A Modified Shuffled Frog Leaping Algorithm for the Topology Optimization of Electromagnet Devices

Wenjia Yang *, Siu Lau Ho and Weinong Fu *

Department of Electrical Engineering, The Hong Kong Polytechnic University; Hong Kong SAR, China; siu-lau.ho@polyu.edu.hk

* Correspondence: wj.yang@connect.polyu.hk (W.Y.); weinong.fu@polyu.edu.hk (W.F.);
Tel.: +852-2766-6158 (W.F.); Fax: +852-2330-1544 (W.F.)

Abstract: The memetic algorithms which employ population information spreading mechanism have shown great potentials in solving complex three-dimensional black-box problems. In this paper, a newly developed memetic meta-heuristic optimization method, known as shuffled frog leaping algorithm (SFLA), is modified and applied to topology optimization of electromagnetic problems. Compared to the conventional SFLA, the proposed algorithm has an extra local search step, which allows it to escape from the local optimum, and hence avoid the problem of premature convergence to continue its search for more accurate results. To validate the performance of the proposed method, it was applied to solving the topology optimization of an interior permanent magnet motor. Two other EAs, namely the conventional SFLA and local-search genetic algorithm, were applied to study the same problem and their performances were compared with that of the proposed algorithm. The results indicate that the proposed algorithm has the best trade-off between the results of objective values and optimization time, and hence is more efficient in topology optimization of electromagnetic devices.

Keywords: topology optimization; numerical method; evolutionary algorithm; shuffled frog leaping algorithm

1. Introduction

Topology optimization (TO) is the process of determining the material distribution for devices that could yield the best performance under given constraints. TO has shown great potential in the device design and optimization process and has become a new paradigm to provide a quantitative design method for modern industries in multiple disciplines [1]. To enhance the speed of the TO process, numerous modern mathematical algorithms have been proposed and developed, and proved to be efficient in solving complicated three-dimensional (3D) problems [2]. Based on the optimization searching methodologies, these methods can be categorized as the deterministic method, where the searching process is based on the gradient information, such as the Newton method; and the meta-heuristic method, where the searching process is guided by the information gathered from random or designed population, such as the evolutionary algorithms, or those from the related disciplines such as the bidirectional evolution structural optimization algorithm (BESO) [3,4].

Evolutionary algorithms (EA) imitate the evolution process of population existing in the natural environment. The population could be species, flock of birds or fishes, or any other possible group of candidates that could share experience and intelligence. EAs have been adopted widely in optimizing the shape and size of electromagnetic devices because of their simplicity in applications and high optimization efficiency [5]. For instance, genetic algorithms (GA) are applied together with the finite element method (FEM) in the design and parameter optimization of interior permanent-magnet motors in various studies [6–8], and they have proved to be efficient in solving these problems. However, in the

case of TO, due to the large degree of freedoms in the design space, the inherit complexity of the problem is much higher than those of the shape and size optimization problems, and hence the searching speed of the algorithm plays an important role affecting the optimization time. One issue with respect to the conventional EAs is the passage of genes is limited to the offsprings, because the generation of children is based on the crossover of elite parents only, and hence the searching direction is limited, resulting in a relatively poor searching efficiency. To overcome this issue, local search steps are often added to the elite population or to the best candidate [9]. In the local search step, the neighborhood of the feasible candidate is searched by adding randomly generated vectors to the candidate and with their performances investigated accordingly. Another type of meta-heuristic algorithms is the memetic algorithm, where memetic evolution is used as the searching mechanism. The entire population is divided into subgroups called memeplex, and, within each memeplex, the searching direction, which can also be regarded as the memes in the memeplex, is determined by the positions of the best solutions. Each individual in the memeplex learns from both the local best solution in the memeplex and from the global best solution in the entire population. Because the memes in the memetic algorithm can spread freely through the population, this type of algorithms is associated with a stronger searching efficiency with fewer objective function evaluations when solving problems with higher degree of freedom and larger number of control variables [10–12].

