Appl Intell (2007) 27:101–111
DOI 10.1007/s10489-006-0022-2

A comparative study of stochastic optimization methods in electric motor design

Tea Tušar · Peter Korošec · Gregor Papa · Bogdan Filipič · Jurij Šilc

Published online: 13 November 2006

Abstract The efficiency of universal electric motors that are widely used in home appliances can be improved by optimizing the geometry of the rotor and the stator. Expert designers traditionally approach this task by iteratively evaluating candidate designs and improving them according to their experience. However, the existence of reliable numerical simulators and powerful stochastic optimization techniques make it possible to automate the design procedure. We present a comparative study of six stochastic optimization algorithms in designing optimal rotor and stator geometries of a universal electric motor where the primary objective is to minimize the motor power losses. We compare three methods from the domain of evolutionary computation, generational evolutionary algorithm, steady-state evolutionary algorithm and differential evolution, two particle-based methods, particle-swarm optimization and electromagnetism-like algorithm, and a recently proposed multilevel ant stigmergy algorithm. By comparing their performance, the most efficient method for solving the problem is identified and an explanation of its success is offered.

Keywords Electric motor · Geometry parameters · Power losses · Numerical simulation · Stochastic optimization · Empirical study

T. Tušar · B. Filipič (✉)
Department of Intelligent Systems, Jožef Stefan Institute,
Jamova 39, SI-1000 Ljubljana, Slovenia
e-mail: bogdan.filipic@ijs.si

T. Tušar
e-mail: tea.tusar@ijs.si

P. Korošec · G. Papa · J. Šilc
Department of Computer Systems, Jožef Stefan Institute,
Jamova 39, SI-1000 Ljubljana, Slovenia
e-mail: peter.korosec@ijs.si

G. Papa
e-mail: gregor.papa@ijs.si

J. Šilc
e-mail: jurij.silc@ijs.si

1 Introduction

Daily used home appliances, such as vacuum cleaners and mixers, as well as home used power tools, such as drills and saws, are most often powered by universal electric motors (UM) [17]. This type of motor is widely used because it exhibits high power despite of its small size, provides high starting and running torques, and is inexpensive to manufacture. For home appliances and power tools it is also crucial that the motor is energy efficient, i.e., its input power should be as low as possible, while satisfying the user needs by providing sufficient output power. The ratio of the output power to the input power defines the efficiency of the motor. The efficiency can be improved by reducing the power losses in the motor that originate in the iron and the copper. An approach to reducing the power losses is to optimize the geometry of the rotor and the stator. Due to the high magnetic saturation of the iron in a UM, this optimization task is non-linear.

In a conventional design of an electric motor, the initial estimation for the geometry of the rotor and the stator is made by an experienced engineer. The suitability of this geometry is then usually analyzed by means of numerical simulation of the electromagnetic field. The manual procedure is repeated until the satisfied evaluation results are obtained. The advantage of this approach is that with their experience the engineers can significantly influence the progress of the design process and react intelligently to any noticeable

electromagnetic response with proper geometry redesign. However, this conventional design approach can be upgraded with stochastic optimization techniques which, in connection with reliable numerical simulators, allow for highly automated design process where the need for an experienced engineer to navigate the process is significantly reduced.

Optimization of the electric motor geometry has already been addressed in the literature. The reported approaches differ in geometry description and applied optimization methods.

In a study by Puternicki and Rudnicki [15], the technical optimization system was used that produces a great number of feasible solutions. The applied method was based on a phase diagram. The design process was oriented towards the synthesis of magnetic circuit dimensions and motor winding parameters. A set of feasible solutions was defined through design parameters specifying the core geometry, dimensions of the magnetic circuit and flux density components. In the design process each parameter was varied with stepping variation. The goal of design was to find an optimum geometry of the motor magnetic circuit corresponding to given flux density distribution under rated current. The optimization was performed with deterministic sequential quadratic programming, which is a generalization of Newton's method for unconstrained optimization. At the end a design was checked for technological and standardization requirements.

Papa and coworkers [12] approached the motor design optimization with a problem-specific genetic algorithm. The main goal was to replace the manual optimization procedure with an automated one capable of escaping from local optima and finding the global optimum. The design optimization system integrated the genetic algorithm with the finite-element method program used to simulate and evaluate each candidate solution. The input parameters referred to the dimensions of the initial engineering design, while the outputs of the system were the dimensions of the optimized design and its reduced power losses. The geometry parameters were optimized (1) with no material cost constraints, and (2) with constraints on the outer motor dimensions to ensure the same material costs as in the initial engineering design. For these two tasks the solutions with 30 and 25% reduction of power losses were found, respectively.

Shaked [18] employed a genetic algorithm-based optimization methodology for structural optimization and torque ripple minimization of a motor. The outer volume of the motor was used as the objective function and selected geometric quantities were regarded as the optimization variables. For the speed up of the optimization process the genetic algorithm was combined with the Simplex method.

In this paper we provide a comparative study of six stochastic optimization methods in designing optimal UM rotor and stator geometries where the primary objective is to minimize the motor power losses. We compare three methods from the domain of evolutionary computation, generational evolutionary algorithm, steady-state evolutionary algorithm and differential evolution, two particle-based methods, particle-swarm optimization and electromagnetism-like algorithm, and a recently proposed multilevel ant stigmergy algorithm. All algorithms are tested in connection with a numerical simulator of candidate designs that makes it possible to evaluate the quality of designs.

From the no-free-lunch theorems for optimization [20] it is known that there is no single algorithm that would outperform all others on all optimization problems. However, in practice we are challenged with particular problems having specific characteristics, and typically certain algorithms are better in solving them than others. Hence, it is important to identify suitable algorithms for a particular problem and possibly understand why they perform well. The study presented in the paper is of this kind. It deals with a high-dimensional nonlinear engineering optimization problem with low proportion of feasible solutions in the search space. By comparing the performance of available optimization methods, the most efficient method for solving it is identified and an explanation of its success is offered. An additional practically important aspect of the work is the comparison of the results with the original engineering UM design created by domain experts and used in motor production before this study was performed.

