An adaptive multipopulation genetic algorithm for the optimization of active magnetic bearings

Anand S. Reddy¹, Praveen Kumar Agarwal², Satish Chand³

¹Department of Mechanical Engineering, KLE Institute of Technology, Hubballi – 580030, Karnataka State, India.
²Department of Mechanical Engineering, Motilal Nehru National Institute of Technology Allahabad, Prayagraj – 211004, Uttar Pradesh, India.
³Department of Mechanical Engineering, G.L. Bajaj Institute of Technology & Management, Knowledge Park III, Greater Noida-201306, Uttar Pradesh, India.

E-mail: reddyanan@gmail.com

Abstract. Active magnetic bearings (AMBs) are mechatronic systems to support the rotors without any physical contact. Compact AMBs are required for better performance. Present work aims at optimization of AMB for achieving the compact size. Genetic algorithms (GAs) are mostly used for the optimization of such nonlinear design problems. However, GAs with fixed parameters may result in convergence to local optimal solutions and sluggish convergence speed. To remove these deficiencies, the adaptive multipopulation genetic algorithm (AMPGA) is used for the optimization of AMB in the present work. In this algorithm, crossover and mutation probabilities and migration rate are regulated based on population diversity. In addition, real-encoded chromosomes are used. Performance of the designed AMPGA is studied in terms of the convergence speed and optimal solution. It is found that AMPGA converges at a faster rate and gives a better optimal solution than both multipopulation genetic algorithm (MPGA) and simple genetic algorithm (SGA). The optimal value of the AMB size obtained by AMPGA is found to be smaller than that obtained by MPGA and SGA.

1. Introduction

Active magnetic bearing (AMB) is the contactless bearing appropriate for high-speed operations. It is used in high-speed turbo-machines and machine tools [1]. However, in comparison to conventional bearings, AMBs are larger for the same load-carrying capacity and rotor diameter. Compact AMBs have various advantages such as reduction in the magnetic material requirement, lower power losses, efficient operation and minimum space requirement for installation. The compact size of AMB can be obtained by optimizing geometrical and operational parameters of AMB. Geometrical parameters of an eight-pole AMB were optimized using differential evolution algorithm to achieve compact AMB with maximum load-carrying capacity [2]. Similarly, geometrical parameters such as stator iron thickness and stator inner diameter of a hybrid AMB were optimized for blood pumping application using 3-D finite element technique [3]. Optimal design of thrust AMB for higher load capacity with the
least magnet volume was proposed, and the 3-D finite element method was used for design optimization [4].

In recent years, evolutionary optimization techniques, mimicking the natural phenomenon, are becoming more popular for engineering optimization applications [5]. Some of the essential techniques are Genetic Algorithm (GA) [6, 7], Particle Swarm Optimization (PSO) [8, 9], Ant Colony Optimization (ACO) [10, 11] and Artificial Bee Colony (ABC) algorithm [12, 13]. All these optimization algorithms are based on various natural phenomena. However, GAs, relying on the principle of natural genetics, are computationally simple and can provide robust search in complex spaces [14]. In GA, the search space can be explored in many directions simultaneously for optimum solution. Therefore, it can deal with optimization problems where the objective function is linear or nonlinear [15, 16]. Simple genetic algorithm (SGA) was used in the design of radial AMB integrated with control [17]. Minimization of the volume of AMB was considered as the objective function along with the constraints to meet dynamic load-carrying capacities and equivalent stiffness. A double-acting hybrid magnetic thrust bearing was also optimized using GA [18]. The optimization problem was formulated with five design objectives (load capacity, weight, power loss, control input and dynamic performance indices), fourteen design parameters and corresponding constraints. However, GA with fixed parameters (crossover probability, mutation probability etc.) does not always converge to optimum solutions. Further, it requires a longer time for convergence. It is due to the inability of maintaining the population diversity for longer time and thus, results in avoiding the exploration of different regions of the search space. Many methods have been proposed for the adaptive tuning of the GA parameters to find out the better solution. A novel GA, which adapts its crossover, mutation and selection parameters, was developed where the adaptation method was based on the measure of population diversity [19]. The fuzzy adaptive search method for tuning the GA parameters was proposed, and this method was used in parallel GA to find high-quality solutions at a faster rate [20].

An adaptive genetic algorithm was proposed for the optimization of Fuzzy controller of AMB [21]. The adaptive tuning of the GA parameters was carried out based on the value of individual fitness and dispersion degree of population. Rule base and Membership functions of Fuzzy controller were optimized to achieve precise and fast control.

The multipopulation genetic algorithm (MPGA) is the parallel implementation of SGA. This algorithm has also been proposed as an effective method to solve the problem of slow convergence speed of SGA and to enhance the searchability of SGA [22]. In MPGA, the total population is divided into sub-populations called islands and migration operation is performed to migrate some of the good solutions from one island to replace the worst solutions of other islands. Therefore, MPGA is capable of producing high-quality solutions at a faster rate. MPGA was applied to a multimodal function by creating sub-populations in a generation to obtain multiple optimal solutions for providing good diversity [23]. MPGA was also applied in the design of manipulators, in which each population worked on one objective function of a task point to reduce the overall time required for the optimization process [24].

