International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

An approach based on non-dominated sorting genetic algorithm III for design of permanent magnet synchronous motor

Mumtaz Mutluer[1*]

Submitted: 19/08/2020 Accepted : 06/10/2020

Abstract: Today due to industrial developments, the use of electric motors has increased in all fields. The increase also preceded the development of higher-specification motors. Although weight, cogging torque, torque ripples and drive technology etc. for the working area are important, the demand for the production of highly efficient and cost-effective motors has risen further due to the energy phenomenon in the world. High-quality algorithms are needed to achieve these objectives as well, because electric motor designs are multi-parameter and nonlinear engineering problems. This study aims to provide a multi-purpose intelligent design with NSGAII and NSGAIII by selecting outputs such as efficiency and cost of permanent magnet synchronous motor as an objective function. The design was intended for low speed and high torque/volume applications and the motor geometry was thus chosen as surface-mounted and double-layer concentrated winding. The optimization results were tested with a finite element program. Both methods resulted in a 3% increase in efficiency and a 37% reduction in cost versus initial design. Also, according to the results obtained, although NSGA-II and NSGA-III achieved similar results, NSGA-III results showed a more robust and stable course than NSGA-II results. The compatibility of the design optimization and the results of numerical analysis are acceptable and highly satisfactory. So, it provides outputs to demonstrate the features of an electric motor design optimization.

Keywords: Multi-objective design optimization, NSGAII, NSGAIII, PMSM

1. Introduction

Electric motors are designed and manufactured to meet industrial needs. So far, direct current motors have lost a wide range of applications due to high cost and maintenance, low efficiency and power density. Alternating current motors such as induction motor and permanent magnet synchronous motor (PMSM) are the most popular motor in industrial fields nowadays. Induction motors are interesting because of low cost and ease of maintenance and PMSMs have high power density and efficiency. In addition, the latest developments in control techniques and drive systems affect the choice of ac motors. Although the induction motor has good features, it is clear that the use of highly efficient PMSMs has increased especially due to performance criteria. The most prominent feature structural of PMSMs is the different layouts of permanent magnets. Naturally, this affects the performance and production costs of PMSMs [1]. Due to ease of design and low production costs, surface-mounted PMSMs are the most preferred types for low speed and high torque/volume applications. This model is also preferred due to the low cogging torque based on the rich combination of slots and poles, stator slot wedge and magnet shapes [2-3].

The design of PMSM is a complex engineering problem due to its nonlinear structure and numerous design parameters. Linear equations, viz. geometric, electrical, magnetic, mechanical and thermal equations by some simplifications are often used in the design optimization to overcome nonlinearity. Although design complexity is a challenge for designers, this problem is solved by the choice of fewer parameters for design objectives. These two cases, nonlinearity and complexity, show that evolutionary algorithms should be used in the design optimizations [4] and it is therefore possible to obtain more effective results by searching in a wide range of solutions.

As in other engineering studies, PMSM design optimizations can be made for one or more objectives. The results depend on the correct modelling of the problem, namely the proper design equations and the power of the algorithm. So far, many evolutionary algorithms have been used in the design optimizations of PMSMs and one of them is undoubtedly the genetic algorithm (GA). GA is generally used in single objectives to increase efficiency, reduce weight, and eliminate harmonics [5-7]. In fact, optimizations in engineering problems such as PMSM design often depend on multi-objective and the objectives often conflict with each other, so single objective algorithms cannot solve these problems at the desired level. Therefore, multi-objective evolutionary algorithms should be used for these problems. In some multi-objective studies, GA, fuzzy approach or taguchi method have been used to form multi-objective from single objective optimization [8-10].

As multi-objective evolutionary algorithms, vector evaluated genetic algorithm (VEGA) developed by Schaffer, multi-objective genetic algorithm (MOGA) suggested by Murata, niched pareto genetic algorithm (NPGA) recommended by Horn and Nafpliotis, non-dominated sorting genetic algorithm (NSGA) recommended by Srinivas and Deb, strength pareto evolutionary algorithm (SPEA) and the extension (SPEA2) proposed by Zitzler, pareto enveloped-base selection algorithm (PESA) proposed by Corne et al., and non-dominated sorting genetic algorithm II (NSGAII) were developed by Deb et al. [11]. With these algorithms, an equivalent set of solutions is obtained to solve a problem. When these

[1] *Department of Electrical and Electronic Engineering, Necmettin Erbakan University, Konya, Turkey*
ORCID NUMBER: 0000-0002-6781-8937
* *Corresponding Author Email: mmutluer@erbakan.edu.tr*

algorithms are tested with test functions, it has been shown that NSGAII provides better convergence and wider solution distribution than other multi-objective evolutionary algorithms [11]. Obviously NSGAII, a second-generation effective algorithm after NSGA, provides a pareto-optimal approach to solutions [12]. In fact, the aim of multi-objective optimization methods is not only to direct research to the best front, but also to preserve the diversity of the population in a non-dominant set of solutions. In this respect, research strategies make a difference. Non-dominated sorting genetic algorithm III (NSGAIII), third generation based on GA, is a powerful method proposed to address the lack of diversity-conserving operators among the best non-dominated solutions of NSGAII [13-14]. In this respect, it is thought that NSGAIII will be more effective in solving multi-parameter and multi-interactive problems such as electric motor designs. If the two methods are considered together, the search of NSGAIII for the solution space is similar to NSGII, i.e. the generation of the parent population, the generation of the new generation and the formation of fronts. The next difference is the homogeneous distribution and acceptance of the population. If qualifications for fronts cannot be achieved in each generation, reference points and reference lines are created with a hyper-plane depending on the objective function and division value (i.e. Das and Dennis's combinatorial reference point selection). After normalization and niche, the relationship of the individuals with each reference lines is determined and the nearest individual is chosen as the vertical distance [15]. This provides an advantage over previous multi-objective algorithms.

In the optimization studies of the PMSM design, finally the results are being tested by numerical methods and so the magnitudes at any point or the input/output values of are calculated. For this, analysis programs based on finite element method may be used. In fact, working with a finite element program is effective in terms of correctness of the results but weak in terms of duration [1].

Due to the rapid increase in the use of PMSMs especially in electric vehicle applications, their efficiency and costs are becoming more and more important day by day. Therefore, the efficiency and cost of PMSM were chosen as objective functions in this study. As variables, magnet thickness, air gap length, stator slot wedge height, stator tooth width, outer rotor diameter, stator slot height and ratio of the slot opening over the slot width of geometric parameters were chosen. To improve efficiency and to decrease cost, NSGAIII was first used in design optimization of PMSM and compared with NSGAII. In this case, a set of pareto-optimal fronts will be obtained in non-dominant solutions and the algorithms will be tested for their equal diversity. As a result, the optimization results were compared with graphs and tables and also validated with a finite element program. In this context, PMSM design optimization has been studied in a versatile way and useful inferences have been obtained. As a result of this study, a higher performance PMSM design was obtained.