The paper is further organized as follows. In Section 2 the design of electric motors is outlined, including the role of power losses and geometry parameters in the design procedure, and a numerical simulation tool capable of evaluating candidate designs. Section 3 describes the applied optimization methods. Numerical experiments and the obtained results are presented in Section 4, where the applied methods are evaluated with respect to their performance, and the resulting designs from the point of view of minimizing the losses and suitability for regular production. The paper concludes with a summary of the findings of this study and directions of future work.

2 Electric motor design

2.1 Power losses and motor efficiency

The efficiency of a UM is defined as the ratio of the output power to the input power and depends on various power losses. They include copper losses, iron losses, and additional losses, such as brush losses, and losses due to ventilation and friction.

The overall copper losses occurring in the rotor and stator slots are as follows:

$$P_{\mathrm{Cu}} = \sum_i (J^2 A \rho l_{\mathrm{turn}})_i, \tag{1}$$

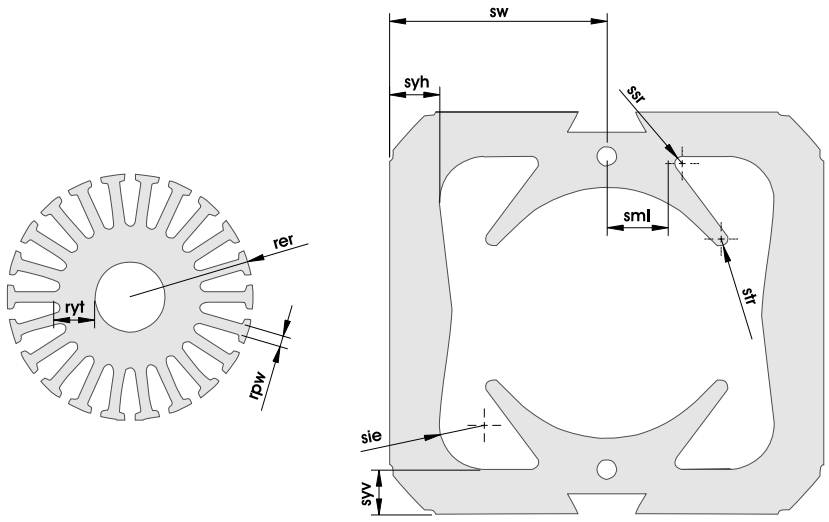

Fig. 1 Rotor and stator geometry parameters

where i stands for each slot, J is the current density, A the slot area, ρ the specific resistance of copper and l_{turn} the length of the winding turn.

The calculation of the iron losses is less exact because of the non-linear magnetic characteristic. The iron losses are of two types: the hysteresis losses and the eddy-current losses. Consequently, iron losses in the motor can be expressed by the following equation:

$$P_{\text{Fe}} = k_{\text{e}} B^2 f_{\text{rot}}^2 m_{\text{rot}} + k_{\text{e}} B^2 f_{\text{stat}}^2 m_{\text{stat}} + k_{\text{h}} B^2 f_{\text{stat}} m_{\text{stat}}, \quad (2)$$

where k_{e} is the eddy-current material constant at 50 Hz, k_{h} the hysteresis material constant at 50 Hz, B the maximum magnetic flux density, f_{rot} the frequency of the magnetic field density in the rotor, f_{stat} the frequency of the magnetic field density in the stator, m_{rot} the mass of the rotor, and m_{stat} the mass of the stator.

Three additional types of losses occurring in a UM, i.e., brush losses P_{brush}, ventilation losses P_{vent}, and friction losses P_{frict}, depend mainly on the motor speed. When optimizing the geometry of the rotor and the stator, the motor speed is assumed fixed, hence P_{brush}, P_{vent}, and P_{frict} have no impact on the motor efficiency. Therefore, these losses are not significantly affected by the geometry of the rotor and the stator.

The output power P_2 of the motor is a product of the electromagnetic torque, T, and the angular velocity, ω:

$$P_2 = T\omega. \quad (3)$$

When considering all the mentioned losses and the output power, the overall efficiency of a UM can be defined as follows:

$$\eta = \frac{P_2}{P_2 + P_{\text{Cu}} + P_{\text{Fe}} + P_{\text{brush}} + P_{\text{vent}} + P_{\text{frict}}}. \quad (4)$$

2.2 Rotor and stator geometry optimization problem

The rotor and the stator of a UM are constructed by stacking the iron laminations. The shape of the laminations is specified by several parameters that define the rotor and stator geometry in two dimensions. There are invariable and variable parameters. Invariable parameters are fixed; they cannot be altered, either for technical reasons (e.g., the air gap) or because of the physical constraints on the motor (e.g., the radius of the rotor shaft). Variable parameters do not have predefined optimum values. Some variable parameters are mutually independent and without any constraints. Others are dependent, either on some invariable parameters or on mutually independent ones.

In our case, ten mutually independent variable parameters defining the rotor and stator geometry are subject to optimization (see Fig. 1):

- rotor yoke thickness (ryt),
- rotor external radius (rer),
- rotor pole width (rpw),
- stator width (sw),
- stator yoke horizontal thickness (syh),
- stator yoke vertical thickness (syv),
- stator middle part length (sml),
- stator internal edge radius (sie),
- stator teeth radius (str),
- stator slot radius (ssr).

The optimization task is to find the geometry parameter values that would generate the rotor and stator geometry with minimum power losses.