In this paper, AMB parameters such as current, number of coil turns in each coil, winding width, pole face included angle, stator diameter and bearing length are optimized by considering the volume of AMB as an objective function and constraints corresponding to operational and geometrical parameters of AMB. Current and number of turns in the coil affect the load-carrying capacity of AMB because the developed flux densities in the poles are directly proportional to supplied currents and number of turns in the coils. Higher values of currents increase the operating cost, whereas the lower values increase the number of coil turns which further requires more winding space and increase the AMB size. An increase in the winding width decreases the angular space available to accommodate the pole width. This increases the AMB axial length to maintain the same pole face area.
decrease in the winding width increases the winding height to maintain the same winding space and this results in the increased stator outer diameter. Pole face included angle also has similar effects on AMB size. Stator diameter and length directly affects the size of AMB. Therefore, in the present work, these six design parameters are considered for optimization of AMB, and adaptive multipopulation genetic algorithm (AMPGA) is proposed for it. Proposed adaptation scheme for tuning AMPGA parameters is based on population diversity. The optimization results obtained by AMPGA are compared with results of MPGAs and SGA to confirm the efficiency of the proposed optimization algorithm.

2. Nonlinear constrained optimization problem formulation
Radial Eight-pole heteropolar configuration of AMB (Figure 1) with independent poles is considered here for the problem formulation. Here, each pole has identical pole face area and number of coil turns.

![Figure 1](image_url) Cross-section of eight-pole AMB

The optimization problem is formulated for minimizing the volume of AMB to achieve the compact size with objective function as given by Eq. (1).

$$\text{obj} = \frac{\pi}{4} D^2 L$$  \hspace{1cm} (1)

where $D$ is the stator outer diameter and $L$ is the axial length of AMB.

Constraints are defined to attain the required load-carrying capacity with the available saturation flux density of the magnetic material and for effective utilization of the available circumferential space between rotor and stator to accommodate poles and coils without any interference.

2.1. Constraint conditions for load carrying capacity and flux density
Load-carrying capacity is defined as the maximum force produced by AMB in any radial direction [25]. It is taken as the minimum of the maximum forces generated by AMB in all directions. Magnitudes of the magnetic forces generated by an AMB depend upon the orientation in which it is considered. These generated forces along the positive X and Y axes are given by Eqs. (2) and (3) respectively [26].
where \( A \) is the pole face area, \( B \) is the flux density of the \( j \)th pole, \( \mu_0 \) is the permeability of air and \( \theta \) is the pole orientation angle of the \( j \)th pole from the positive X-axis.

Stator and rotor of AMB are made of ferromagnetic materials having relative permeability greater than 1000. Because of this, metal path reluctances are negligible in comparison to air gap reluctances, and therefore, only air gap reluctances are considered in the magnetic circuit of AMB [25]. Further, the sources of excitation are the coils of each pole. The equivalent magnetic circuit for the eight-pole AMB is shown in Figure 2. The flux densities in different poles are obtained by applying the Ampere’s loop law and law of conservation of fluxes as given by Eqs. (4) and (5) [27].

\[
\left( R_j B_j - R_{j1} B_{j1} \right) A = N(I_j - I_{j1}) \quad (j = 1, 2, \ldots, 7)
\]

\[
\sum_{j=1}^{8} B_j = 0
\]

where \( R_j \) is the reluctance of the \( j \)th air-gap, \( N \) is the number of coil turns in each coil and \( I_j \) is the current in the \( j \)th coil.

Air-gap reluctance of \( j \)th pole is given by Eq. (6).

\[
R_j = g_j / A \mu_0
\]

Air-gap length \( g_j \) at \( j \)th pole is given by Eq. (7).

\[
g_j = g_0 - x \cos \theta_j - y \sin \theta_j
\]

where \((x, y)\) is the displaced position of rotor centre and \( g_0 \) is the nominal air-gap length.

Figure 2 Equivalent magnetic circuit for eight-pole AMB

At the steady state, various air-gaps between the rotor and the magnetic poles will be equal to the nominal air-gap. On the basis of this, \( g_1 = g_2 = g_3 = g_4 = g_5 = g_6 = g_7 = g_8 = g_0 \) and therefore, \( R_1 = R_2 = R_3 = R_4 = R_5 = R_6 = R_7 = R_8 = R_0 \). Now Eq. (4) reduces to Eq. (8).

\[
\left( B_j - B_{j1} \right) A R_0 = N(I_j - I_{j1}) \quad (j = 1, 2, \ldots, 7)
\]

where \( R_0 = g_0 / A \mu_0 \).
Eq. (8) gives the relation between the flux densities in different poles of AMB and currents supplied to the coils of respective poles.

Magnitude of the magnetic force generated by AMB varies for different load orientations. For generating the maximum force, poles lying closest to the desired orientation \( \theta \) will have the maximum allowable flux density. The flux densities in the remaining poles are selected to attain the force in the perpendicular direction as zero and to satisfy the law of conservation of fluxes [28]. Based on it, flux density distribution to obtain the maximum force for 90° load orientation is given by Eq. (9).

\[
B_i = -B_z = B_y = B_{sat}, \quad B_i = -B_y = B_z = -B_z = 0
\]  

(9)

where \( B_{sat} \) is the saturation flux density of the magnetic material.

At the nominal air gap, the saturation flux density in a pole is given by Eq. (10).

\[
B_{sat} = \frac{\mu_0 N I_{max}}{g_{20}}
\]  

(10)

where \( I_{max} \) is the maximum magnitude of the current supplied to the coil.