2. Multidirectional Analysis of the PMSM

In the design of PMSM, electrical, magnetic, mechanical and thermal analyses are performed. The solution of the differential equations provided for these interactive analyses is quite complex. Since the magnetic structure of the motor is nonlinear, it is almost impossible to obtain precise solutions to these problems. In this case, numerical methods such as finite element method are used after preliminary analytical calculations. On the other hand, the use of linear equations for a basic design is obviously sufficient. In this context, mechanical and thermal analyses were excluded, the geometric model of the PMSM was created and the electrical and magnetic circuits were examined as follows.

The magnetic and electrical design studies of a PMSM based on the geometric model provide rapid analysis [1]. As the 3D model was shown in Figure 1, the designed PMSM with surface-mounted and inner rotor has 12 slots and 10 poles. The lateral edges of the magnets are radial to engage the origin. The stator teeth have the same width everywhere and the slots have trapezoidal shape. In general, the rotor yoke is wider than the stator yoke.

2.1. Magnetic Circuit

According to the magnetic circuit in Figure 2, it is very complicated to calculate the magnetic flux in each region of the PMSM. The most important issue in magnetic design is the accurate calculation of the magnetic flux density in the air gap [1]. The magnetic flux density in the stator and rotor is also particularly important for saturation. From this perspective, there may be values of magnetic limitations of 1.8T for the stator tooth, 1.4T for the stator yoke and 1.4T for the rotor yoke. By using the maximum magnetic flux density calculation of the magnet (neglecting saturation), the air gap magnetic flux density equation is obtained for the rectangular signals with the help of the basic harmonic equation. The general equations for the magnetic flux densities of stator tooth, stator yoke and rotor yoke are as follows [1], [16-17].

$$B_m = (B_r k_{leak} l_\mathrm{m})/(l_\mathrm{m} + \mu_r \delta k_C) \tag{1}$$

$$\hat{B}_\delta = (4/\pi) B_m \sin\alpha \tag{2}$$

$$B_{st} = \left(4\alpha B_m (D/2 - \delta)\right)/(2p b_{st}) \tag{3}$$

$$B_{sy} = \left(4\alpha B_m (D/2 - \delta)\right)/\left(2p h_{sy}\right) \tag{4}$$

$$B_{ry} = \left(4\alpha B_m (D/2 - \delta)\right)/\left(2p h_{ry}\right) \tag{5}$$

where, remanence flux density of permanent magnet is B_r, maximum of air gap flux density is B_m, fundamental of air gap flux density is $\hat{B}_\delta$, flux density in a stator tooth is B_{st}, flux density in stator yoke is B_{sy}, flux density in rotor yoke is B_{ry}, correction factor for air gap flux density is k_{leak}, relative magnet permeability is μ_r, Carter factor is k_C, pole angle is 2α, inner stator diameter is D, number of slots per pole per phase is q, stator yoke height is h_{sy}, rotor yoke height is h_{ry}.

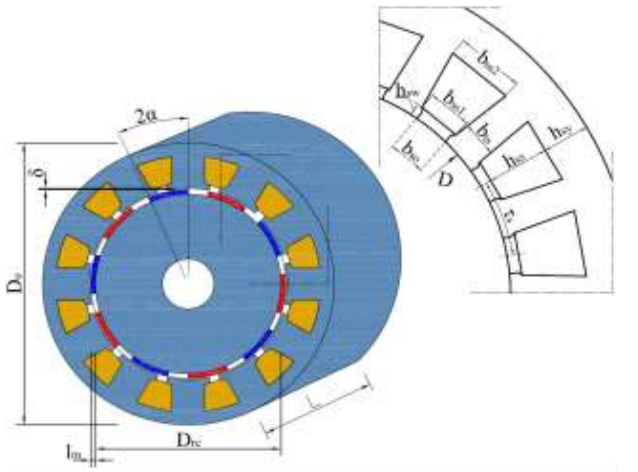


Figure 1. 3D geometric model of the PMSM

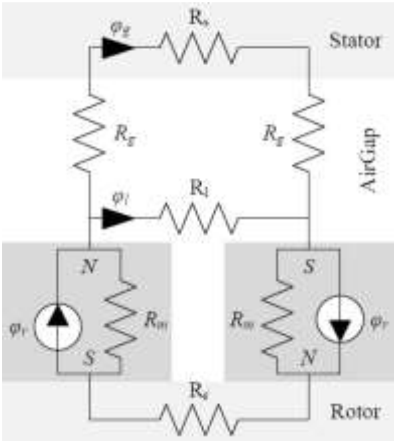


Figure 2. Magnetic circuit of the PMSM

2.2. Electrical Circuits

At the base speed $d-q$ electrical circuits of the PMSM were given in Figure 3. When the moment equation is examined, the number of windings of the motor is calculated only according to the I_q current because the I_d current is zero in the surface-mounted (non-salient) PMSMs [17]. The induced phase voltage, herein the $d-q$ axes synchronous inductances are equal and the phase resistance:

$$\hat{E} = \omega k_{\omega 1} q n_s \hat{B}_\delta L(D-\delta) \tag{6}$$

$$R_{Cu} = \left(\rho_{Cu}\left(pL + \pi k_{coil}(D + h_{ss})\right)n_s^2 q\right)/f_s A_{sl} \tag{7}$$

$$L_{d,q} = \left(pq\lambda + \frac{3(qk_{\omega 1})^2(D-\delta)}{\pi(\delta k_C + l_m/\mu_r)}\right)\mu_0 L n_s^2 \tag{8}$$

$$\hat{U} = \sqrt{U_q^2 + U_d^2} = \sqrt{\left(\hat{E} + RI_q\right)^2 + \left(L_q\omega I_q\right)^2} \tag{9}$$

where, electrical angular frequency is ω, fundamental winding factor is $k_{\omega 1}$, conductor number per slot is n_s, stack length is L, copper wire resistivity is ρ_{Cu}, end-winding coefficient is k_{coil}, slot fill factor is f_s, slot area is A_{sl}, specific permeance coefficient of the slot opening is λ, $d-q$ axes terminal voltages are is $U_{d,q}$, $d-q$ axes currents are $I_{d,q}$, fundamental of the induced voltage is $\hat{E}$, winding resistance is R, $d-q$ axes magnetizing inductance is $L_{d,q}$.