2.3 Numerical simulation of motor designs

Reliable numerical simulation is a prerequisite for automated design optimization. To evaluate settings of the rotor

and stator geometry parameters with respect to the resulting power losses, the ANSYS finite-element method simulation package [1] was used. This software provides a variety of structural, thermal, fluid and electromagnetic analysis capabilities suitable to simulate arbitrary products under real-world conditions.

For the purpose of this study, the simulation software was connected with the applied optimization algorithms. Communication within the simulator-optimizer loop was done through file data transfer. To evaluate a geometry parameter setting, an optimization algorithm generated a script file containing the parameter values and activated the simulation software. This software read the parameter values from the script file, performed the finite-element calculation of the motor electromagnetic states, assessed the power losses, and wrote this result into an output file. This result was then received by the algorithm that used it as a cost value for the evaluated solution.

3 Applied stochastic optimization methods

There exists a variety of stochastic optimization methods and selecting an appropriate one is part of a challenge in solving real-world design optimization problems. We tested six stochastic methods on optimizing the UM rotor and stator geometries to eventually draw conclusions on their suitability for this problem. The selected methods included well-known evolutionary methods widely used in optimization: traditional generational evolutionary algorithm, steady-state evolutionary algorithm and differential evolution; two particle-based techniques: particle swarm optimization and similar, but less frequently applied electromagnetism-like algorithm; and the recently proposed multilevel ant stigmergy algorithm.

All applied methods use real vector representation of candidate solutions where each vector component represents one geometry parameter of the UM rotor or stator. The parameter search space is put into discrete form and the stopping criterion is given by the number of solutions to be evaluated. The applied optimization methods are described in the following subsections.

3.1 Generational evolutionary algorithm

The generational evolutionary algorithm (GEA) is an optimization method that imitates the principles of Darwinian theory of evolution. By applying selection, crossover and mutation to a population of solutions, it creates better and better offspring populations (Fig. 2). This method was originally studied by Holland [8] and made popular by Goldberg [7].

Generational Evolutionary Algorithm

1. Evaluate the initial population S of random individuals.
2. While stopping criterion not met, do:
   2.1. Create an empty offspring population $\overline{S}$.
   2.2. Repeat until population $\overline{S}$ full:
      (a) Select two parents from S with tournament selection.
      (b) Create two offspring by crossing the parents.
      (c) Mutate and evaluate the offspring.
      (d) Add the offspring into the new population $\overline{S}$.
   2.3. Copy $\overline{S}$ into S.

Fig. 2 Outline of the generational evolutionary algorithm

GEA starts with a population of randomly created individuals encoded as real vectors. In the main loop (2.2 in Fig. 2), GEA creates new individuals from the parent population. First, tournament selection is used to select two parent individuals from the population. In tournament selection a few individuals compete to be chosen as parents for new individuals. The fittest individual wins the tournament. In case of a tie, the winner is chosen randomly. Tournament selection is applied twice and the two winning individuals, called parents, are then subject to crossover. Crossover consists of exchanging the genetic material (solution components) between the two parents. As a result, two offspring individuals are created. After that, the offspring are mutated (some vector components of each individual are randomly perturbated), evaluated and inserted into the offspring population. This loop is repeated until the new population is filled, i.e., until the number of created offspring reaches the predefined population size.

3.2 Steady-state evolutionary algorithm

The steady-state evolutionary algorithm (SSEA) is similar to GEA, with the exception of maintaining a single population of solutions. Like in GEA, at every step two offspring are created by applying the evolutionary operators. But instead of filling a new population, the offspring replace the worst two individuals in the current population (Fig. 3).

Steady-State Evolutionary Algorithm

1. Evaluate the initial population S of random individuals.
2. While stopping criterion not met, do:
   2.1 Select two parents from S with tournament selection.
   2.2 Create two offspring by crossing the parents.
   2.3 Mutate and evaluate the offspring.
   2.4 Replace the two worst individuals in S with the offspring.

Fig. 3 Outline of the steady-state evolutionary algorithm

Differential Evolution

1. Evaluate the initial population S of random individuals.
2. While stopping criterion not met, do:
   2.1. For each parent $\mathbf{s}_i$ $(i = 1, \ldots, |S|)$ from S repeat:
      (a) Randomly select three individuals $\mathbf{s}_{i_1}, \mathbf{s}_{i_2}, \mathbf{s}_{i_3}$ from S, where i, i_1, i_2 and i_3 are pairwise different.
      (b) Calculate candidate $\mathbf{c}$ as $\mathbf{c} = \mathbf{s}_{i_1} + w \cdot (\mathbf{s}_{i_2} - \mathbf{s}_{i_3})$, where w is a scaling factor.
      (c) Modify the candidate by binomial crossover with the parent $\mathbf{s}_i$ using crossover probability $crossProb$.
      (d) Evaluate the candidate.
      (e) If the candidate is better than the parent, replace the parent with the candidate.
   2.2. Randomly change the order of the individuals in S.

Fig. 4 Outline of differential evolution

3.3 Differential evolution

Differential evolution (DE) is a population-based algorithm for optimizing functions on totally ordered spaces. It was developed by Price and Storn [13] as a variant of evolutionary algorithms. The basic idea of DE is outlined in Fig. 4.

DE can adopt many different strategies that are by convention [14] denoted as *DE/x/y/z*. *DE* stands for Differential Evolution, *x* represents the method of selection of the first individual $\mathbf{s}_{i_1}$ that can be either random (*rand*) or the best so far (*best*), *y* is the number of difference vectors used and *z* defines the type of crossover, which can either be binomial (*bin*) or exponential (*exp*). In this paper the most widely used strategy *DE/rand/1/bin* has been applied.