By substituting the above flux density distribution (Eq. (9)) for all the poles in Eq. (3), the maximum force generated by AMB along 90° load orientation is given by Eq. (11).

\[
F_{ymax} = \frac{A B_{sat}^2}{2 \mu_0} (2.613)
\]  

(11)

This maximum generated force is taken as the load capacity of AMB for that particular direction. Similarly, the load capacity of AMB is determined for different angular directions, and the polar plot of load capacity is drawn as shown in Figure 3. The radius of the polar plot indicates the load capacity of AMB in that angular orientation and its value for any angle is specified as a factor of the minimum radius of the polar plot. It can be observed that the minimum load capacity is attained when \( \theta \) lies at the location of the poles. Further, the maximum load capacity is achieved when \( \theta \) lies at the middle of any two consecutive poles. The load-capacity at this location is 1.08 times the minimum load capacity. The minimum load-capacity of the plot is taken as the load-capacity of eight-pole AMB. This value is used to determine the size of AMB, and it is given by Eq. (12).

\[
F_{LC} = \frac{A B_{sat}^2}{2 \mu_0} (2.414)
\]  

(12)

Now, the constraint condition for the load capacity is specified that in any operating condition \( F_{LC} \) has to be greater than or equal to the applied external load, and it is given by Eq. (13).

\[
F_{LC} \geq F_{ext}
\]  

(13)

where \( F_{ext} \) is the applied external load.

Further, the constraint condition for flux density is specified that at any instant, the flux density induced in \( j \)th pole of AMB should not exceed the saturation flux density of the magnetic material of the stator and it is given by Eq. (14).

\[
B_j \leq B_{sat}
\]  

(14)
2.2. Constraint conditions for the coil geometry

All the poles of AMB are considered to be having identical coils with coil geometry detail as shown in Figure 4. The size of the coil is determined by the maximum winding width, $W_w$. It should be less than or equal to the maximum allowable width to avoid interference between two adjacent electromagnets [17].

\[
W_w \leq \beta \left[ \left( r + g_0 + W_p \right) \cos \left( \frac{\alpha_2}{2} \right) + t \right] \tan \left( \frac{1}{2} \alpha_1 + \alpha_2 \right) - \frac{1}{2} W_p \tag{15}
\]
where  \( r \) is the rotor shaft radius, \( W_p \) is the width of the pole, \( \alpha_i \) is the pole face included angle, \( \alpha_z \) is the angle defining the winding width, \( t \) is clearance thickness and \( \beta_1 \) is assembly ratio. The values of \( \alpha_z \) and \( t \) are given by Eqs. (16) and (17) respectively.

\[
\alpha_z = \frac{\pi}{n} - \frac{1}{2} \alpha_i \tag{16}
\]

\[
t = \beta_2 H_w \tag{17}
\]

where \( H_w \) is the height of the winding and \( \beta_1 \) is clearance ratio. \( \beta_1 \) and \( \beta_2 \) are constants less than unity. \( \beta_2 \) is used to prevent \( \alpha_z \) from reaching its maximum value even if both parts of Eq.(15) are equal and \( \beta_2 \) is used to avoid the assembly error of AMB in fabricating processes.

The width of the pole is given by Eq.(18).

\[
W_p = 2(r + g_o) \sin \left( \frac{\alpha_i}{2} \right) \left( 1 - 2 \sin \left( \frac{\alpha_i}{2} \right) \right) \tag{18}
\]

Area available for the winding should be equal to the winding area required, and is given by Eq.(19) [17].

\[
(W_w - t_c)(H_w - 2t_c) = \frac{N \pi (r_w + t_m)^2}{w_f} \tag{19}
\]

where \( r_w \) is the radius of the winding wire used in the coil winding, \( t_m \) is the thickness of the chemical insulation coating provided between the successive windings to avoid winding short-circuit, \( w_f \) is the winding factor and it defines the fraction of the copper wire cross-sectional area in the total winding cross-sectional area and \( t_c \) is the thickness of coil casing which is provided to avoid direct contact of the coil with the stator. The height of the winding is given by Eq.(20) [17].

\[
H_w = - (r + W_p + g_o) \cos \left( \frac{\alpha_i}{2} \right) - t_c + \left[ \frac{D}{2} - W_p \right]^2 - \left( \frac{W_p}{2} + W_w \right)^2 \right)^{1/2} \tag{20}
\]

In this paper, AMPGA is proposed to solve this nonlinear constraint single objective optimization problem, with the objective function defined by Eq.(1), three inequality constraints defined by Eqs.(13), (14), (15), and one equality constraint defined by Eq.(20).

3. Adaptive multipopulation genetic algorithm (AMPGA)
AMPGA is the parallel implementation of SGA. The flow chart for the optimization of AMB using AMPGA is shown in Figure5. The p sub-populations of \( n_{pop} / p \) individuals are simultaneously processed to find out the best optimal solutions of the objective function. Further, real-encoded chromosomes are used in the present work to avoid the coding and decoding stages of binary encoding of chromosomes [29].

In GAs, the initial population is generated randomly. The objective function value for each individual of the population indicates the performance of that individual in the problem domain. For determining the relative performance of individuals in a population, another function called fitness function is usually used, and is given by Eq.(21)[5].

\[
f_{fit} = \frac{1}{1 + f_{obj}} \tag{21}
\]

where \( f_{fit} \) is the fitness function and \( f_{obj} \) is the objective function.

However, an individual’s performance in a given generation is not restricted based on fitness value. It may result in domination of the reproduction step by the highly fit individual in early generations and
may further cause rapid convergence to possibly sub-optimal solutions. To avoid this situation, a rank-based fitness assignment scheme was suggested [30]. Here, ranks of the individuals in the population are used to assign the fitness values to respective individuals. Initially, the objective function values of various individuals are used to sort the respective individuals. The sorted list consists of the least fit individual in the first position and the fittest individual in the last place. The location of an individual in this sorted list is then used to assign a corresponding fitness value as given by Eq. (22) [15].

\[
f_{p_2}(i_{pos}) = 2 - SP + 2 \times (SP - 1) \times (i_{pos} - 1) / (n_{pop} - 1) \tag{22}
\]

where \( i_{pos} \) is the position of the individual in the sorted population, \( n_{pop} \) is the population size and \( SP \) is the selective pressure. In this paper, the rank-based method is used for fitness assignment with selective pressure value as 2.

