Differential Evolution for Power Electronic Circuit Optimization

Zhi-Hui Zhan, *Member, IEEE* and Jun Zhang, *Senior Member, IEEE*

***Abstract*—Power electronic circuit (PEC) design and optimization is a significant problem in both scientific and engineering communities. Due to the complex search space of the PEC optimization problem, lots of works have tried to use evolutionary computation (EC) algorithms to solve it, and have gained great progress. However, some existing EC based algorithms for PEC are still complex in algorithm design, or the solutions are still needed to be improved when considering the solution accuracy. Therefore, design a simpler yet powerful algorithm to solve the PEC problem efficiently is in great need. This paper makes the first attempt to proposing a novel differential evolution (DE), which is a kind of new, simple, yet efficient EC algorithm for the PEC design and optimization. The advantage of this paper is that the DE algorithm is the first time directly applied to PEC design and optimization, making the approach very simple for use. The results are compared with those obtained by using genetic algorithm (GA), particle swarm optimization (PSO), and brain storm optimization (BSO). Results show that the DE algorithm outperforms GA, PSO, and BSO in our PEC design and optimization study.**

***Keywords*—** Differential evolution (DE), power electronic circuit (PEC), optimization

I. Introduction

During the past several decades, power electronics circuit (PEC) design has been fast developed mainly due to the significances of PEC in various applications, such as in the industrial, commercial, residential, aerospace, military, and utility areas [1][2]. The PEC design is a typical optimization problem that lots of the components in the circuit, such as resistors, capacitors, and inductors, have to be optimally designed so as to obtain better circuit performance [3]. In the early years, the engineers might use the trial-and-error process to determine the component values of the PEC. However, as the complexity increased in PEC, efficient optimization methods become great need because suitable components design and parameters tuning of PEC often challenge the engineers. Therefore, various optimization approaches such as heuristic method [4], knowledge based method [5], gradient descent or hill-climbing method [6][7], and simulated annealing method [8], have been proposed for some specific analog circuit design automation. However, these approaches are very sensitive to the initial solution and might be inefficient to search globally when the problems are complex [9]. As a result, the obtained values for the circuit components may be sub-optimal, leading to low satisfaction when used in practical applications.

As a kind of popular global optimization tool, evolutionary computation (EC) algorithms have been fast developed and have been applied in many real-world problems such as multicast routing optimization [10], intelligent transportation scheduling [11][12], and wireless sensor networks design [13][14]. Several EC algorithms such as the genetic algorithm (GA) [9], ant colony optimization (ACO) [15], particle swarm optimization (PSO) [16], and brain storm optimization (BSO) [17] have also been reported successfully applied to solve the PEC problem. These works have provided very encouraging results and have shown great promising of using EC algorithms in the PEC optimization problem. However, PEC is a very tough optimization problem due to its complex circuit structure and the multimodal search space. Moreover, Zhan *et al*. [18] argued that the PEC problem should be optimized in the free search range, further making the PEC model nearer to practical application but at the same time making the problem more challenging. Although an orthogonal learning PSO (OLPSO) has been applied to PEC optimization with free search range and obtained promising performance, the OLPSO approach seems to be more complex than traditional PSO [19]. Therefore, we still hope to find a simpler yet powerful algorithm to solve the PEC problem efficiently. Later, Zhang *et al*. [20] proposed to use normalization group strategy to extend BSO for PEC design and optimization. The normalization group based BSO (NGBSO) shown better performance than GA and traditional PSO. However, the solutions are still needed to be improved when considering the solution accuracy.

Differential evolution (DE) is a kind of simple yet efficient EC algorithm [21]. Due to its simple algorithm implementation and efficient performance in many kinds of

Z.-H. Zhan and J. Zhang are with the Department of Computer Science, Sun Yat-Sen University, Guangzhou, 510275, China, with the School of Advanced Computing, Sun Yat-Sen University, Guangzhou, 510275, China, with the Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education, China, with the Engineering Research Center of Supercomputing Engineering Software (Sun Yat-sen University), Ministry of Education, China, and also with the Key Laboratory of Software Technology, Education Department of Guangdong Province, China. Jun Zhang is the corresponding author, email: issai@mail.sysu.edu.cn.

This work was partially supported by the National Natural Science Foundations of China (NSFC) with No. 61402545, the Natural Science Foundations of Guangdong Province for Distinguished Young Scholars with No. 2014A030306038, the Project for Pearl River New Star in Science and Technology with No. 201506010047, the NSFC Key Program with No. 61332002, the NSFC for Distinguished Young Scholars with No. 61125205, the Fundamental Research Funds for the Central Universities (15lgzd08), and the National High-Technology Research and Development Program (863 Program) of China No.2013AA01A212.

optimization problems, the DE algorithm and its variants have been extensively studied [22][23][24] and have been applied to many real-world optimization problems [25][26]. The widely successful use of DE in various real-world problems has shown that DE is a promising approach for the PEC problem.

To the best our knowledge, this is the first attempt to propose the DE algorithm, which is a kind of new, simple, yet efficient EC algorithm for the PEC design and optimization. The advantage of this paper is that the DE algorithm is the first time directly applied to PEC design and optimization, making the approach very simple for use. The results are compared with those obtained by using GA, PSO, and BSO.