In DE the notion of parent is somewhat different from the one used in GEA or SSEA. The candidate individual is generated as a weighted sum of three randomly chosen individuals that are different from the parent individual. Only then, the parent participates in the creation of the candidate—the candidate is modified by crossover with its parent. Finally, the candidate is evaluated and compared to the parent. The candidate replaces the parent in the population, only if it is better than the parent. The described procedure (Loop 2.1 in Fig. 4) is repeated consecutively for all parent individuals in the population. When it is finished, the order of the parents is randomly changed and the procedure is repeated.

3.4 Particle swarm optimization

Particle swarm optimization (PSO) is a stochastic optimization technique developed by Eberhart and Kennedy [6, 9] and inspired by social behavior of bird flocking or fish schooling. Suppose the following scenario: a group of birds are randomly searching for a single piece of food in an area. None of the birds knows where the food is, but they know how far from the food they are. The most effective strategy for finding the food is to follow the bird which is nearest to the food.

The PSO algorithm is derived from such scenarios and applied to optimization problems. In PSO, each solution is a "bird" in the search space. We call it a particle. Every particle $\mathbf{s}_i$ has a fitness value, which is evaluated by the fitness function to be optimized, and velocity $\mathbf{v}_i$, which directs the flying of the particle. The particles fly through the search space by following the current best particles.

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating populations (see Fig. 5). In every iteration, each particle is updated by following two "best" solutions. The first one, $\mathbf{s}_i^{p_{\text{best}}}$, is the best solution found by the particle $\mathbf{s}_i$ so far. The second "best" solution considered by the particle swarm optimizer is the best solution obtained so far by any particle in the population. This best is the global best, called $\mathbf{s}^{g_{\text{best}}}$. After finding the two best solutions, every particle updates its velocity $\mathbf{v}_i$, and position $\mathbf{s}_i$ according to the following equations:

$$\mathbf{v}_i = \omega \mathbf{v}_i + c_1\left(\mathbf{s}_i^{p_{\text{best}}} - \mathbf{s}_i\right) + c_2(\mathbf{s}^{g_{\text{best}}} - \mathbf{s}_i) \tag{5}$$

$$\mathbf{s}_i = \mathbf{s}_i + \mathbf{v}_i, \tag{6}$$

where $-\mathbf{v}_{\max} \leq \mathbf{v}_i \leq \mathbf{v}_{\max}$, c_1 and c_2 are learning factors, and ω is an inertia weight employed as an improvement proposed by Shi and Eberhart [19] to control the impact of the previous history of velocities on the current velocity. The weight ω plays the role of balancing between the local and global search and is updated with algorithm iterations.

Particle Swarm Optimization

1. Evaluate the initial population S of random solutions (particles).
2. While stopping criterion not met, do:
   2.1. For each particle $\mathbf{s}_i$ $(i = 1, \ldots, |S|)$ from S repeat:
      (a) Set $\mathbf{s}_i^{p_{\text{best}}}$ to be the best position of particle $\mathbf{s}_i$.
   2.2. Set $\mathbf{s}^{g_{\text{best}}}$ as the best particle found so far.
   2.3. For each particle $\mathbf{s}_i$ $(i = 1, \ldots, |S|)$ from S repeat:
      (a) Calculate particle velocity according to Eq. (5).
      (b) Update particle position according to Eq. (6).
   2.4. Update the value of the inertia weight ω.

Fig. 5 Outline of particle swarm optimization

3.5 Electromagnetism-like algorithm

The electromagnetism-like (EM) algorithm is similar to PSO. The main difference is in calculating moves in the search space. Like in PSO, each solution is moved to some new position at every step of EM algorithm. But instead of being influenced only by the personal best and global best solutions as in PSO, each solution is influenced by every other solution in the population.

The EM optimization heuristic was proposed by Birbil and Fang [2] for unconstrained global optimization of nonlinear functions. In a multidimensional search space where each point represents a candidate solution, a charge is associated with each point, calculated upon the objective function value of the solution. A population of solutions is created, in which each solution point exerts attraction or repulsion on other points, the magnitude of which is proportional to the product of the charges and inversely proportional to the distance between the points (Coulomb's Law). The overall move of a point depends on the influence of all other points of the population, which is expressed by a force vector. The principle behind the algorithm is that worse solutions prevent a move in their direction by repelling other solutions in the population, while better solutions facilitate moves in their direction.

The total force exerted on each point by all other points (Step 2.1 in Fig. 6) depends on the charge of the point under consideration as well as of the points exerting the force, and the Euclidean distance between them. The charge of each point $\mathbf{s}_i$ is determined by its objective function value $f(\mathbf{s}_i)$ in relation to the objection function value of the current best point $\mathbf{s}_{\text{best}}$ in the population, with better objective function values resulting in higher charges. For a minimization problem, the charge q_i of the point $\mathbf{s}_i$ is determined according to equation:

$$q_i = \exp\left(-d\,\frac{f(\mathbf{s}_i) - f(\mathbf{s}_{\text{best}})}{\sum_{k=1}^{|S|}(f(\mathbf{s}_k) - f(\mathbf{s}_{\text{best}}))}\right), \tag{7}$$

where $|S|$ represents the population size and d is the dimension of the search space. For each point $\mathbf{s}_i$, $i = 1, \ldots, |S|$, a force vector $\mathbf{F}_i$ is determined as follows:

$$\mathbf{F}_i = \begin{cases} \sum_{j=1, j\neq i}^{|S|} (\mathbf{s}_j - \mathbf{s}_i)\frac{q_i q_j}{\|\mathbf{s}_j - \mathbf{s}_i\|^2} & \text{if } f(\mathbf{s}_j) < f(\mathbf{s}_i) \\ \sum_{j=1, j\neq i}^{|S|} (\mathbf{s}_i - \mathbf{s}_j)\frac{q_i q_j}{\|\mathbf{s}_j - \mathbf{s}_i\|^2} & \text{if } f(\mathbf{s}_j) \geq f(\mathbf{s}_i) \end{cases}. \tag{8}$$

In this way, a point with a superior objective function value attracts the other points, while a point with an inferior objective value repels them. The forces exerted on $\mathbf{s}_i$ by the other points are combined by means of vector summation. The movement of the points according to the resulting forces is then performed (Step 2.2 in Fig. 6), which generates a new population. The imposed force is normalized by division with its norm and therefore only identifies the direction of the move, not the magnitude. The magnitude of each move is determined for each dimension separately according to the charges ratio of the involved solutions.