Thereafter, three genetic operators namely reproduction, crossover and mutation are used to determine the population for the next generation. Reproduction operation is also identified as the selection operation [5]. In this, a mating pool for a population is formed by selecting the strings with higher fitness values. Various reproduction methods have been used for GA based optimization. Among them, roulette wheel selection, tournament selection and stochastic universal sampling (SUS) are mostly preferred. SUS selection method provides the minimum spread for an individual. Spread is defined as the possible number of times an individual may be selected. Therefore, it results in slow convergence but preserves diversity and results in a successful search. In the present research work, SUS technique is used. In this method, individuals are represented as continuous sectors such that the size of each individual’s sector is equal to its probability of selection. This method uses \( N_{ad} \) equally spaced pointers, where \( N_{ad} \) is the number of selections required [15]. The population is shuffled randomly, and the position of the first pointer is marked by a randomly generated number in the range \([0, 1/N_{ad}]\). For the selection of \( N_{ad} \) individuals, the distance between the pointers is given as \( 1/N_{ad} \). Figure 6 shows the application of SUS selection process to a population of five individuals with five numbers of selections required. The individual segments are shuffled randomly, and equally spaced pointers are placed. Random placement of first pointer results in the selection of all individuals except the third individual to form the mating pool. The second individual is selected twice as it is having the highest probability of selection amongst all other individuals.

The mating pool generated by reproduction operation is then used to create new offsprings or children by crossover and mutation operations. Crossover is the process of producing two children from two-parent solutions, and is applied to the mating pool to create better offsprings. The next step involves randomly adding new information by using mutation operation for avoiding the trapping of solution at local optima [15]. The repeated use of reproduction and crossover operators may result in the homogenous population and mutation will add the required diversity in it [5]. In the present work, arithmetical crossover and uniform mutation operators are used. Arithmetical crossover is performed by determining a linear combination of the two randomly selected parents \( x_i \) and \( x_k \) from the initial population, and the two children are given by Eq. (23) [31].

\[
\begin{align*}
    x'_j &= \delta x_j + (1 - \delta) x_k \\
    x'_k &= (1 - \delta) x_j + \delta x_k
\end{align*}
\tag{23}
\]

where \( \delta \) is a randomly generated positive constant between 0 and 1.

In uniform mutation operation, a new individual replaces a randomly selected individual \( x'_j \) from the initial population as given by Eq. (24) [31].

\[
x'_j = x_{j_{min}} + \delta (x_{j_{max}} - x_{j_{min}})
\tag{24}
\]
where \( x_{\text{min}} \) and \( x_{\text{max}} \) are the lower and the upper bounds of the variable \( x_i \) respectively.

Crossover and mutation operations are not executed for all the individuals in the mating pool to preserve some of the good individuals in the mating pool. Crossover probability \( P_c \) and mutation probability \( P_m \) are used to define the percentage of the population involved in crossover and mutation operations respectively.

Further, in MPGA, an interchange of individuals between various sub-populations is periodically performed through migration operation to help the creation of good individuals. These individuals are operated on by genetic operators to provide sub-populations for the next generation representing the still unexplored regions. Migration operation involves replacing a few selected individuals with the best fitness in a sub-population with the worst individuals in the other sub-populations. The selection of these individuals is performed at random. The fraction of the population to be migrated from each sub-population is defined by migration rate \( M_i \), which helps in maintaining the level of diversity inside each sub-population. Figure 7 shows the scheme of migration operation used in MPGA.

### 3.1. Proposed adaptive tuning of MPGA parameters

In AMPGA, both \( P_c \) and \( P_m \) of each sub-population are regulated adaptively to attain superior search-ability. Here, population diversity of each sub-population of the previous generation is used for determining \( P_c \) and \( P_m \) values for the respective sub-population of a generation as given by Eqs.(25) and (26) respectively.

\[
P_c^{(\text{gen})} = P_c \times \left( \frac{f_{\text{obj}(\text{min})}}{f_{\text{obj}(\text{avg})}} \right)_{\text{gen-1}}
\]

\[
P_m^{(\text{gen})} = P_m \times \left( 1 - \left( \frac{f_{\text{obj}(\text{min})}}{f_{\text{obj}(\text{avg})}} \right)_{\text{gen-1}} \right)
\]

where \( P_c \) and \( P_m \) are the initial values of crossover and mutation probability for each sub-population for the first generation. \( f_{\text{obj}(\text{min})} \) and \( f_{\text{obj}(\text{avg})} \) are the minimum and average values of the objective function of the corresponding sub-population. \( f_{\text{obj}(\text{avg})} \) is given by Eq.(27).

\[
f_{\text{obj}(\text{avg})} = \frac{\sum_{i=1}^{i=n_{\text{pop}}/p} f_{\text{obj}(i)}}{n_{\text{pop}}/p}
\]