The Shuffled Frog-Leaping Algorithm (SFLA), formulated in 2006, has been successfully applied to scheduling problems, network design problems, and parameter estimation of electromagnetic devices [13–15], and it has shown great potential in handling high dimension problems. The SFLA can be viewed as an extension of the classical particle-swarm optimization (PSO) algorithm, with certain newly added features such as the usage of memeplex. In this paper, the SFLA together with the Gaussian radial basis function (RBF) modeling method is employed to optimize the two-dimensional (2D) design of the rotor in a conventional interior permanent-magnet motor (IPM). To further improve the efficiency of the SFLA and avoid the problem of finding false optima, a local search step is added to the algorithm to add randomness to the optimization process. The design space of the optimization problem is discretized into small elements that will be filled with either iron or air, and Gaussian RBFs are used to calculate the topology function of each element to create the topology. The finite element method (FEM) is employed to evaluate the objective function of the TO problem. To showcase the prominent efficiency of the SFLA in the electromagnetic TO problems, three algorithms were employed for the same IPM TO problem: the local-search GA (LS-GA), conventional SFLA, and the proposed SFLA. Both the speed and accuracy of the algorithms were compared, and the results indicate that the proposed SFLA is more appropriate in the design and optimization of complicated electromagnetic devices such as IPMs.

2. The Modified Shuffled SFLA

The SFLA was proposed by Eusuff and Lansey in 2006 [13]. Similar to the other algorithms in the memetic family, the evolution process in SFLA also contains both local exploitation mechanism and global exploration mechanism. An initial population is randomly generated, evaluated, and sorted. Based on the objective function values, the individuals in the initial population are partitioned into different subgroups of equal size named memeplexes. Within each memeplex, individuals will evolve via moving towards either the local best or the global best in the memeplex. After certain iterations of local evolvement, the entire population will be re-shuffled for the spreading of the memes. These steps will be repeated until the stopping criteria are met [16]. In the ordinary SFLA algorithm, in each iteration, the best candidate, i.e., frog, of each memeplex will not update its position. As each candidate will always remain at the same position, especially for the best frog in the entire population, it is easy for the algorithm to be trapped in some local optima found at the early stage of the optimization process and hence ending prematurely. To avoid this problem, a local search process for the best frog in each memeplex is added to the algorithm to help it escape from false best solutions.

The flowchart of the proposed SFLA is given in Figure 1. For an optimization problem with n control parameters, each frog in the algorithm is an n dimension vector, $X_i = [x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,n}]$. At the beginning of the optimization, N frogs are generated using the Latin-hypercube sampling method to ensure the entire design space can be well explored. Based on the objective value of each individual frog, the initial population is then sorted, and evenly partitioned into m memeplexes $S_1, S_2, \ldots, S_m$, with each memeplex containing p frogs, where $N = m \times p$. For instance, if $m = p = 3$, then S_1 contains X_1, X_4, X_7, S_2 contains X_2, X_5, X_8, and S_3 contains X_3, X_6, X_9, given all X_i have been sorted. The number of frogs in each memeplex is set to be $n + 1$ based on the Nelder–Mead method [17]. Following the partition process, a local search is conducted in memeplexes. A random vector with 1/10 upper and lower bound is added firstly to the best frog in the memeplex, $X_{b,k}$:

$$X'_{b,k} = X_{b,k} + \frac{1}{10}[\text{rand}\,(lb, ub), \ldots, \text{rand}(lb, ub)] \tag{1}$$
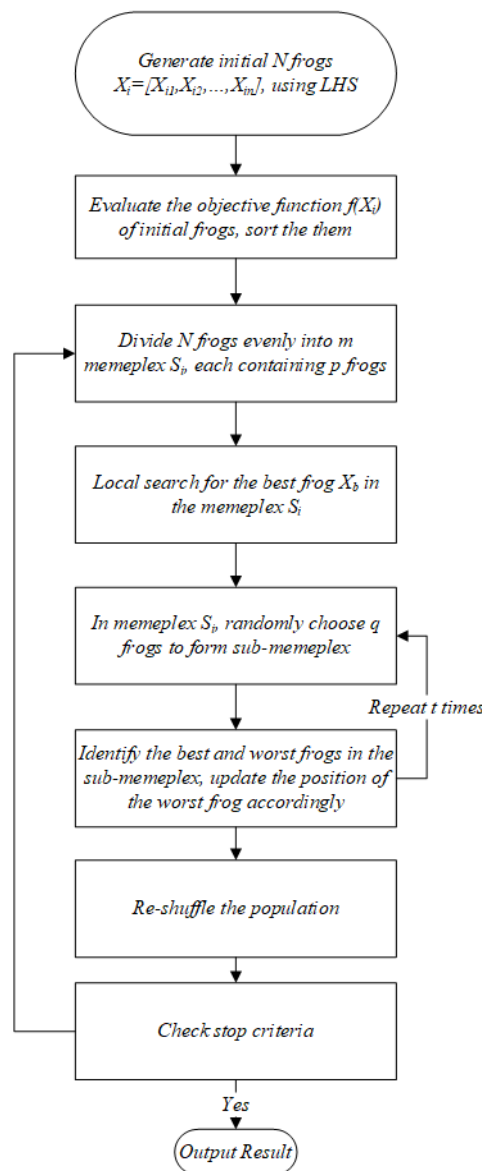

Figure 1. Flowchart of the shuffled frog leaping algorithm (SFLA).