Electromagnetism-Like Algorithm

1. Evaluate the initial population S of random solutions.
2. While stopping criterion not met, do:
   - 2.1 Calculate the force of solutions $\mathbf{s}_j$ on solution $\mathbf{s}_i$, where $j \neq i$.
   - 2.2 Move each solution $\mathbf{s}_i$ according to the forces of other solutions.
   - 2.3 Evaluate each new solution.

Fig. 6 Outline of electromagnetism-like optimization algorithm

3.6 Multilevel ant stigmergy algorithm

Dorigo and coworkers originally proposed the ant colony optimization algorithm [3, 4] inspired by the social behavior of real ants in exploring their environment. To find an optimal path from their nest to a location of interest, such as food source, they deposit a chemical called pheromone and follow the pheromone trails of other ants. In the ant colony optimization algorithm, the artificial ants incrementally construct solutions by adding solution components to a partial solution under consideration. The multilevel ant stigmergy algorithm (MASA) [10] is a new approach to solving multiparameter optimization problems based on ant colony optimization and stigmergy, a type of collective work observed in ant colonies. MASA consists of four phases that are explained in the following paragraphs (see also Fig. 7).

Search graph construction. The problem parameters are transformed into a search graph where vertices represent discrete values of parameters. A vertex representing a parameter value is connected to all vertices representing the values of the next parameter. In this way, the multiparameter optimization problem is transformed into a problem of finding the cheapest path. The complexity of search graph construction is $O(nm)$ where nm is the total number of vertices in the graph that equals to the product of the number of problem parameters, n, and the average number of values per parameter, m.

Coarsening. The graph is coarsened to a predetermined size. Coarsening is done by merging two or more vertices that represent discretized values of the same parameter into one vertex; this is achieved in L iterations (we call them levels). In the coarsened graph the initial amount of pheromone is deployed in all vertices.

Fig. 7 Outline of the multilevel ant stigmergy algorithm

Multilevel Ant Stigmergy Algorithm

1. Construct the search graph from all parameters.
2. Coarsen the graph in L levels.
3. Initialize vertices with initial amount of pheromone.
4. For all levels ℓ from L down to 1, do:
   4.1. While current level ℓ stopping criterion not met, do:
      (a) For all ants find the cheapest path (using probability rule).
      (b) Update pheromone amounts in all vertices visited by the ants.
      (c) Additionally increase the pheromone amounts on currently best path (daemon action).
      (d) Evaporate pheromone in all vertices.
   4.2. Refine the graph by one level.

Optimization. Here the algorithm applies the optimization procedure based on ant colony optimization [5] (Loop 4.1 in Fig. 7). All ants simultaneously start from the starting vertex. The probability of choosing the next vertex depends on the amount of pheromone in the vertices. Ants repeat this action until they reach the ending vertex. The parameter values gathered on each ant's path represent a candidate solution which is then evaluated according to the given objective function. Afterwards, each ant returns to the starting vertex, on its way depositing pheromone in the vertices according to the evaluation result: the better the result, the more pheromone is deposited. If the gathered parameter values form an infeasible solution, the amount of pheromone in the parameter vertices is slightly decreased. When the ants return to the starting vertex, two additional actions are performed. First, like in ant colony optimization, a "daemon action" is applied as a kind of elitism, i.e., the pheromone amount on the currently best path is additionally increased. Second, the pheromone in all vertices evaporates, i.e., in each vertex the amount of pheromone is decreased by some predetermined percentage.

Refinement. The coarsened graph is refined by one level. All vertices created from one vertex have the same amount of pheromone as the original one. When refinement is done, the optimization phase continues. These two phases are repeated until the graph is expanded to its original size and the optimization performed on every level of the expansion.

4 Experimental evaluation

4.1 Experimental setup

As explained in Section 2.2, we optimize ten parameters of the UM rotor and stator geometry. Predefined search intervals for their values are used and the discretization step for all parameters is 0.1 mm. Table 1 summarizes the number of possible settings for each parameter. Therefore, the size of the search space can be obtained as a product of the numbers of possible settings over all parameters. It turns out to be approximately 1.92×10^{20} points.

Table 1 Number of possible settings for the optimized UM geometry parameters

Parameter	Number of settings
stator width	102
stator yoke horizontal thickness	170
stator yoke vertical thickness	270
stator middle part length	150
stator internal edge radius	140
stator teeth radius	37
stator slot radius	45
rotor yoke thickness	130
rotor external radius	200
rotor pole width	45

Five out of six applied optimization methods, GEA, SSEA, DE, PSO and EM, need an initial population of solutions to start the search. If the initial solutions are created as random points in the search space, many of them are infeasible, i.e., result in a nonfunctional geometry. To estimate the proportion of infeasible solutions in the search space, a simple experiment was carried out. 30,000 points in the search space were chosen randomly. Among them only seven were found feasible. To assist the population-based methods in finding feasible solutions, the initial populations did not consist of random solutions, but rather of solutions that were random perturbations of the original engineering solution.

Unlike the other applied methods that gradually improve the solutions, MASA is a solution-construction method. Initial experiments with MASA on the UM geometry optimization problem have shown this algorithm is capable of successfully navigating the search from infeasible to feasible regions. The multilevel approach significantly reduces the search space in the early stages of exploration. This reduction enables MASA to perform well without any background information on the feasibility of solutions. Through stigmergy, infeasible regions in the search space are found less attractive

by the ants, and consequently the search focuses on feasible ones.