The value of \( \left( \frac{f_{\text{obj}(\text{min})}}{f_{\text{obj}(\text{avg})}} \right) \) will be minimal in the early generations. Therefore, \( P_c \) value should be selected as the minimum value and \( P_m \) value should be selected as the maximum value of the respective ranges in the beginning. The value of \( \left( \frac{f_{\text{obj}(\text{min})}}{f_{\text{obj}(\text{avg})}} \right) \) approaches unity with the increase in number of generations. It will lead to an increase in \( P_c \) value and a decrease in \( P_m \) value with a corresponding increase in number of generations. Here, \( P_c \) and \( P_m \) values are varying in the range \([0.8, 0.2]\).
- Objective function with constraints
- Number of generations ($N_{gen}$)
- Crossover probability ($P_c$)
- Mutation probability ($P_m$)
- Migration rate ($M_r$)

Inputs

Randomly generate ($p$) sub-population of ($n_{pop}/p$) individuals for first generation

$1^{st}$ sub-population $2^{nd}$ sub-population $3^{rd}$ sub-population $\ldots p^{th}$ sub-population

Calculate the objective function value for each individual

Calculate the fitness value for each individual

Adaptive tuning

Calculate the new values for GA parameters ($P_c$, $P_m$ and $M_r$) based on population diversity

Migration

Termination condition is met

Best individual with the corresponding objective function value

No

Reproduction

Crossover

Mutation

$p^{th}$ sub-population for next generation

Reproduction

Crossover

Mutation

$p^{th}$ sub-population for next generation

Reproduction

Crossover

Mutation

$p^{th}$ sub-population for next generation

Reproduction

Crossover

Mutation

$p^{th}$ sub-population for next generation

Figure 5 AMPGA based AMB optimization flow chart

Figure 6 Stochastic universal sampling (SUS) selection
Further, the migration rate for a generation is calculated based on the diversity in optimal solutions obtained by all the sub-populations of the preceding generation, and is given by Eq. (28).

\[ M_{r(\text{gen})} = M_r \times \left(1 - \left(\frac{F_{\text{obj(min)}}}{F_{\text{obj(avg)}}}\right)_{\text{gen-1}}\right) \]  

where \( M_r \) is the migration rate value taken for the first generation, \( F_{\text{obj(min)}} \) is the minimum value of the objective function in the optimal solutions obtained by all the sub-populations of the previous generation and \( F_{\text{obj(avg)}} \) is the mean of average values obtained for all the sub-populations of the previous generation and it is given by Eq. (29).

\[ F_{\text{obj(avg)}} = \frac{\sum_{i=1}^{p} f_{\text{obj(avg)}}(i)}{p} \]  

In early generations, the value of \( \left(\frac{F_{\text{obj(min)}}}{F_{\text{obj(avg)}}}\right) \) will be minimal. Therefore, \( M_r \) value should be selected as the maximum value of the respective range in the beginning. The value of \( \left(\frac{F_{\text{obj(min)}}}{F_{\text{obj(avg)}}}\right) \) approaches unity with the increase in the number of generations. This will lead to a decrease in \( M_r \) value with the increase in number of generations. Here, the value of \( M_r \) is varying in the range [0.8, 0.2].

The individuals of the new population created in such a way are stored, and the objective function values of this new population are determined. These values indicate the fitness of the solutions of the new generations. If a better solution is obtained for a new generation, it is stored as the best solution. This process is continued until the number of generations reaches the defined upper limit on number of generations. After that, the process converges by picking the best individual out of all sub-populations of that generation along with the resultant objective function value.

### 3.2. Dynamic penalty function for handling constraints

The penalty function method is used to handle the constraints. In this method, a significant penalty term equivalent to the constraint violation is included in the objective function. It results in elimination of all the infeasible individuals from the search space. The objective function with inclusion of the penalty function for solving nonlinear constrained optimization problem is given by Eq. (30).

\[ f'_{\text{obj}} = f_{\text{obj}} + f_{\text{penalty}} \]  

where \( f_{\text{penalty}} \) is the dynamic penalty function, and is given by Eq. (31) [32].

\[ f_{\text{penalty}} = \sum_{a=1}^{m_a} G_a \left[ g_a(x) \right] + H_b \sum_{b=1}^{m_b} h_b(x) \]
where, \( g_a(x) = \begin{cases} 0, & g_a(x) \leq 0 \\ g_a(x), & g_a(x) > 0 \end{cases} \) and \( h_b(x) = \begin{cases} 0, & -\varepsilon \leq h_b(x) \leq \varepsilon \\ |h_b(x)|, & \text{otherwise} \end{cases} \)

where \( G_a \) is the penalty parameter of an inequality constraint, \( H_b \) is the penalty parameter of \( b \)th equality constraint, \( n_{eq} \) is the number of inequality constraints, \( n_{eq} \) is the number of equality constraints and \( \varepsilon \) is a small positive quantity representing the allowable error in solution for satisfying equality constraint. In the present work, \( \varepsilon \) value is taken as \( 10^{-6} \).

The values of \( G_a \) and \( H_b \) play a vital role to find a reasonable solution. If the values of \( G_a \) and \( H_b \) make the \( f_{\text{penalty}} \) value small compared to \( f_{\text{obj}} \) value, then the resulting solution may not be the constrained optimal solution. Whereas if the values of \( G_a \) and \( H_b \) make the \( f_{\text{penalty}} \) value very large compared to \( f_{\text{obj}} \) value, then GA may find a local optimum. Therefore, suitable selection of the problem-specific values is required to obtain the desired solution. In the present work, the values of penalty parameters \( G_a \) and \( H_b \) are taken as \( 10^{10} \).

4. Results and Discussion

The results for optimization of eight-pole AMB are presented in this section. AMPGA, MPGA and SGA developed in MATLAB® programming environment are applied for the optimization of AMB volume with the three inequality constraints and one equality constraint. The results obtained by the AMPGA are compared with the MPGA and SGA to evaluate its performance to search for better optimal solutions.