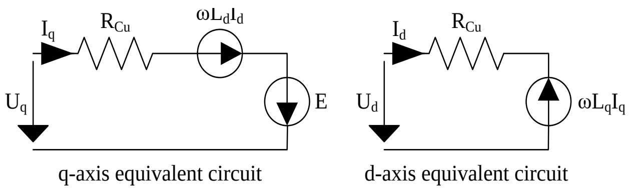


Figure 3. $d-q$ equivalent circuits of the PMSM

According to the electrical and magnetic analysis of PMSM, an evaluation can be made in terms of efficiency and cost functions. In general, gearless PMSMs are more efficient than others with gears. In order to increase the efficiency in a fixed output power motor design, it is necessary to reduce losses, especially copper losses in multi-pole low frequency structures. In terms of cost, motor and magnet volumes are important. Permanent magnets are the most expensive parts of PMSMs and prices are changing rapidly due to technological advances. In this study, it was aimed to increase efficiency and reduce cost. Obtaining all PMSM design equations is a very detailed process and therefore references have been made to different studies [1], [7], [16-17]. As a result, the objective functions are obtained as follows.

$$P_{out} = \frac{3}{2}\frac{p}{2}\psi_m I_q = \frac{3}{2}\hat{E}\hat{I}_q = T\omega_m \tag{10}$$

$$f_1(x_1, x_2, \dots, x_k) = Efficiency = \frac{P_{out}}{P_{out}+P_{Cu}+P_{Fe}} \tag{11}$$

$$f_2(x_1, x_2, \dots, x_k) = Cost = \sum_{n=1}^{N} Cost_n Material_n \tag{12}$$

where, output power is P_{out}, copper losses is P_{Cu}, iron losses are P_{Fe}.

3. Multi-Objective Optimization Algorithms

NSGAII and NSGAIII multi-objective optimization algorithms are based on conventional genetic algorithm. GA has been used in many optimization problems such as efficiency, weight, cost and harmonics of electric motors or other industrial problems [5-6], [18-21]. In these studies, solutions often depend on multiple conflicting objectives, so it is necessary to use multi-objective evolutionary algorithms for such optimization problems. In this section, while GA was briefly explained, NSGAII and NSGAIII were introduced in more detail.

GA structurally consists of genes and chromosomes. These building blocks correspond to probabilities of the individual variables and their values, and the results converge to the optimal solutions. The population size, gene and chromosome numbers related to the input parameters influence the solution accuracy. GA does not need initial solution, optimizes continuous and discrete parameters, does not require derivative information, can search the objective function in a wide spectrum, can work with many parameters. However, GA does not guarantee the optimal solution and may converge to a local solution [22]. So far, GA provides the basis for the development of multi-objective evolutionary algorithms such as NSGAII and so on [12].

Algorithms such as GA, differential evolution algorithm, particle swarm optimization algorithm or others are generally used as minimum or maximum single objective in solving optimization problems. However, the optimization problems especially electric motor designs may be more suitable for multi-objective [23-24]. In this case, it is inevitable to obtain solutions for each objective and a single solution that is best for all objectives may not be available. Thus, the decision maker is asked to select any solution from a final set of agreed upon terms. The appropriate solution should provide an acceptable level of performance for all objectives.

In multi-objective evolutionary algorithms, different techniques as aggregation functions, population-based approaches and pareto-based approaches are used to classify solutions. In the aggregation functions, it is tried to obtain a single scalar output by multiplying the aims by different weight values [25]. The pareto-based approaches are used to diversify the search process such that the population is divided into sub-populations for each objective to find the pareto-surface. The most widely used is the pareto-optimal approach. In this approach, non-dominated individuals in each iteration attempt to produce pareto-optimal surfaces and appropriate solutions. Strictly speaking, in pareto-based approaches, attempts are made to obtain a set of non-dominated appropriate solutions.

The general form of a multi-objective optimization problem can be mathematically expressed as follows [26]:

$$\begin{aligned}
&find\ x\ which\ maximize/minimize\ \ f(x) \\
&subject\ to;\quad h_m(x) = 0, \qquad m = 1,2,\dots,M \\
&\qquad\qquad\qquad g_j(x) \le 0, \qquad j = 1,2,\dots,J \\
&\qquad\qquad\qquad x_k^l \le x_k \le x_k^u, \qquad k = 1,2,\dots,K
\end{aligned}$$

where objective functions form the multi-objective function vector as $f(x) = [f_1(x), f_2(x), \ldots, f_n(x)]$, $h_m(x)$ and $g_j(x)$ are equality and inequality constraint functions. x_k^u and x_k^l are the upper and lower limits of the input parameters.

It is necessary to explain the pareto-optimal approach in order to understand NSGAII and even NSGAIII too well. In multiple optimization problems, the objectives are not compatible with each other, and therefore a common single objective is not possible. In this respect, it is necessary to choose the solution balance between antagonisms or objectives between multiple objectives. In order to do this, an evaluation theory should be developed. Herein, when the common solution for each purpose is compared with other solutions, the "pareto-optimal" approach is used. Accordingly, any solution may be better, worse or the same as other solutions. In this case, the best solution is a non-dominated solution that is better at least for one objective than others. Solutions with this feature create pareto-optimal solutions. Pareto-optimal solutions form a pareto-optimal surface and any solution from them can be chosen by the decision maker about the problem [12].

3.1. Non-Dominated Sorting Genetic Algorithm II (NSGAII)

NSGAII was developed by Deb with reference to the NSGA [12], [27-28]. The computational complexity, the weaknesses such as lack of elitism and uncertainty in setting the share parameter value in NSGA have been tried to be solved in NSGAII. In the NSGAII, it is necessary to compare the solutions to form the pareto-optimal surface by detecting the non-dominated solutions in the population of N individuals. In order to be able to do this classification, the n_p value indicating the number of the superiority of the solutions suppressing a solution p and the set of S_p solutions set dominated by the solution p must be computed. Using these values, fronts are determined according to the level of suppression in the population. For each p solution in the second and subsequent fronts, the n_p value can be at most $N - 1$. The pseudo code for NSGAII's fast-non-dominated-sort is given in Figure 4.

```
fast-non-dominated-sort(P)
for each p ∈ P
  S_p = ∪∅
  n_p = 0
  for each q ∈ P
    if (p ≺ q) then              If p dominates q
      S_p = S_p ∪ {q}            Add q to the set of solutions dominated by p
    else if (q ≺ p) then
      n_p = n_p + 1              Increment the domination counter of p
  if n_p = 0 then                p belongs to the first front
    p_rank = 1
    F_1 = F_1 ∪ {p}
i = 1                            Initialize the front counter
while F_i ≠ ∅
  Q = ∅                          Used to store the members of the next front
  for each p ∈ F_i
    for each q ∈ S_p
      n_q = n_q − 1
      if n_q = 0 then            q belongs to the next front
        q_rank = i + 1
        Q = Q ∪ {q}
  i = i + 1
  F_i = Q
```

Figure 4. The pseudo code for NSGAII [12]

In order to be able to achieve diversity in multi-objective evolutionary algorithms, the solutions must have a good spreading towards pareto-optimal surface. For this orientation, NSGAII is separated from NSGA. When NSGA uses a sharing method [27], NSGAII uses a crowded-comparison approach to ensure uniform convergence to pareto-optimal surface and so that solutions have an appropriate spread for each objective. This approach based on density estimation and crowded-comparison operator is not user intrusive and has a good computational complexity. The distance of neighbours close to each other is determined for each objective on the pareto-surface. Determination is based on the principle of calculating the distances of the cuboids to each other. The crowding-distance values in the non-dominated front are calculated as the sum of the individual distances for all objectives (Figure 5 and pseudo code in Figure 6).