The objective function value of the new frog is evaluated. If it is better than the original frog, then $X_{b,k}$ is replaced by $X'_{b,k}$. After the update of the best frog in the kth memeplex S_k, a small portion of frogs are selected to form a sub-memeplex. The selection probability of each frog in the memeplex is derived based on their objective value ranking in the memeplex using the triangular probability function [11]:

$$P_{i,k} = 2(p+1-i)/p(p+1), i = 1, 2, 3, \ldots, p \tag{2}$$

where $P_{i,k}$ is the frog selection probability in the kth memeplex and i is the ranking of the frog. Inside the sub-memeplex, the frog with the best objective value, $X_{b,k}$, and the frog with the worst objective value, $X_{w,k}$, are identified. To imitate the process of sharing information, the worst frog will learn from both the best frog in the sub-memeplex and from the global best in the entire population, which is a process similar to the PSO algorithm [13]. Mathematically, the position of the worst frog is updated by:

$$v = S \times [w \times (X_{b,k} - X_{w,k}) + (1-w) \times (X_{b,\text{overall}} - X_{w,k})]$$
$$X'_{w,k} = X_{w,k} + v \tag{3}$$

where S is the predefined step size, and w is the random weight between [0,1]. If the new position $X'_{w,k}$ is within the design space, its objective value is evaluated. If the result is not better than the original objective value of the same frog, the position of the worst frog will be updated again using the position of the global best frog, $X_{b,\text{overall}}$, instead of the position of the best frog in the sub-memeplex:

$$v = S \times w \times (X_{b,\text{overall}} - X_{w,k})$$
$$X'_{w,k} = X_{w,k} + v \tag{4}$$

If the new position $X'_{w,k}$ is within the design space, again its objective value is evaluated. If in both cases there is no improvement in the objective value, the worst frog is mutated according to:

$$X'_{w,k} = [\text{rand}\,(lb, ub), \ldots, \text{rand}(lb, ub)] \tag{5}$$

The local search process will repeat for t times, and then the entire population is shuffled and moved to the next iteration. As extra numbers of function evaluations are required, the optimization time for each iteration is longer than that in conventional SFLA. These steps repeat until the stop criterion is met.

3. Normalized Gaussian Radial Basis Function Formulation

In the TO process for the electromagnetic devices, a commonly practiced method when constructing the topology of the model is to divide the design space into a number of small elements, and assign material to each element based on the value of its topology function controlled by the design variables of the problem. Because the position vectors of the frogs can be any arbitrary numbers in the design space, the design variables in SFLA should be continuous. Therefore, in this paper, the normalized RBF modeling method [16] is employed to reconstruct the topology of the electromagnetic devices. In the normalized RBF method, the value of topology function at location x in the design space can be described as:

$$s(x) = \frac{\sum\limits_{i=1}^{M} w_i \times \phi(\|x - x_i\|)}{\sum\limits_{i=1}^{M} \phi(\|x - x_i\|)}, w_i \in [0, 1] \tag{6}$$

where M is the total number of RBFs in the model, $\|\cdot\|$ is the multidimensional norm, w_i is the weight of each RBFs and is the design parameter, and the function ϕ is the predefined radial function. The value

of $s(x)$ determines the material type of the element x_e located at the position x, which is either iron or air in this paper:

$$s(x) > 0.5,\ Mat(x_e) = Iron$$
$$s(x) < 0.5,\ Mat(x_e) = Air$$

(7)

where $Mat(x_e)$ defines the material type of element x_e. Because only the weight of each RBF is controlled during the optimization process, the number of control parameters as well as the complexity of the problem can be significantly reduced. Moreover, the usage of RBFs helps to transfer the discrete topology matrix into function of continuous variables. The radial function used in this paper is the normalized Gaussian function with zero mean. For a RBF centered at x_e, its value at any point x in the design space is:

$$\phi(x) = \frac{1}{\sigma 2\pi} e^{-\frac{1}{2}\left(\frac{\|x - x_e\|}{\sigma}\right)^2}$$