The AMB specifications considered in this paper have been taken from literature (Chang and Chung, 2002).

\[ F_{\text{max}} = 200 \text{ N}, \quad B_{\text{sat}} = 1.2 \text{ T}, \quad g_0 = 0.5 \text{ mm}, \quad r = 30 \text{ mm}, \quad r_w = 0.23 \text{ mm}, \quad \mu_0 = 4\pi \times 10^{-7} \text{ H/m}, \quad t_{\text{ins}} = 0.02 \text{ mm}, \quad \beta_1 = 0.9, \quad \beta_2 = 0.1, \quad t_c = 1 \text{ mm}, \quad w_j = 0.7. \]

In the present optimization problem, six design variables are considered, and the assumed lower and upper limit values for these design variables are given in Table 1. Two cases are considered with 100 and 200 \( N_{\text{gen}} \). The values of optimization parameters for SGA and MPGA are given in Table 2.

Migration operation in MPGA and AMPGA is performed for every five generations.

| Table 1 Initial bound of the design variables |
|-----------------------------------------------|
| Design variables | \( I \) (A) | \( D \) (mm) | \( W_w \) (mm) | \( L \) (mm) | \( \alpha_i \) (rad) | \( N \) |
|-------------------|-----------|-----------|-------------|-----------|----------------|-------|
| Lower limit       | 0.1       | 100       | 4           | 10        | 0.1            | 100   |
| Upper limit       | 1         | 200       | 12          | 25        | 0.6            | 500   |

| Table 2 Optimization parameters |
|----------------------------------|
|                               | \( n_{\text{pop}} \) | \( P_c \) | \( P_m \) | \( P \) | \( M \) |
| SGA                        | 100       | 0.8      | 0.2       | ----   | ----   |
In the present optimization problem, the objective function (Eq. (1)) is dependent on the variables $D$ and $L$ of AMB. Therefore, the explorations of the search space by SGA, MPGA and AMPGA for these two variables for 100 generations are shown in Figures 8, 9 and 10 respectively. It can be observed from Figure 8 that SGA convergence speed is slow and most of the solutions are not converging towards the optimum solution even after the final generation. Whereas, the results corresponding to MPGA (Figure 9) and AMPGA (Figure 10) show the convergence of the solutions for the variables $D$ and $L$ towards the best possible solution at a faster rate. However, the best solution of variables $D$ and $L$ obtained by AMPGA results in a smaller volume of the AMB than the corresponding best solutions obtained by both MPGA and SGA. It shows the better performance of AMPGA in comparison to SGA and MPGA.

**Figure 8** Exploration of the search space by SGA for variables $D$ and $L$ for 100 generations (a) after 1st generation (b) after 10th generation (c) after 50th generation (d) after final generation
\textbf{Figure 9} Exploration of the search space by MPGA for variables $D$ and $L$ for 100 generations (a) after 1$^{\text{st}}$ generation (b) after 10$^{\text{th}}$ generation (c) after 50$^{\text{th}}$ generation (d) after final generation

\textbf{Figure 10} Exploration of the search space by AMPGA for variables $D$ and $L$ for 100 generations (a) after 1$^{\text{st}}$ generation (b) after 10$^{\text{th}}$ generation (c) after 50$^{\text{th}}$ generation (d) after final generation

\textit{Crossover probability, mutation probability and migration rate} in AMPGA are tuned as explained in section 3.1. Values of \textit{crossover probability}, \textit{mutation probability} of a subpopulation and \textit{migration rate} of AMPGA with respect to generations are plotted in Figures 11, 12 and 13 respectively. Value of \textit{crossover probability} is lower in the early generations, and it increases as the generations increases and solutions converge towards the optimal global solution. The values of \textit{mutation probability} and \textit{migration rate} are higher in the early generations, and their values decrease as the solutions converge towards the optimal global solution to avoid the breaking of schema relation of excellent individuals.

\textbf{Figure 11} Crossover probability variation of a subpopulation in AMPGA
The evaluations of performance of SGA, MPGA and AMPGA with variation in number of generations are shown in Figures 14 and 15 for 100 and 200 $N_{gen}$ respectively. The corresponding optimized values of the variables for the two cases are given in Table 3. The results provided by all the algorithms are improved with the increase in the number of generations and result in smaller AMB volume. However, in both cases, the optimum value of the bearing volume obtained by AMPGA is smaller than those given by SGA and MPGA. The optimal value of the AMB volume for 200 $N_{gen}$ obtained by AMPGA is found to be 0.8% smaller than that obtained by MPGA and 1.31% smaller than that obtained by SGA.
The results by SGA and MPGA are obtained by manual tuning of respective parameters, and their best results are compared with that by AMPGA. It shows that without requiring manual tuning, AMPGA can provide similar/improved results which are attained with several refinements of optimization parameters with human knowledge. The improved result by AMPGA is due to the effective utilisation of the available space between the rotor and the stator. It is observed that the value of the angle $\alpha_1$ is reduced resulting in the reduction of pole width. Therefore, the diameter of the bearing stator obtained by AMPGA is smaller than MPGA and SGA for the same length of the bearing. Further, to check the convergence speed of designed AMPGA in comparison with MPGA and SGA, two cases are considered. In the first case, optimal solutions are obtained for the same number of generations and the time taken by all three algorithms are compared as given in Table 4. It can be observed that the time taken by AMPGA is more than MPGA and SGA. This is because AMPGA involves a higher number of computations than MPGA and SGA. However, the optimal solution obtained by AMPGA is nearly equal to MPGA and 0.9% less than SGA for 100 generations. The convergence speed of SGA is also improved with the rise in number of generations. However, the AMB volume obtained by AMPGA is still better one. In the second case, to achieve the same optimal solution, the number of generations required and the time taken by all the three GAs are noted as given in Table 5. In this case, AMPGA requires approximately 42% and 62% lesser number of generations and 25% and 51% less time to obtain the same optimal solutions ($1.8580 \times 10^5$ and $1.8267 \times 10^5$ mm$^3$) as obtained by MPGA and SGA respectively. This convergence study is carried out in
MATLAB® programming environment on a 2.40 GHz Intel Core i3 PC with 3 GB RAM under 64 bit Windows 7 operating system. From the convergence study, it is found that the convergence speed of the designed AMPGA is superior than that of MPGA and SGA.