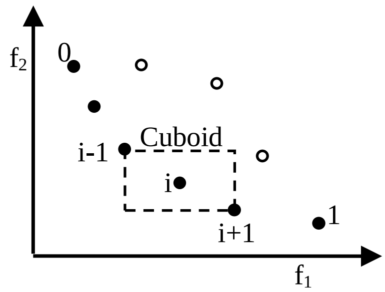


Figure 5. Crowding-distance calculation. Points marked in filled circles are solutions of the same non-dominated front [12]

```
crowding-distance-assignment(I)
l = |I|                          number of solutions in I
for each i, set I[i]_distance = 0   initialize distance
for each objective m
  I = sort(I, m)                 sort using each objective value
  I[i]_distance = I[i]_distance = ∞   so that boundary points are always selected
  for i = 2 to (l − 1)           for all other points
    I[i]_distance = I[i]_distance + (I[i+1].m − I[i−1].m)/(f_m^max − f_m^min)
```

Figure 6. The pseudo code for crowding-distance-assignment [12]

NSGAII compliance is considered to be minimization. First of all, a population is randomly generated and is sorted according to non-domination and then tournament selection, crossover and mutation operators are applied. As a result of the first iteration, individuals of size $2N$ are combined as $P_t + Q_t$. After the first iteration, the elitism process is performed by comparing the current population with the previous best non-dominated solutions. The solutions of best non-dominated individuals are listed as set F_1. If F_1 is smaller than N length, then F_1 is selected to the next population and the other individuals are selected from the lower non-dominated solutions. This is how the sets are selected from F_1 to F_l. If the count of solutions in all sets is larger than N individuals, the solutions of the last front (F_1) using the crowded-comparison operator are sorted. NSGAII main procedure and flowchart are in Figure 7 and in Figure 8 respectively [12].

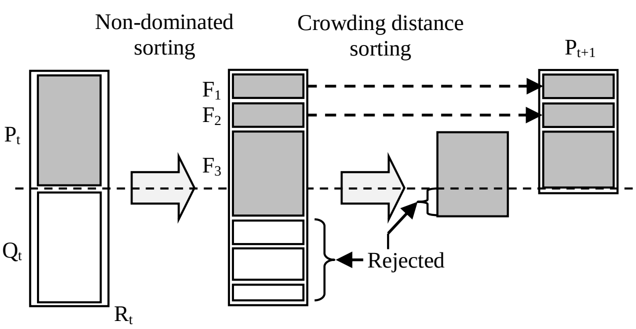


Figure 7. NSGAII main procedure [12]

$R_t = P_t \cup Q_t$ combine parent and offspring population
$F = fast-non-dominated-sort(R_t)$ $F = (F_1, F_2, \ldots)$, all non-dominated fronts of F
$P_{t+1} = \varnothing \; and \; i = 1$
$until \left|P_{t+1}\right| + \left|F_i\right| \le N$ until the parent population is filled
$crowding-distance-assigment(F_i)$ calculate crowding-distance in F_i
$P_{t+1} = P_{t+1} \cup F_i$ include ith non-dominated front in the parent pop
$i = i + 1$ check the next front for inclusion
$Sort(F_i, \prec_n)$ sort in descending order using $\prec_n$
$P_{t+1} = P_{t+1} \cup F_i\left[1 : N - \left|P_{t+1}\right|\right]$ choose the first $N - \left|P_{t+1}\right|$ elements of F_i
$Q_{t+1} = make-new-pop(P_{t+1})$ use selection, crossover and mutation to create a new population Q_{t+1}
$t = t + 1$ increment the generation counter

Figure 8. NSGAII flowchart [12]

3.2. Non-Dominated Sorting Genetic Algorithm III (NSGAIII)

Multi-objective evolutionary algorithms have been developed to solve multiple (two or three) objective problems. Since the objective function for more complex problems will increase, today's many-objective evolutionary algorithms have been developed specifically to solve more than three purposeful optimization problems. One of these algorithms, NSGAIII, was developed based on NSGAII. The main distinction between the two algorithms is that the strategies for scanning the solution space are different [13].

The needs of many-objective evolutionary algorithms, namely the factors affecting the development of algorithms, can be listed as follows:

i. Decreasing in diversity of non-dominated solutions and slow down search as objective values are achieved

ii. Increasing the size of the algorithm by the methods developed to ensure such diversity (crowding distance)

iii. Affecting the decision-making process of the algorithm, since the growth of the non-dominated solution front will create a visualization problem.

Here, while explaining the work of NSGAIII algorithm, the algorithms are compared by expressing the differences from NSGAII.

In NSGAIII, the size of the population is compared with the number of new population individuals to select individuals in the population based on their non-dominated fronts. What is important here is the total size of the new individuals from the population size. If the two values are equal, no operator is required, but if the population size is exceeded, the remaining individuals are selected from the last front. This selection uses the hyper-plane selection criterion.

The strategy of selecting reference points is effective in maintaining the diversity of non-dominated solutions. While any strategy may be preferred for the selection of reference points, the combination of Das and Dennis' combinatorial point selection is preferred at this stage due to the activity in question. The mathematical expression of this selection criterion is shown in Equation 13. This method is more widely used in a combined application of decision-making and multi-objective optimization because it is very likely that the proposed algorithm is located near the Pareto-optimal solutions corresponding to the reference points. In Figure 9, a hyper-plane selection criterion is shown with three reference functions ($M = 3$), four sections ($p = 4$), 15 reference points with axes $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$. The flow code of this criterion algorithm is given in Stage 1 (Figure 10).

$$H = \binom{M+p-1}{p} \quad \begin{matrix} M, objective\ problem \\ H, reference\ points \\ p, division\ of\ hyperplane \end{matrix} \tag{13}$$

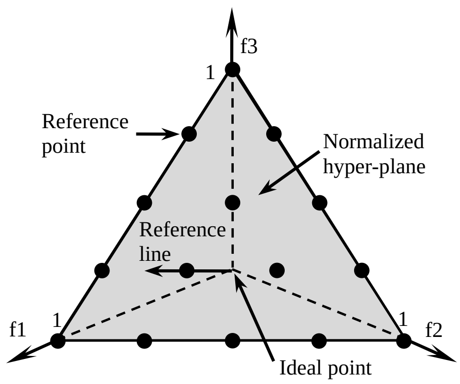


Figure 9. Determination of reference points on a hyper-plane [13]

In the next step, the minimum values of the objective functions of each individual of the population are obtained. The scaling function is provided by using the maximum objective function values with the differential approach between the minimum values and normal values of the objective functions. For the creation of Pareto optimal fronts, Das and Dennis's combinatorial reference point selection criteria and normalization process are applied again, resulting in a new hyper-plane. This flow (Stage 2) contributes to the robustness of the variety of solutions. The reference points are then combined with the origin and reference lines are formed to associate each individual in the population with a reference point. It can be said that the perpendicular distance of the individuals is related to the relevant reference point for the nearest reference line (Stage 3). In this case, the reference points are connected to the population as in Stage 4.