(8)

The radius of the RBFS is determined by its variance σ. To ensure that the entire design domain can be reconstructed, the RBFs are set to be evenly distributed, and their radii are larger than the distance between any two consecutive RBFs. In the proposed TO process, both the position and the radius of RBFs are predefined to reduce the number of control variables with a less complicated reproduction process. The design parameter, which is the weight of each RBF, is set to be positive to avoid any possible conflict in the topology definition, and is bounded between [0,1] to reduce the design space. After the calculation of the topology function, the material type of the element is also determined and assigned in order to create a topology matrix accordingly. To further remove the checkerboard pattern in the topology, a 2D filter is used to refine the topology matrix. The actual FEM evaluation is then conducted using the created model.

4. Problem Formulation and Numerical Experiments

In this paper, the proposed SFLA is applied to the topology optimization of the iron part in the rotor of a benchmark IPM. To reduce the time for objective value evaluation, the 2D model of the 1/4 cross-section of the prototype machine with periodic boundary condition is used for FEM evaluation, and the stack length is set to be 120 mm. All other parameters in the model are set to be constants. Table 1 lists all the fixed parameters.

Table 1. IPM fixed parameter values.

Parameters	Value	Unit
Stator outer radius	110	mm
Stator inner radius	75	mm
Air gap length	0.45	mm
Rotor outer radius	74.55	mm
Rotor inner radius	52.5	mm
Permanent magnet length	20.4	mm
Permanent magnet width	5.5	mm
Phase current amplitude	9.69	A
Coil number of turns	30	
Motor speed	2800	rpm
Number of slots/poles	48/8	

The design space of the problem is the 1/16th portion of the rotor which is then replicated with two mirror processes to re-construct the quarter model. The usage of smaller design space helps to improve the resolution of the obtained topology with the same number of elements, as well as ensuring a symmetrical rotor structure. The entire design domain is divided into 990 elements, 22 in a row and 45 in a column, with each element occupying 0.5° in width and 1 mm in length and is filled with either iron or air. Numerical experiments were conducted to search for the best number of RBFs and Gaussian variance for the RBF modeling process. To obtain a smooth topology with relatively small

number of design variables, $(4 \times 9) = 36$ RBFs are used to cover the entire design domain. Each RBF is set to be fiveelements from its neighbor RBF, in both radial direction and along the arcs, and the variance of the Gaussian function is set to 1. The control parameters, which are the weight of each RBF, are set to be bounded within [0, 1].

Figure 2 depicts the initial design domain and the 36 RBFs are used to construct the topology. The red circles in the design domain indicate the RBFs and their effective area, where inside the effective area the values of RBFs are positive, and outside the effective area their values are assumed to be zero. Hence, each RBF controls only the topology of a small region in the design domain, and a large variety of topology can be generated using the RBF method. The aim of the topology optimization is to maximize the average torque and minimize the torque ripple, and hence the objective function of the TO problem is defined as the weighted sum of the normalized average torque and the torque ripple of the IPM evaluated using Ansys Maxwell, evaluated by the virtual work method employed in the software:

$$\max f(X) = c_1 \frac{T_{\text{avg}}(X)}{T_{prototype}} - c_2 \frac{T_{\max}(X) - T_{\min}(X)}{T_{\text{avg}}(X)}, X = [X_1, X_2, \dots X_{36}] \tag{9}$$
$$\text{s.t. } Area_{iron} \leq Area_{\text{design}}$$

where c_1 and c_2 are two predefined weight coefficients and equal to 1 in this paper, $Area_{iron}$ is the area of iron part in the IPM of the design, and $Area_{\text{design}}$ is the area of the design space, which has a fixed size in the optimization problem, and hence can serve as the volume constraint of the TO problem.
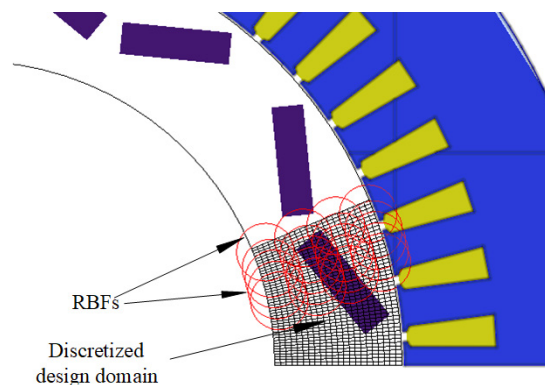

Figure 2. The initial discretized design domain and RBFs used to model it (in red).