The algorithmic settings were determined according to computational complexity limitations and experience from previously performed simulation-based optimization studies [11,16]. The stopping criterion for all optimization methods was specified by the number of solutions to be evaluated. It was set to 1400 and this value was chosen considering the computational complexity of the optimization procedure. The evaluation of a single solution through ANSYS simulation on an AMD 1.8 GHz computer took approximately two minutes and the execution of 1400 evaluations took about two days. Other algorithm parameters were set as follows. The evolutionary algorithms GEA, SSEA and DE all used population size 20 and crossover probability 0.9. GEA and SSEA used mutation probability 0.05 and tournament size 2. The scaling factor in DE was set to 0.5. PSO also used population size 20. The learning factors c_1 and c_2 were calculated for each move as $1 + r$, where r denotes a random number in the interval (0, 1). The inertia weight ω was calculated for each move as $0.1 + 0.9r$ and the maximum change of particle velocity $\mathbf{v}_{\max}$ was set according to the bounds of each optimized value. EM used population size 20 and magnitude factor of moves 1.3. The settings for MASA were determined regarding the optimization parameter with the highest number of possible values. This was the stator yoke vertical thickness with 270 possible settings (see Table 1). Accordingly, MASA operated with seven levels. At each level, 200 evaluations were performed by ten ants climbing down the graph 20 times.

4.2 Results of the applied methods

The optimization methods were run 20 times. The obtained results in terms of the UM power losses are presented statistically in Table 2. The methods' performance diagrams are split across three figures for better clarity. Figure 8 compares the performance of the three evolutionary algorithms, namely GEA, SSEA and DE, with the performance of the original engineering solution that amounted to 177.9 W, while Fig. 9 presents the diagrams of the other three methods, PSO, EM and MASA, together with the original engineering solution and DE as the best evolutionary method. The performance of MASA is shown entirely in Fig. 10.

The results first of all show that all the applied methods significantly improve the original engineering design of the UM rotor and stator. Specifically, the geometry parameter settings with minimum power losses were found by MASA. The second best method was DE, followed by GEA, SSEA and PSO that performed comparably well, and EM that performed a little worse. While PSO and EM have a very fast initial convergence, they are later unable to find solutions as good as those of the other methods. Figures 8 and 9 show

Table 2 Result statistics for the optimization methods (UM power losses in watts)

Method	Best	Average	Worst	St. dev.
EM	134.9	141.9	148.0	3.7
PSO	132.1	139.2	145.5	4.6
SSEA	131.4	137.7	148.8	5.4
GEA	131.3	136.7	147.4	4.5
DE	129.1	132.9	139.9	3.3
MASA	114.2	128.9	135.9	7.8

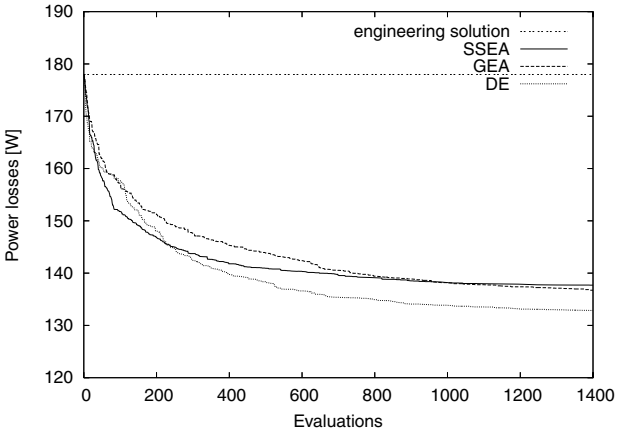

Fig. 8 Performance of the evolutionary algorithms in optimizing UM rotor and stator geometry parameters: averages over 20 runs

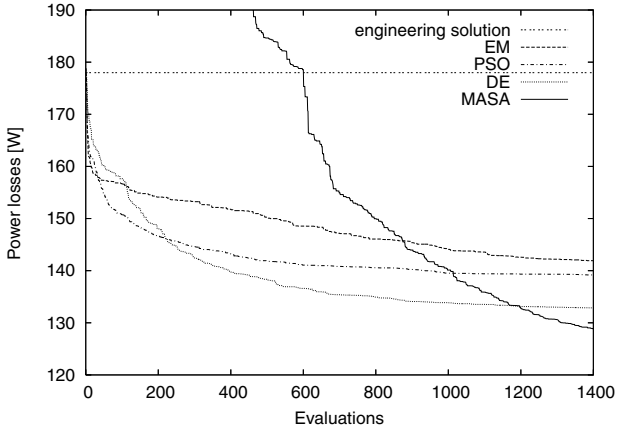

Fig. 9 Performance of DE, PSO, EM and MASA in optimizing UM rotor and stator geometry parameters: averages over 20 runs

the difference in the course of optimization of the six applied methods. While DE, GEA, SSEA, PSO and EM all start with the engineering solution originally used in motor production and evolve rather slowly, MASA starts with randomly created solutions that result in high power losses, but rapidly improve during the course of run (see Fig. 10). The averaged performance trace of MASA is not monotonic, as only feasible solutions are taken into account in calculating the average. Infeasible solutions get a predefined high value of losses during the evaluation procedure and are excluded when

Table 3 Improvement of the results (UM power losses in watts) with local optimization (LO)

Method	Best before LO	Best after LO	Average before LO	Average after LO	Average improvement	LO steps
EM	134.9	133.4	141.9	139.3	2.6	153
PSO	132.1	131.9	139.2	139.0	0.2	38
SSEA	131.4	130.5	137.7	137.4	0.4	49
GEA	131.3	129.5	136.7	135.7	1.1	76
DE	129.1	129.1	132.9	132.5	0.4	50
MASA	114.2	111.1	128.9	126.2	2.7	116