Stage 1

Input: *H structured reference points* Z^s *or supplied aspiration points* Z^a, *parent population* P_t

Output: $P_{(t+1)}$

$S_t = \varnothing, i = 1$
$Q_t = Recombination + Mutation(P_t)$
$R_t = P_t \cup Q_t$
$(F_1, F_2, \ldots) = Non-dominated-sort(R_t)$
repeat
$\quad S_t = S_t \cup F_i$ *and* $i = i + 1$
until $|S_t|\ ?\ N$
Last front to be included: $F_l = F_i$
if $|S_t| = N$ *then*
$\quad P_{(t+1)} = S_t$, *break*
else
$\quad P_{(t+1)} = U_{(j=1)}^{(l-1)} F_j$
$\quad$ *Points to be chosen from* F_l: $K = N - |P_{(t+1)}|$
$\quad$ *Normalize objectives and create reference set* Z^r: $Normalize(f^n, S_t, Z^r, Z^s, Z^a)$
$\quad$ *Associate each numbers of* S_t *with a reference point*: $[\pi(s), d(s)] = Associate(S_t, Z^r)$,
$\quad \pi(s)$: *closest reference point*, d: *distance betweens and* $\pi(s)$
$\quad$ *Compute niche count of reference point* $j \in Z^r$: $\rho_j = \sum_{s \in S_t / F_l} \left(\left(\pi(s) = j \right) ?1:0 \right)$
$\quad$ *Choose K members one at a time from* F_l *to construct* $P_{(t+1)}$: $Niching(K, \rho_j, \pi, d, Z^r, F_l, P_{t+1})$
end if

Figure 10. Generation $\boldsymbol{t}$ of NSGAIII procedure [13]

Stage 2

Input: S_t, Z^s (*structured points*) *or* Z^a (*supplied points*)

Output: f^n, Z^s (*reference points on normalized hyper − plane*)

for $j = 1$ *to* M *do*
$\quad$ *Compute ideal point*: $z_j^{\min} = \min_{s \in S_t} f_j(s)$
$\quad$ *Translate objective*: $f_j^{'}(s) = f_j(s) - z_j^{min} \quad \forall s \in S_t$
$\quad$ *Compute extreme points* $\left(z_j^{max}, j = 1, \ldots, M \right)$ *of* S_t
end for
Compute intercepts a_j *for* $j = 1, \ldots, M$
Normalize objectives (f^n) *using the function* $\left(f_i^n(x) = \frac{f_i'(x)}{a_i}, \text{ for } i = 1, 2, \ldots, M \right)$
if Z^a *is given then*
$\quad$ *Map each (aspiration) point on normalized hyper − plane using* $\left(f_i^n(x) = \frac{f_i'(x)}{a_i}, \text{ for } i = 1, 2, \ldots, M \right)$ *and save the points in the set* Z^r
else
$\quad Z^r = Z^s$
end if

Figure 11. Normalize $\left(\boldsymbol{f^n}, \boldsymbol{S_t}, \boldsymbol{Z^r}, \boldsymbol{Z^s} / \boldsymbol{Z^a} \right)$ procedure [13]

Stage 3

Input: Z^r, S_t

Output: $\pi(s \in S_t), d(s \in S_t)$

for each reference point $z \in Z^r$ *do*
$\quad$ *Compute reference line* $w = z$
end for
for each $s \in S_t$ *do*
$\quad$ *for each* $w \in Z^r$ *do*
$\quad\quad$ *Compute* $d^{\perp}(s, w) = \left\| \left(s - w^T s w / \|w\|^2 \right) \right\|$
$\quad$ *end for*
$\quad$ *Assign*$(s) = w : \operatorname{argmin}_{w \in Z^r} d^{\perp}(s, w)$
$\quad$ *Assign* $d(s) = d^{\perp}(s, \pi(s))$
end for

Figure 12. Associate $\left(\boldsymbol{S_t}, \boldsymbol{Z^r} \right)$ procedure [13]

Stage 4

Input: $K, \rho_j, \pi(s \in S_t), d(s \in S_t), Z^r, F_l$

Output: P_{t+1}

$k = 1$
while $k \le K$ *do*
$\quad J_{min} = \left\{ j : argmin_{j \in Z^r} \rho_j \right\}$
$\quad \bar{j} = random\left(J_{min} \right)$
$\quad I_{\bar{j}} = \left\{ s : \pi(s) = \bar{j}, s \in F_l \right\}$
$\quad$ *if* $I_{\bar{j}} \ne \varnothing$ *then*
$\quad\quad$ *if* $\rho_{\bar{j}} = 0$ *then*
$\quad\quad\quad P_{t+1} = P_{t+1} \cup \left\{ s : argmin_{s \in I_{\bar{j}}} d(s) \right\}$
$\quad\quad$ *else*
$\quad\quad\quad P_{t+1} = P_{t+1} \cup random\left(I_{\bar{j}} \right)$
$\quad\quad$ *end if*
$\quad\quad \rho_{\bar{j}} = \rho_{\bar{j}} + 1, F_l = F_l \setminus s$
$\quad\quad k = k + 1$
$\quad$ *else*
$\quad\quad Z^r = Z^r / \left\{ \bar{j} \right\}$
$\quad$ *end if*
end while

Figure 13. Niching $\left(\boldsymbol{K}, \boldsymbol{\rho_j}, \boldsymbol{\pi}, \boldsymbol{d}, \boldsymbol{Z^r}, \boldsymbol{F_l}, \boldsymbol{P_{t+1}} \right)$ procedure [13]

4. Optimization Steps and Results

PMSM was modelled analytically using with the developed design program and then was analysed with Ansys/RMxprt module where stator, rotor, winding, magnet and overall motor size should be entered. Some geometric dimensions and winding dimensions can be selected automatically in RMxprt module. Analytical analysis can be performed for data such as constant speed, torque, power and ultimately full load, no load, motor size and so on. According to the developed mathematical model and the calculations made with RMxprt and Maxwell, the input power of the motor was 2607.36W, 2579.25W and 2723.62W3W, the output power was 2400W, 2405.81W and 2436.11W, and the efficiency was 92.05%, 93.28% and 89.44%, respectively. The resulting error values are -1.32% for RMxprt and 2.91% for Maxwell. In addition, the cost of the PMSM was obtained 227.6$.