Moreover, a transient analysis in Ansys Maxwell is used to compute the average torque and the torque ripple. Consequently, the length of the time step has a significant impact on the computational results. To address this issue, one first conducts a numerical experiment to decide the time step before the optimization, i.e., for a fixed topology, one incrementally reduces the length of time step until the errors of the computed average torque and the torque ripple between two successive time step reductions are smaller than a given tolerance. From this numerical experiment, the time step is finally set as period/100 for this case study.

The prototype IPM design and its performance are given in Figure 3. The average torque of the prototype, $T_{prototype}$, is 117.6 Nm, and the torque ripple of the machine is 9%. The algorithm stops when the best solution stays unchanged for five consecutive iterations or the iteration number exceeds 1000. To determine the appropriate algorithm parameters including the sub-memeplex size, local search times for the worst frogs and for the best frogs in the memeplex, the SFLA is run twice, using two sets of parameters, for each single objective optimization of the prototype IPM when optimizing its average torque. The parameter values and the corresponding algorithm performances are listed in Table 2. For both cases, the sizes of the population, memeplex, and sub-memeplex are identical. In the SFLA algorithm, the best frog local search step and the worst frog improvement step are the two core steps responsible for the exploration of the design domain. In the first run, the algorithm concentrates more on the directed search, and in each memeplex, the update of worst frog's position towards the best

frog's position is repeated thrice, while the random local search for the best frog is only conducted once. In the second run, the algorithm focuses on the random search, and hence the random local search for the best frog in each memeplex is conducted thrice, and the worst frog improvement step is only done once. The results of two runs indicate that, with comparable numbers of FEM evaluations, the second run can obtain a better optimization result. This is because in the first run more computing resources are used to improve the overall objective values of the entire population, which is not necessary in the case of TO of electromagnetic devices. In the second run, the addition of randomness to the population via the best frog in each memeplex significantly improves the algorithm's ability to escape from the local optimum, and hence the optimized result is better than that of the first run. Hence, in the actual multi-objective TO problem for the IPM, the second set of parameters is chosen.

(a)　　　　　　　　　　　　　　　　　　(b)
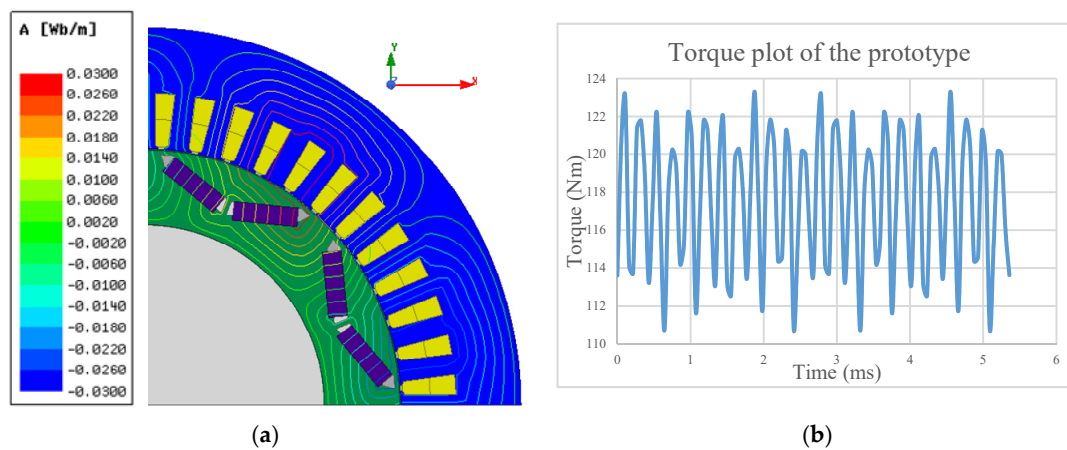
Figure 3. The conventional design of IPM used for comparison: (**a**) cross-section; and (**b**) torque plot.