Table 3 Optimum values of design variables

| Design Variables | 100 Generations | 200 Generations |
|------------------|-----------------|-----------------|
|                  | SGA             | MPGA            | AMPGA           | SGA             | MPGA            | AMPGA           |
| x₁ (= I ) (A)    | 0.9625          | 1.00            | 0.9967          | 0.9998          | 1.00            | 1.00            |
| x₂ (= D ) (mm)   | 153.8           | 146.3           | 153.1           | 152.5           | 152.1           | 151.5           |
| x₃ (= Wₑ ) (mm)  | 10.6            | 10.7            | 10.4            | 10.6            | 10.6            | 10.6            |
| x₄ (= L ) (mm)   | 10.0            | 11.0            | 10.0            | 10.0            | 10.0            | 10.0            |
| x₅ (= αₑ ) (rad) | 0.3281          | 0.3035          | 0.3278          | 0.3274          | 0.3256          | 0.3233          |
| x₆ (= N )        | 491             | 477             | 475             | 476             | 476             | 478             |
| Volume (mm³)     | 1.8580 × 10⁵    | 1.8492 × 10⁵    | 1.8411 × 10⁵    | 1.8267 × 10⁵    | 1.8172 × 10⁵    | 1.8028 × 10⁵    |

Table 4 Comparison of convergence speed with respect to fixed generations

| Number of generations | AMB volume achieved (mm³) | Time taken in seconds |
|-----------------------|---------------------------|-----------------------|
|                       | SGA           | MPGA             | AMPGA             | SGA           | MPGA             | AMPGA             |
| 100                   | 1.8580 × 10⁵    | 1.8492 × 10⁵    | 1.8411 × 10⁵    | 0.28          | 0.31             | 0.34             |
| 200                   | 1.8267 × 10⁵    | 1.8172 × 10⁵    | 1.8028 × 10⁵    | 0.56          | 0.62             | 0.68             |

Table 5 Comparison of convergence speed with respect to a fixed solution

| Volume of AMB (mm³) | Number of generations | Time taken in seconds |
|--------------------|-----------------------|-----------------------|
|                    | SGA       | MPGA       | AMPGA       | SGA       | MPGA       | AMPGA       |
| 1.8580 × 10⁵       | 98        | 75         | 57          | 0.27      | 0.23       | 0.21        |
| 1.8267 × 10⁵       | 196       | 91         | 74          | 0.55      | 0.28       | 0.27        |

The geometries of AMB are drawn from the values of the variables obtained by the SGA, MPGA and AMPGA for 200 Nₚₑₑ and are shown in Figure16. It is found that there is no geometrical interference between the windings and all the windings are firmly accommodated between the poles in all the three cases.
5. Conclusions

In this paper, optimization of AMB for achieving a compact size is presented. The nonlinear constrained optimization problem is formulated with three inequality constraints and one equality constraint for the minimization of the AMB volume. To handle this nonlinear constrained optimization problem, AMPGA is proposed over MPGA and SGA. Adaptation in MPGA is achieved by tuning its parameters such as crossover probability, mutation probability and migration rate based on the population diversity. Results of AMPGA are compared with that of MPGA and SGA. Convergence of the AMPGA is studied with respect to fixed number of generations and same optimal solutions. Time taken by AMPGA to complete fixed number of generations is more. However, the optimal solution obtained is better in AMPGA compared to MPGA and SGA. For 100 generations, the optimal solution obtained by AMPGA is found to be nearly equal to MPGA and 0.9% less than SGA. Similarly, for 200 generations, the optimal solution obtained by AMPGA is found to be 0.8% better than that obtained by MPGA and 1.31% better than that obtained by SGA. Further, AMPGA takes approximately 42% and 62% lesser number of generations and 25% and 51% less time to obtain the same optimal solutions as obtained by MPGA and SGA respectively.

References

[1]. Schweitzer G., Bleuler H. and Traxler A., Active Magnetic Bearing, Zurich, Switzerland vdf Hochschulverlag AG. 1994.

[2]. Stumberger G., Dolinar D., Pahner U. and Hameyer K., Optimization of radial active magnetic bearings using the finite element technique and the differential evolution algorithm, IEEE Transactions on Magnetics. 2000; 36 (4): 1009-1013.

[3]. Cheng S., Olles M. W., Burger A. F., and Day S. W., (2011), Optimization of a hybrid magnetic bearing for a magnetically levitated blood pump via 3-D FEA, Mechatronics. 2011; 21: 1163–1169.

[4]. Yoo S. Y., Kim W. Y., Kim S. J., Lee W. R., Bae Y. C., and Noh M. (2011), Optimal design of non-contact thrust bearing using permanent magnet rings, International Journal of Precision Engineering and Manufacturing. 2011; 12 (6): 1009-1014.