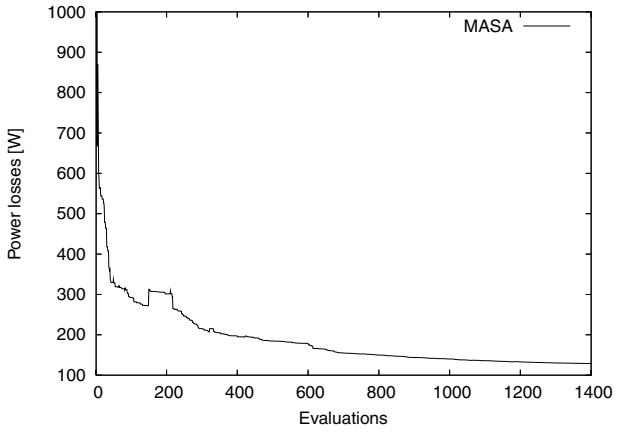

Fig. 10 Performance of MASA averaged over 20 runs

calculating the average. Furthermore, when MASA finds the first feasible solution, this solution typically results in high power losses and can increase the average value found. This phenomenon disappears when all 20 runs reach feasible solutions. In the experiments shown, this happened after 421 evaluations.

In six out of 20 runs, MASA was able to find geometry parameter values resulting in power losses under 120 W. In the rest 14 runs it performed comparably to DE.

4.3 Local optimization of solutions

To check for possible further improvement, the best solution from each run of every applied optimization method was subject to local optimization. The steepest-descent local optimization was applied and the procedure was not limited with a predefined number of steps. It rather stopped when a local minimum was encountered. The statistics of the improvements are given in Table 3.

From these results one can see that the solutions found by the methods SSEA, GEA, DE and PSO were closer to local minima than the ones found by MASA and EM, i.e., fewer steps were needed to reach a local minimum from the solutions obtained by evolutionary methods and PSO than MASA and EM. Accordingly, the former could be improved with local optimization to a lower extent than the latter. Moreover, even together with local optimization, the evolutionary methods could not find the solutions with power losses around 115 W. They seem to be stuck in local minima with values around 130 W. This was also noted for those solutions produced by MASA that had power losses values near 130 W before local optimization.

4.4 Resulting rotor and stator geometries

By applying selected stochastic optimization methods and locally improving the obtained solutions, we found various UM rotor and stator geometry parameter settings that minimize the power losses. The optimization procedures were however driven according to the results of computer simulation. To provide a more realistic evaluation, we submitted the resulting designs to an expert designer to analyze them from the technical and production points of view. Here we compare the original engineering design created by domain experts and used as a starting point for the population-based methods with two optimized designs.

The original engineering rotor and stator design results in power losses of 177.9 W and can be viewed in Fig. 11. The figure shows the magnetic flux density in the laminations (darker areas denote higher magnetic flux density that causes higher power losses).

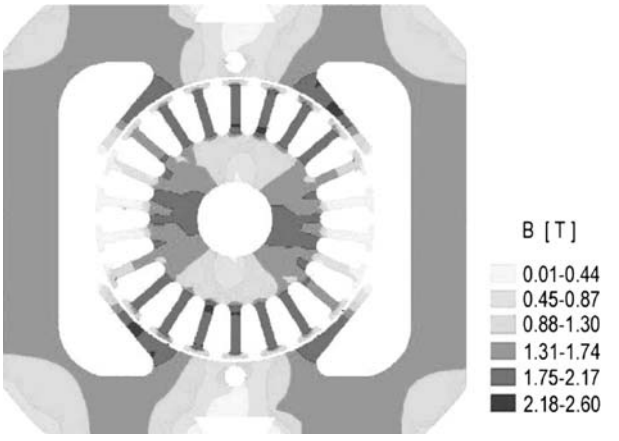

Fig. 11 Laminations of the original engineering rotor and stator design with power losses of 177.9 W

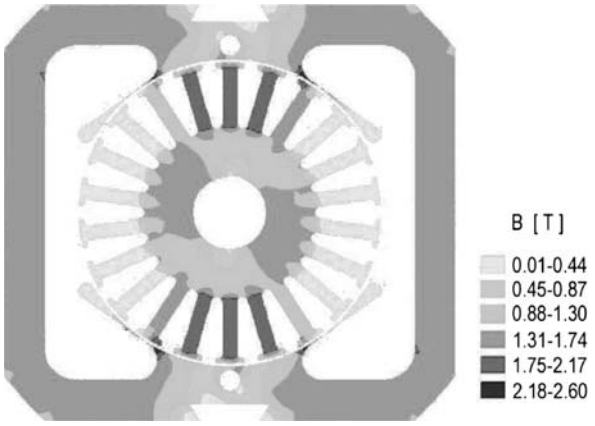

Fig. 12 Laminations of the rotor and stator design with minimum power losses (111.1 W) as found in the optimization experiments

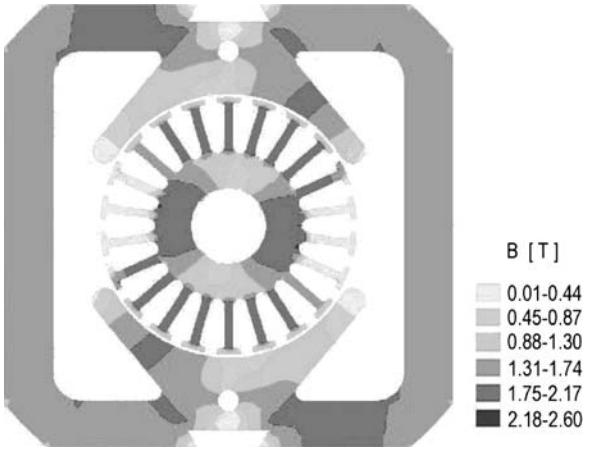

Fig. 13 Laminations of a locally optimal rotor and stator design acceptable from the production point of view (power losses 129.1 W)