In order to work with evolutionary algorithms, the input parameters and their limits, constants and variables must first be specified. Since PMSM designs require multiplex equations, the limits of input parameters should be carefully selected based on experience and requirements. The used input parameters and their limits were given in Table 1. Constants were selected as 340V, 2.4kW, 250rpm, 300mm, 120mm and 126° for the supply voltage, power, speed, stator outer diameter, stack length and electrical magnet angle respectively. To reduce number of equations with making some negligence and using coefficients is an effective approach in obtaining objective functions [1]. Material unit prices for the second objective function were given in Table 2 [29].

Table 1. Input parameters and limits

Parameter	Symbol	Lower limit	Upper limit
Magnet thickness (mm)	l_m	2	5
Air gap length (mm)	δ	0.5	1.2
Slot wedge height (mm)	h_{sw}	2	5
Stator tooth width (mm)	b_{ts}	30	40
Outer rotor diameter (mm)	D_{rc}	150	250
Stator slot height (mm)	h_{ss}	15	22
Ratio of the slot opening over the slot width	k_{open}	0.25	0.4

Table 2. Material unit prices ($/TON, JAN-2018)

Copper	Lamination	Permanent Magnet (NdFeB)
7048	1122	68747

For design optimization of PMSM, NSGAII and NSGAIII methods were run thirty times for 25, 50, and 100 populations and iterations. The average values of the best results for each objective function were given in Figure 14. As the population and the number of iterations increase, the results of both algorithms improve and ultimately show no change. According to the graphs of average efficiency and cost, the NSGAIII gave better results than NSGAII in all population and iteration numbers. This is very promising in solving a complex design problem with many equations. Subsequent analyses were performed for parameter values providing the best solution obtained with both algorithms and also only outputs of the NSGAIII were presented graphically.

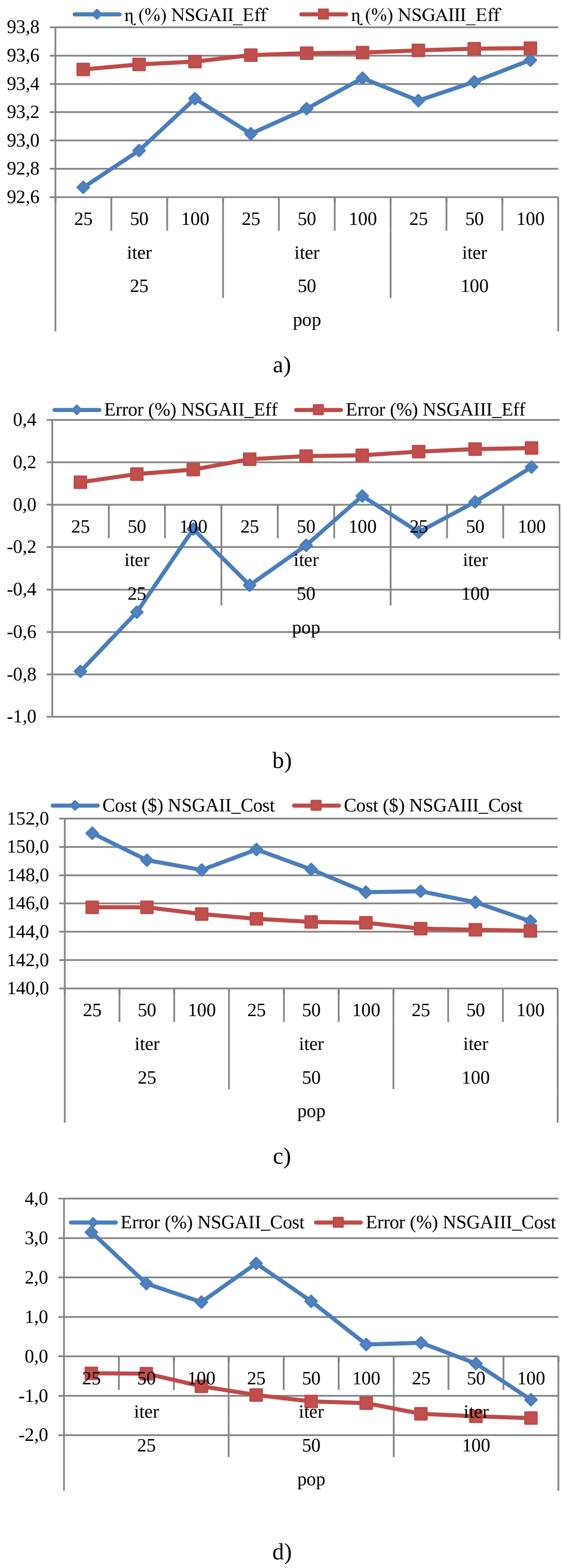
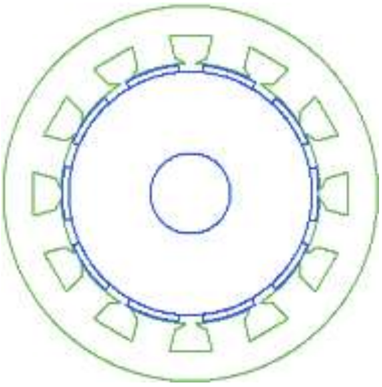
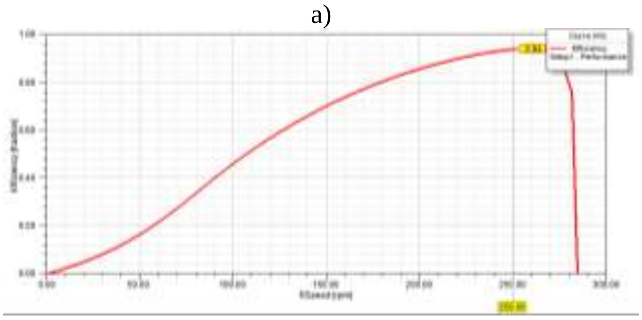


Figure 14. Average efficiency and cost values obtained with NSGAII and NSGAIII for thirty iterations

The maximum efficiency obtained are 93.65% for NSGAII and %93.66 for NSGAIII and the minimum cost 148.81$ for NSGAII and 145.17$ for NSGAIII. Then the motor models which have high efficiency found by both algorithms were validated with RMxprt and Maxwell and so the efficiency obtained by RMxprt are %93.57 and %93.61 and obtained by Maxwell %92.17 and %92.17 respectively (Table 3). The resulting error values for NSGAII are 0.08% and 1.61%, for NSGAIII 0.062% and 1.61% according to RMxprt and Maxwell. The general view and efficiency/speed curve of RMxprt for the PMSM with NSGAIII were given in Figure 15.



a)



b)

Figure 15. (a) General view (b) the efficiency/speed curve of the PMSM with RMxprt for NSGAIII

2D or 3D magnetic dynamic analysis of the PMSM can be done by taking the analytical analysis to the Maxwell module. In the finite element analysis, the magnetic flux distribution of the PMSM and the outputs such as moment, voltage and current graphs can be taken in detail. The torque/time graph was given in Figure 16 and the magnetic flux and magnetic flux density distribution of the PMSM belonging to Maxwell and the overall mesh view were given in Figure 17. According to the selected model and optimization results, the PMSM has the appropriate magnetic flux distribution and the magnetic flux density is generally low. Although it is obvious that the magnetic flux density in the stator slot opening part is high, these obtained results have a positive effect on the operation of the motor.