[5]. Deb K., Optimization for Engineering Design, Algorithms and Examples, Prentice-Hall of India Private Limited, New Delhi. 1995.
[6]. Ozturk N., and Celik E., Speed control of permanent magnet synchronous motors using fuzzy controller based on genetic algorithms, Electrical Power and Energy Systems. 2012; 43: 889-898.

[7]. Crevecoeur G., Sergeant P., Dupre L. and Walle R. V., A Two-Level Genetic Algorithm for Electromagnetic Optimization, IEEE Transactions on Magnetics. 2010; 46 (7).

[8]. Yushu B., and Zhihui G., Parameter optimization of controllable local degree of freedom for reducing vibration of flexible manipulator, Chinese Journal of Aeronautics. 2013; 26(2): 487-494.

[9]. Chen C. C., Two-layer particle swarm optimization for unconstrained optimization problems, Applied Soft Computing.2011; 11: 295-304.

[10]. Ding Q., Hu X., Sun L., and Wang Y., An improved ant colony optimization and its application to vehicle routing problem with time windows, Neurocomputing. 2012; 98: 101-107.

[11]. Neto R. F. T., and Filho M. G., An ant colony optimization approach to a permutational flowshop scheduling problem with outsourcing allowed, Computers & Operations Research.2011; 38: 1286-1293.

[12]. Derakhshan S., Pourmahdavi M., Abdolahnejad E., Reihani A., and Ojaghi A., (2013) Numerical shape optimization of a centrifugal pump impeller using artificial bee colony algorithm’, Computers & Fluids. 2013; 81: 145-151.

[13]. Yildiz A., R., Optimization of cutting parameters in multi-pass turning using artificial bee colony-based approach’, Information Sciences. 2013; 220: 399–407.

[14]. Goldberg D. E., Genetic Algorithms in Search, Optimization and Machine Learning, New York, Addison-Wesley. 1989.

[15]. Sivanandam S. N. and Deepa S. N., Introduction to Genetic Algorithms, Springer, Berlin Heidelberg, New York. 2008.

[16]. Xin-She Y., Engineering Optimization an Introduction with Metaheuristic Application, John Wiley & Sons, Inc. 2010.

[17]. Chang H., and Chung, S. C., Integrated design of radial active magnetic bearing systems using genetic algorithms, Mechatronics. 2002; 12(1): 19–36.

[18]. Rao J. S. and Tiwari R., Design optimization of double-acting hybrid magnetic thrust bearings with control integration using multi-objective evolutionary algorithms, Mechatronics. 2009; 19: 945-964.

[19]. Ginley B. M., Maher J., O’Riordan C., and Morgan F., (2011), Maintaining healthy population diversity using adaptive crossover, mutation, and selection, IEEE Transactions on Evolutionary Computation. 2011; 15(5).

[20]. Maeda Y., Ishita M. and Li Q., Fuzzy adaptive search method for parallel genetic algorithm with island combination process, International Journal of Approximate Reasoning.2006; 41: 59–73.

[21]. Chen H. C., Optimal fuzzy PID controller design of an active magnetic bearing system based on adaptive genetic algorithms, Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming. 2008; 2054-2060.
[22]. Keyong, S., Fei, L., Beiyan, J., Hongyan, Z., Miaomiao, T. and Wencheng, L., Multipopulation genetic algorithm with adaptive search area, International Conference on Intelligent Control and Information Processing, Dalian, China. 2010.

[23]. Siarry P., Petrowski A. and Bessaou M., A multipopulation genetic algorithm aimed at multimodal optimization, Advances in Engineering Software. 2002; 33: 207-213.

[24]. Kim J. O., and Khosla P. K., A multi-population genetic algorithm and its application to design of manipulators, Proceeding of IEEE/RSJ International Workshop on Intelligent Robotics and Systems (IROS'92), Raleigh, NC. 1992.

[25]. Agarwal P. K. and Chand S., Fault-tolerant control of three-pole active magnetic bearing, Expert Systems with Applications. 2009; 36: 12592–12604.

[26]. Agarwal P. K. and Chand S., Fuzzy logic control of three-pole active magnetic bearing system, International Journal of Modelling Identification and Control. 2011; 12 (4).

[27]. Maslen E. H., and Meeker D., Fault tolerance of magnetic bearings by generalized bias current linearization, IEEE Transactions on Magnetics.1995; 31(3): 2304–2314.

[28]. Maslen E., Magnetic bearings, Department of Mechanical and Aerospace Engineering, University of Virginia.http://www.people.virginia.edu/~ehm7s/MagneticBearings/home.html. 2000.

[29]. Johnson J. M., and Rahmat S. Y., Genetic algorithms in engineering electromagnetics, IEEE Antennas and Propagation Magazine.1997; 39 (4).

[30]. Baker J. E., Adaptive selection methods for genetic algorithms, International Conference on Genetic Algorithms. 1985; 1: 101-111.

[31]. Ben H. N., Tounsi, S., Neji, R. and Sellami, F.: Real coded genetic algorithm for permanent magnet motor mass minimization for electric vehicle application, Third International Symposium on Computational Intelligence and Intelligent Informatics (ISCIII), Agadir, Morocco. 2007: 28-30.

[32]. Yokota T., Taguchi, T. And Gen, M. (1998), A solution method for optimal weight design problem of 10 bar truss using genetic algorithms, Computers and Industrial Engineering. 1998; 35(1-2): 367-372.