Table 2. Algorithm parameters used and the results obtained for two test runs of SFLA.

|	First Run of SFLA	Second Run of SFLA
Population size	185	185
Memeplex size	5	5
Sub-memeplex size	3	3
Local search step size	1	2
Worst frog improvement repetition	3 times	1 time
Best frog improvement repetition	1 time	3 times
Number of FEM calls	636	901
Optimized results	106.85 Nm	134.73 Nm
No. of cycles to find the Optimum	1	6

After determining the algorithm parameters, the proposed modified SFLA is applied to the multi-objective TO problem formulated in the afore-section. To showcase the merit of the proposed method, its performance is compared with the performance of other two EAs, namely the original SFLA and LS-GA. Based on the famous no free lunch theorem, one cannot improve the performance of an optimization algorithm without sacrificing its performance in other aspects [18], and hence both the optimization time and the optimized results are compared as an indication of algorithm efficiency. The optimized topology designs and their corresponding torque plots of the three algorithms are given in Figures 4–6. The details of their performances are summed up in Table 3. All three algorithms are repeated three times and their performances are averaged to make the results more reliable.

**Table 3.** Summary of optimization performances of all three EAs.

	Conventional SFLA	LS-GA	Proposed SFLA
Average number of FEM evaluations	661 times	1800 times	905 times
Standard deviation in FEM evaluations	60	246	141
Optimized average torque	100.8 Nm	130.8 Nm	126.8 Nm
Optimized torque ripple	28.2%	20.0%	19.8%
Weighted sum objective	0.539	0.91	0.88
No. of cycles to find the Optimum	2	7	6
Ratio of the volume of the material to that of the entire design domain	67.7%	60.0%	63.5%