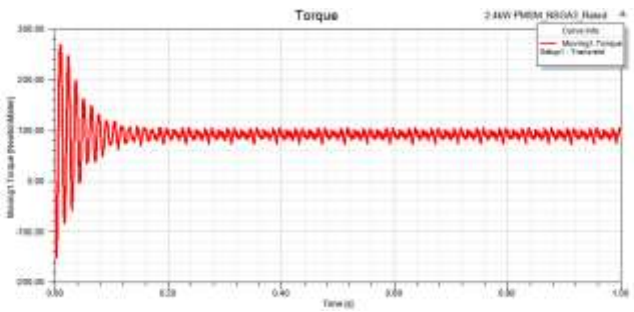
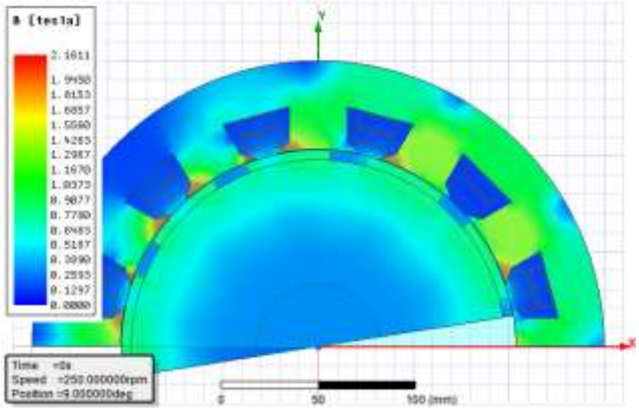
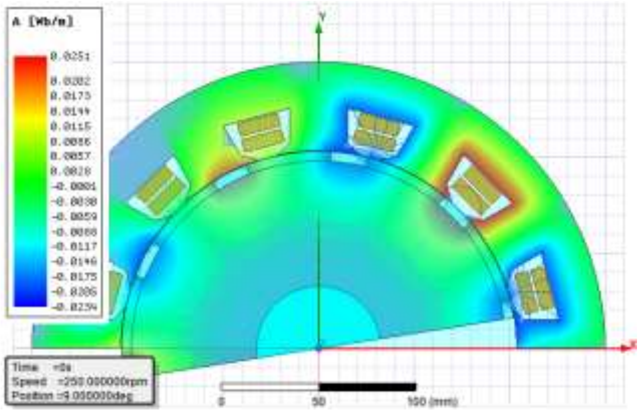


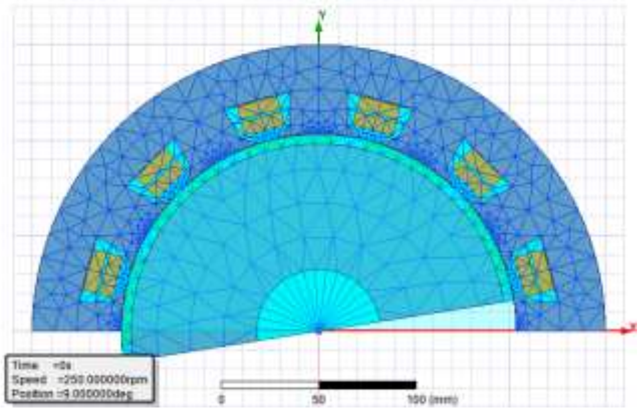
Figure 16. Torque/time curve for NSGAIII



a)



b)



c)

Figure 17. Magnetic flux (a), magnetic flux density (b) and mesh (c) view with Maxwell for NSGAIII

Table 3 which contains all results shows that the optimization with both algorithms improves the PMSM design. NSGAII achieved the best efficiency of 93.65% in 100 population and 100 iteration numbers. NSGAIII achieved the best efficiency of 93.66% in 100 population and 50 iteration numbers. The NSGAII method provided a cost of $244.36 for the best efficiency, while the NSGAIII provided $243.43 for the best efficiency. This value of NSGAIII is more preferred. When the losses are examined, it can be said that the losses increase by increasing the magnetic flux on the stator tooth surfaces with the increase of stator slot opening according to magnetic flux density in Figure 17. In this context, the regulation of the limits of the input parameters can be determined. In this study, it is emphasized that the NSGAIII does not require any additional adjustable parameters and in this respect, NSGAIII requires less computational complexity when compared with NSGAII. Based on the search feature, NSGAIII are more strong method than NSGAII to find non-dominated solutions by balancing every objective namely by providing and updating a range of well-spread reference points and so NSGAIII is more successful and useful in all populations and iterations than NSGAII. It is possible to perform different multi-parameter PMSM design optimization studies and get better results with NSGAIII method which is developed as many objectives.

5. Conclusion

In general, the design of electrical machines is very complex because it is multi-dimensional and nonlinear. For this reason, it is inevitable to use multi-objective evolutionary algorithms in the analysis of these problems and to test the results with numerical methods. In this study, it was aimed to improve the design of 12 slots 10 poles permanent magnet synchronous motor with good geometric architecture for high torque low speed applications. Target outputs were determined as efficiency and cost and seven input parameters of motor geometry were selected. NSGAII and NSGAIII methods based on GA were chosen for optimization to obtain strong results. NSGAII and NSGAIII methods resulted in an approximate 1.75% increase in efficiency and a 36.2% reduction in cost versus initial analytical design. Algorithms were run for 25, 50 and 100 population and iteration numbers, as a result, the NSGAIII method outperformed NSGAII in all population and iteration numbers. The analytical and optimization results were validated, and very close values were obtained. According to finite element graphics, it was determined that the losses may increase due to the increase in the magnetic flux density in the stator slot opening section. Application of NSGAIII method in permanent magnet synchronous motor design and comparison with NSGAII for the first time, it is predicted that the structure of NSGAIII which enables many objective functions, multiple optimizations can also improve the design.

Acknowledgemnt

I would like to thank Selçuk University for their support in using the finite element program.

Table 3. Optimization and FEM results

	Initial			**RMxprt**				**Maxwell**			
	Pin	**Pout**	**% Eff**	**Pin**	**Pout**	**% Eff**	**% Err**	**Pin**	**Pout**	**% Eff**	**% Err**
Analytic	2607.36	2400	92.05	2579.25	2405.81	93.276	-1.317	2723.62	2436.11	89.444	2.911
NSGAII	2562.76	2400	93.65	2580.15	2414.32	93.573	0.081	2655.77	2447.81	92.170	1.605
NSGAIII	2562.56	2400	93.66	2559.25	2395.58	93.605	0.055	2658.16	2450.11	92.173	1.609

References

[1] D. C. Hanselman (2006). Brushless permanent magnet motor design. Magna Physics Publishing, Ohio.