The best rotor and stator geometry obtained in numerical optimization experiments generates power losses of 111.1 W. It was found by MASA with local optimization, and is presented in Fig. 12. These laminations have large rotor and stator slots, and therefore low copper losses and overall power losses. The difficulty with this design is, however, in the strange dimensions of the stator pole. Its narrow middle part makes it unacceptable for production. This outcome is due to the settings in the simulation script. It could be modified by inserting additional constraints on the stator geometry. Once they are specified, a new optimization cycle will be necessary. At this stage, however, the goal of our study is to check for possible improvements of the engineering UM design and compare the optimization methods on this problem.

We finally present a typical example of a feasible rotor and stator geometry (Fig. 13). It was found by both DE and MASA and its power losses are 129.1 W. This solution has very low iron losses in the rotor due to its small size and despite its high magnetic saturation (dark area). The small rotor and its saturation are compensated by large stator poles that ensure large enough magnetic flux. This design is feasible from the technical and production points of view.

5 Conclusion

We have performed a comparative study of six stochastic optimization methods in computer-assisted design of UM rotor and stator geometry. The primary design goal was to minimize the power losses, however, the resulting designs were also evaluated by an expert designer from the point of view of feasibility for use in regular production. The applied methods were generational evolutionary algorithm, steady-state evolutionary algorithm, differential evolution, particle swarm optimization, electromagnetism-like algorithm and multilevel ant stigmergy algorithm. They were employed in optimizing the geometry parameters of an UM already in regular production. The optimization procedures were coupled with a numerical simulator that evaluated the candidate solutions.

The output of this study can be summarized in several important findings. Above all, all tested optimization methods were able to significantly improve the original engineering design. Among the tested methods, the recently proposed optimization technique MASA generated designs with minimum power losses. Its additional advantage shown on this problem was the capability of successfully performing the optimization from random starting points, which was not the case with other methods. In our opinion, the superiority of MASA arises from its multilevel search feature, which appears to be particularly suitable for coping with search spaces characterized by small proportion of feasible solutions. While this advantageous property has been empirically confirmed on a specific problem in this study, further research will be needed to investigate if it generalizes to a class of optimization problems.

The analysis of the optimized UM rotor and stator geometries by an expert designer revealed some deficiencies of the designs with very low power losses. Although perfect with respect to the primary design goal, they turned out to be infeasible for regular production because of the limitations in the manufacturing process. This result calls for the refinement of the optimization script with additional constraint and new optimization cycle. This is planned as part of the future work. On the methodological side, the multilevel approach found beneficial with MASA, is worth of exploration in combination with other optimization methods.

Acknowledgment The work presented in this paper was supported by the Slovenian Research Agency. The authors are grateful to dr. Boris Benedičič from Domel electric motor company for providing the optimization scripts and evaluating the optimized motor designs.

References

1. ANSYS User's Manual (2000) ANSYS version 5.6. ANSYS Inc. Canonsburg, PA
2. Birbil SI, Fang SC (2003) An electromagnetism-like mechanism for global optimization. J Global Optim 25(3):263–282
3. Dorigo M (1992) Optimization, learning and natural algorithms. Ph.D. Thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy (in Italian)
4. Dorigo M, Maniezzo V, Colorni A (1996) The ant system: Optimization by a colony of cooperating agents. IEEE Trans Syst, Man, and Cybern–Part B 26(1):1–13
5. Dorigo M, Di Caro G, Gambardella LM (1999) Ant algorithms for discrete optimization. Artif Life 5(2):137–172
6. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the 6th international symposium micro machine and human science, Nagoya, Japan, pp 39–43
7. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading, MA
8. Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, MI
9. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference neural networks. Perth, Australia, pp 1942–1948
10. Korošec P, Šilc J (2005) The multilevel ant stigmergy algorithm: An industrial case study. In: Proceedings of the 7th international conference on computational intelligence and natural computing. Salt Lake City, UT, pp 475–478
11. Korošec P, Papa G, Šilc J (2005) Optimization algorithms inspired by electromagnetism and stigmergy in electro-technical engineering. WSEAS Trans Inf Sci Appl 5(2):587–591
12. Papa G, Koroušić-Seljak B, Benedičič B, Kmecl T (2003) Universal motor efficiency improvement using evolutionary optimization. IEEE Trans Ind Electron 50(3):602–611
13. Price KV, Storn R (1997) Differential evolution—a simple evolution strategy for fast optimization. Dr. Dobb's J 22(4):18–24
14. Price KV, Storn R, Lampinen JA (2005) Differential evolution: a practical approach to global optimization. Springer-Verlag, Secaucus, NJ
15. Puternicki P, Rudnicki M (1999) Optimal design methodologies with application to small commutator motors. In: Proceedings of the international symposium on electromagnetic fields in electrical engineering, Pavia, Italy, pp 397–400
16. Robič T, Filipič B (2004) In search for an efficient parameter tuning method for steel casting. In: Proceedings of the international conference on bioinspired optimization methods and their applications. Ljubljana, Slovenia, pp 83–94
17. Sen PC (1996) Principles of electric machines and power electronics. John Wiley & Sons, New York
18. Shaked NT (2004) Optimization of switched reluctance motors using genetic algorithms. M.Sc. Thesis, Ben-Gurion University of the Negev, Beer Sheva, Israel
19. Shi Y, Eberhart RC (1998) A modified particle swarm optimizer. In: Proceedings of the IEEE international conference evolutionary computation. Anchorage, AK, pp 69–73
20. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82