Strous,* "Design of a permanent magnet radial flux concentrated coil generator for a range extender application," Master's thesis, Delft University of Technology, 2010.