Among all three algorithms, the conventional SFLA has the fastest optimization speed, and the overall optimization requires only 600+ FEM evaluations to locate the optimum. However, because there is a lack of escape mechanism, the optimized result found by the SFLA has a relative low objective value, with an average torque of 100.8 Nm, torque ripple of 28.2%, and weighted objective value of 0.539. The population size is 185, and the number of cycles to find the optimum is two and does not go through any evolvement for the remaining optimization process, making the result of the algorithm highly dependent on the initial guess of the optimization process. On the other hand, the LS-GA algorithm has the longest optimization time, as well as the best optimization results. It can successfully locate the solution with the largest average torque value of 130.8 Nm, as well as the best weighted objective values of 0.91. In view of the material consumptions, the ratios of the volumes of the material to that of the entire design domain under the optimized topologies of the conventional SFLA, the LS-GA, and the proposed SFLA are, respectively, 67.7%, 60%, and 63.5%. However, the average number of FEM evaluations for LS-GA is about 1800, which is three times the optimization time of the conventional SFLA algorithm. The performance of the proposed algorithm falls in between the conventional SFLA algorithm and the LS-GA algorithm. The average time cost of the proposed algorithm is 950 FEM evaluations, which is 1.5 times of the conventional SFLA and almost half of the LS-GA, and the optimized result found by the proposed method is 0.88, with an average torque of 126.8 Nm and torque ripple of 19.8%. The tradeoff between the optimization speed and the result performance of the proposed algorithm is the most efficient among all three EAs. Comparing with the LS-GA algorithm, the proposed algorithm sacrifices only a small degradation in the quality of the optimization result (96.7% objective value) in return for a shorter optimization time (50% optimization time). On the other hand, the local search step in the proposed algorithm enhances the algorithm's ability for exploration, and the number of cycles to find the optimum is found is 6, which is the last iteration. Because the design space is more extensively explored, the optimization time is relatively longer (1.5 times of the conventional SFLA), but with a much better optimization results (1.27 times the objective value of the conventional SFLA).
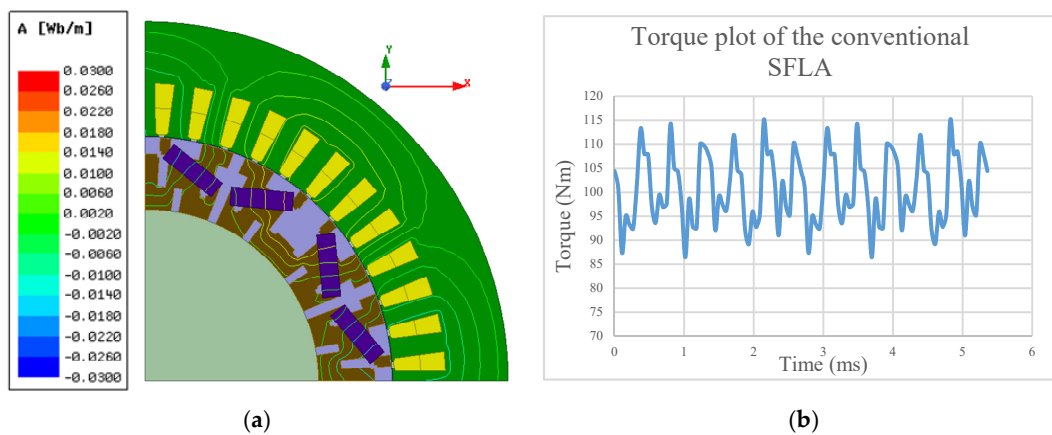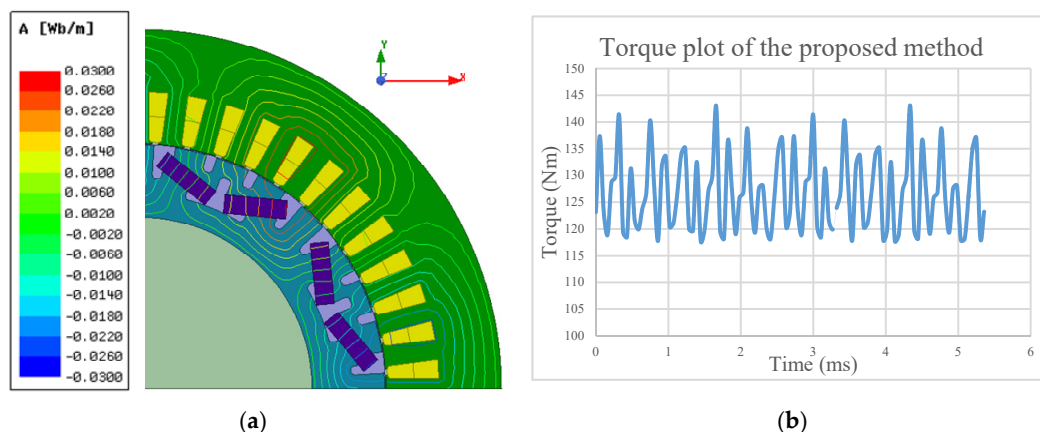
Figure 4. The Optimized design of IPM obtained by the conventional SFLA: (**a**) cross-section; and (**b**) torque plot.
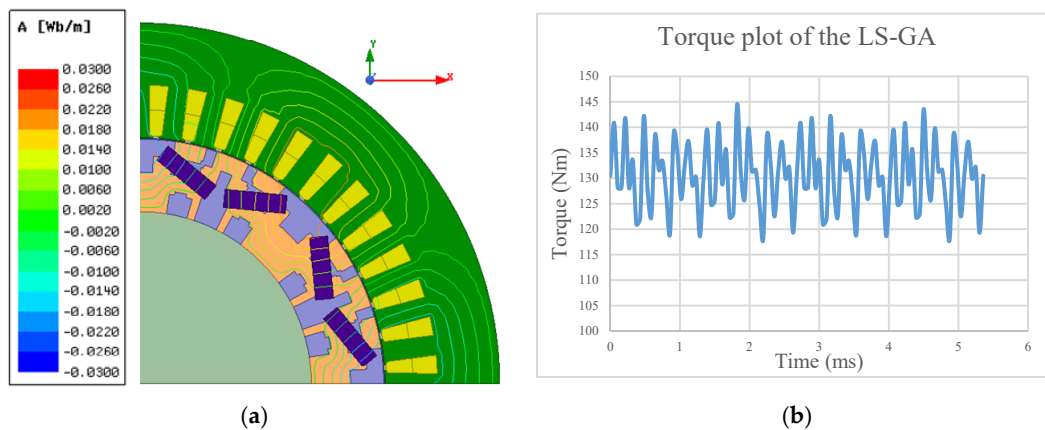
Figure 5. The Optimized design of IPM obtained by the LS-GA: (**a**) cross-section; and (**b**) torque plot.

Figure 6. The Optimized design of IPM obtained by the proposed algorithm: (**a**) cross-section; and (**b**) torque plot.

5. Conclusions

In this paper, a modified shuffled frog-leaping algorithm that is suitable for complex electromagnetic problem is proposed. Compared with conventional SFLA, the proposed algorithm adds a local search step to the global best in each memeplex formed during the optimization and discards some repetitions in the worst candidates improvement process, and hence the algorithm has a

better exploitation abilities with comparable time cost. To examine the performance of the proposed algorithm, it was applied to the topology optimization of an IPM prototype, and the results were then compared with the TO results of two other evolutionary algorithms, namely the conventional SFLA and the LS-GA. The LS-GA finds the best result with the longest optimization time, while the conventional SFLA finds the result with the least objective value using the shortest time. The optimized result found by the proposed method is 95% of the result found by LS-GA using 50% of its optimization time, and hence the proposed algorithm has the most efficient trade-off between the optimization time and the objective values, making it suitable for the optimization of electrical machine.