[2] V. S. Sempere, M. B. Payán, J. R. C. Bueno (2017). Cogging torque cancellation by magnet shaping in surface-mounted permanent-magnet motors. IEEE Transactions on Magnetics, Vol. 53, Issue 7, doi:10.1109/TMAG.2017.2676090.

[3] Herlina, R. Setiabudy, A. Rahardjo (2017). Cogging torque reduction by modifying stator teeth and permanent magnet shape on a surface mounted PMSG. International Seminar on Intelligent Technology and Its Applications, pp: 227-232, doi:10.1109/ISITIA.2017.8124085.

[4] B. N. Cassimere, S. Sudhoff (2009). Population-based design of surface-mounted permanent-magnet synchronous machines. IEEE Transactions on Energy Conversion, Vol. 24, No. 2, pp 338-346, doi:10.1109/TEC.2009.2016150.

[5] L. Jing, R. Qu, W. Kong, D. Li, H. Huang (2017). Genetic-algorithm-based analytical method of SMPM motors. IEEE International Electric Machines and Drives Conference, doi:10.1109/IEMDC.2017.8002030.

[6] W. Zhao, J. W. Kwon, X. Wang, T. A. Lipo, B. I. Kwon (2017). Optimal design of a spoke-type permanent magnet motor with phase-group concentrated-coil windings to minimize torque pulsations. IEEE Transactions on Magnetics, Vol. 53, Issue 6, doi:10.1109/TMAG.2017.2664075.

[7] M. Mutluer, O. Bilgin (2016). An intelligent design optimization of a permanent magnet synchronous motor by artificial bee colony algorithm. Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 24, pp 1826-1837, doi:10.3906/elk-1311-150.

[8] S. Owatchaiphong, N. H. Fuengwarodsakul (2009). Multiobjective based optimization for switched reluctance machines using fuzzy and genetic algorithms. International Conference on Power Electronics and Drive Systems, doi:10.1109/PEDS.2009.5385926.

[9] M. T. Chui, J. A. Chiang, J. M. Lee, Z. L. Gaing (2014). Multiobjective optimization design of interior permanent-magnet synchronous motors for improving the effectiveness of field weakening control. 17th International Conference on Electrical Machines and Systems, doi:10.1109/ICEMS.2014.7013534.

[10] Trisnal, Marimin, Y. Arkeman (2016). Solving fuzzy multi-objective optimization using non-dominated sorting genetic algorithm II. International Conference on Advanced Computer Science and Information Systems, doi: 10.1109/ICACSIS.2016.7872798.

[11] T. Sağ, M. Çunkaş (2009). A tool for multiobjective evolutionary algorithms. Advances in Engineering Software, Vol. 40, Issue 9, pp 902-912, doi:10.1016/j.advengsoft.2009.01.001.

[12] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan (2000). A fast elitist nondominated sorting genetic algorithm for multiobjective optimization: NSGA-II. International Conf. on Parallel Problem Solving from Nature, pp 849-858.

[13] K. Deb and H. Jain (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I. IEEE Transactions on Evolutionary Computation, Vol. 18, Issue 4, pp 577-601, doi:10.1109/TEVC.2013.2281535.

[14] H. Jain and K. Deb (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part II. IEEE Transactions on Evolutionary Computation, Vol. 18, Issue 4, pp 602-622, doi:10.1109/TEVC.2013.2281534.

[15] R. H. Bhesdadiya, I. N. Trivedi, P. Jangir, N. Jangir, A. Kumar (2016). An NSGA-III algorithm for solving multi-objective

economic/environmental dispatch problem. Cogent Engineering, Vol. 3, Issue 1, doi:10.1080/23311916.2016.1269383.
[16] J. F. Gieras, M. Wing (2002). Permanent magnet motor technology design and applications, second edition, revised and expanded, Marcal Dekker Inc., New York.
[17] F. Libert (2004). Design, optimization and comparison of permanent magnet motors for a low-speed direct-driven mixer. Technical Licentiate, School of Computer Science, Electrical Engineering and Engineering Physics, KTH, Sweden.
[18] R. Chaudhary, R. Sanghavi, S. Mahagaokar (2017). Optimization of induction motor using genetic algorithm and GUI of optimal induction motor design in MATLAB. Advances in Systems, Control and Automation, pp 127-132.
[19] Z. S. Liu (2017). Design and performance simulation of direct drive hub motor based on improved genetic algorithm. The Fourth Euro-China Conference on Intelligent Data Analysis and Applications, pp 303–313, doi:10.1007/978-3-319-68527-4_33.
[20] M. A. Şahman, M. Çunkaş, Ş. İnal, F. İnal, B. Coşkun and U. Taşkiran (2009). Cost optimization of feed mixes by genetic algorithms. Advances in Engineering Software, 40(10), 965-974.
[21] M. Mutluer, M. A. Şahman and M. Çunkaş (2020). Heuristic optimization based on penalty approach for surface permanent magnet synchronous machines. Arabian Journal for Science and Engineering, 45(8), 6751-6767.
[22] R. L. Haupt, S. E. Haupt (1998). Practical Genetic Algorithms. A Willey-Interscience Publication, USA.
[23] S. F. Contreras, C. A. Cortes, M. A. Guzmán (2017). Modelling of squirrel cage induction motors for a bio-inspired multiobjective optimal design. IET Electric Power Applications, Vol. 11, Issue 4, pp 512-523.
[24] B. Anvari, H. A. Toliyat (2017). Simultaneous optimization of geometry and firing angles of in-wheel switched reluctance motor. IEEE Energy Conversion Congress and Exposition, pp 760-767, doi:10.1109/ECCE.2017.8095861.
[25] Y. Duan, R. G. Harley, T. G. Habetler (2009). Method for multiobjective optimized designs of surface mount permanent magnet motors with concentrated or distributed stator windings. IEEE International Electric Machines and Drives Conference, doi:10.1109/IEMDC.2009.5075225.
[26] K. Deb (2001). Multiobjective optimization using evolutionary algorithms. John Wiley & Sons, England.
[27] N. Srinivas, K. Deb (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation, Vol. 2, Issue 3, pp 221-248, doi:10.1162/evco.1994.2.3.221.
[28] K. Deb (2011). Multiobjective optimization using evolutionary algorithms: an introduction. multiobjective evolutionary optimisation for product design and manufacturing. pp 3–34.
[29] R. Pichot, L. Schmerber, D. Paire, A. Miraoui (2018). Robust BLDC motor design optimization including raw material cost variations. XIII International Conference on Electrical Machines, doi:10.1109/ICELMACH.2018.8507039.