Conflicts of Interest: The authors declare no conflict of interests.

References

1. Acar, E.; Du, J.; Kim, Y.Y.; Saka, M.P.; Sigmund, O.; Silva, E.C.N. Special issue for the 13th world congress on structural and multidisciplinary optimization—Editorial note. *Struct. Multidiscip. Optim.* **2020**, *48*, 1–2. [CrossRef]
2. Rozvany, G.I.N. A critical review of established methods of structural topology optimization. *Struct. Multidiscip. Optim.* **2008**, *37*, 217–237. [CrossRef]
3. Fan, Z.; Xia, Q.; Lai, W.; Xia, Q.; Shi, T. Evolutionary topology optimization of continuum structures with stress constraints. *Struct. Multidiscip. Optim.* **2018**, *59*, 647–658. [CrossRef]
4. Blachowski, B.; Tauzowski, P.; Lógó, J. Yield limited optimal topology design of elastoplastic structures. *Struct. Multidiscip. Optim.* **2020**, *61*, 1–24. [CrossRef]
5. Li, B.; Li, J.; Tang, K.; Yao, X. Many-Objective Evolutionary Algorithms. *ACM Comput. Surv.* **2015**, *48*, 1–35. [CrossRef]
6. Woo, D.-K.; Kim, I.-W.; Jung, H.-K. Optimal Rotor Structure Design of Interior Permanent Magnet Synchronous Machine based on Efficient Genetic Algorithm Using Kriging Model. *J. Electr. Eng. Technol.* **2012**, *7*, 530–537. [CrossRef]
7. Ho, S.L.; Chen, N.; Fu, W. An Optimal Design Method for the Minimization of Cogging Torques of a Permanent Magnet Motor Using FEM and Genetic Algorithm. *IEEE Trans. Appl. Supercond.* **2010**, *20*, 861–864. [CrossRef]
8. Ho, S.L.; Xu, G.; Fu, W.; Yang, Q.; Hou, H.; Yan, W. Optimization of Array Magnetic Coil Design for Functional Magnetic Stimulation Based on Improved Genetic Algorithm. *IEEE Trans. Magn.* **2009**, *45*, 4849–4852. [CrossRef]
9. Hakimi-Asiabar, M.; Ghodsypour, S.H.; Kerachian, R.; Ghodsypour, S.H. Multi-objective genetic local search algorithm using Kohonen's neural map. *Comput. Ind. Eng.* **2009**, *56*, 1566–1576. [CrossRef]
10. Boryczka, U.; Szwarc, K. Selected variants of a Memetic Algorithm for JSP–a comparative study. *Int. J. Prod. Res.* **2019**, *57*, 7142–7157. [CrossRef]
11. Neri, F.; Cotta, C. Memetic algorithms and memetic computing optimization: A literature review. *Swarm Evol. Comput.* **2012**, *2*, 1–14. [CrossRef]
12. Amiri, B.; Fathian, M.; Maroosi, A. Application of shuffled frog-leaping algorithm on clustering. *Int. J. Adv. Manuf. Technol.* **2009**, *45*, 199–209. [CrossRef]
13. Eusuff, M.; Lansey, K.E.; Pasha, F. Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization. *Eng. Optim.* **2006**, *38*, 129–154. [CrossRef]
14. González, M.G.; Jurado, F.; Pérez, I. Shuffled frog-leaping algorithm for parameter estimation of a double-cage asynchronous machine. *IET Electr. Power Appl.* **2012**, *6*, 484. [CrossRef]
15. Elbeltagi, E.; Hegazy, T.; Grierson, D. Comparison among five evolutionary-based optimization algorithms. *Adv. Eng. Inform.* **2005**, *19*, 43–53. [CrossRef]

16. Sasaki, H.; Igarashi, H. Topology Optimization Accelerated by Deep Learning. *IEEE Trans. Magn.* **2019**, *55*, 1–5. [CrossRef]

17. Luersen, M.; Le Riche, R.; Guyon, F. A constrained, globalized, and bounded Nelder? Mead method for engineering optimization. *Struct. Multidiscip. Optim.* **2004**, *27*, 43–54. [CrossRef]

18. Ho, Y.; Pepyne, D. Simple Explanation of the No-Free-Lunch Theorem and Its Implications. *J. Optim. Theory Appl.* **2002**, *115*, 549–570. [CrossRef]