The rest of this paper is organized as follows. In Section II, the brief description of the DE algorithm and the PEC optimization problem are presented. Then Section III proposes to use DE to solve the PEC problem. Section IV verifies the performance of DE in optimizing the PEC by comparing with the well-studied GA, PSO, and BSO algorithms for PEC in the literature. Finally, conclusions are summarized and future work is highlighted in Section V.

II. DE AND PEC

A. DE

The DE algorithm is a typical EC algorithm with three basic operations for the population reproduction: mutation, crossover, and selection.

In every generation, a population P first goes through mutation. The mutation operation of DE is very special in that it uses a linear combination of a base vector and one differential vector or more to generate a mutated vector. For example, for every individual X_i (i =1, 2, .., N, where N is the population size) in P, its mutated vector v_i is generated by:

$$V_i = X_{r_1} + F \cdot (X_{r_2} - X_{r_3}) \quad (1)$$

where r_1, r_2, and r_3 are three randomly selected individuals, and are also different from i. In this mutation scheme, the difference between individuals r_2 and r_3 is used as the mutation step while factor F controls the step scale. After a mutated population V is created, it will go through a crossover process with the parent population P. The crossover operation for DE can be in *binary* or *exponential*. Here, without loss of generality, we only illustrate the binary crossover, as it is used in most cases. The crossover operation recombines every pair of individuals of (V_i, X_i) to generate a new individual U_i as shown in:

$$u_{id} = \begin{cases} v_{id}, & \text{if } rand(0,1) \le CR \,||\, d == drand \\ x_{id}, & \text{otherwise} \end{cases} \quad (2)$$

where $rand(0,1)$ is a random number uniformly distributed within interval [0, 1], D is the number of dimensions, CR is the crossover rate which controls how many dimensions of the newly generated individual are from the mutated vector V_i, and $drand$ is a randomly selected index to make sure that at least one dimension of the mutated vector will enter into the newly generated individual.

The crossover process creates a temporary population U, which is evaluated and then enters into the selection procedure. This procedure uses a pair-wise comparison of U and P. As shown in (3), individual U_i and X_i are compared and the better one will enter into the next generation.

$$X_i^{new} = \begin{cases} X_i, & \text{if } fitness(x_i) \text{ is better than } fitness(u_i) \\ U_i, & \text{otherwise} \end{cases} . \quad (3)$$

B. PEC

PEC contains lots of components, including resistors, capacitors, inductors, etc. Fig. 1 is a block diagram of a basic PEC. Note that this figure to show a typical PEC was also previously used by us in our work of using other EC algorithms to solve PEC [9][18][20]. In the figure, we can see that a typical PEC can be decoupled into two parts. One part is the power conversion stage (PCS) that transfers input power to the output load. Another part is the feedback network (FN) that can control and coordinate the power.

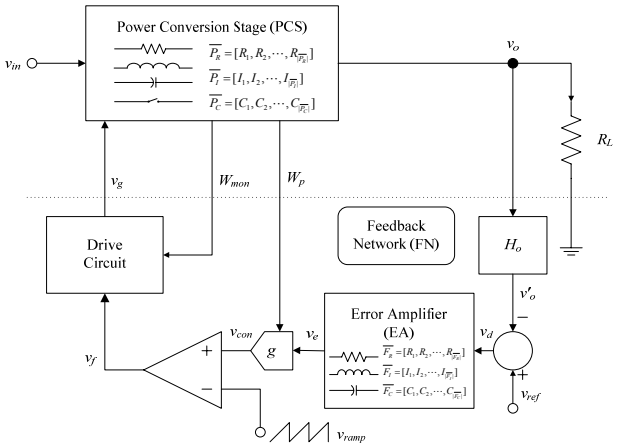

Figure 1. A block diagram of PEC.

In the PCS part, the power from the input source v_{in} to the output load R_L. The PCS consists of P_R resistors, P_I inductors, and P_C capacitors. In the FN part, it also consists lots of components that we denote as F_R resistors, F_I inductors, and F_C capacitors. The circuit works generally as follows: First, the PCS part transfers the input source v_{in} to the output load R_L, the output voltage is v_o. Then a signal conditioner H in the FN circuit converts v_o into a suitable form $v_o^{'}$ which is used to compared with the reference voltage v_{ref}. Their difference v_d is then sent to an error amplifier in order to obtain a output v_e. Then v_e combines with the feedback signals W_p from the PCS part to form an output control voltage v_{con}. Then v_{con} is modulated by a pulse-width modulator to give a feedback voltage v_g to the PCS part.

As we discussed in our previous works [9][18][20], the PCS part of a PEC is always with static characteristics and the component values are relative stable. Therefore, the components in PCS are not optimized by our approach but set to some fix values. Herein, the components in the FN part which are crucial to the circuit performance are only optimized by the DE approach.

III. DE FOR PEC OPTIMIZATION

A. Solution Encoding

When using DE algorithm to solve the PEC, the components in the FN part can be represented with the use of a vector $\boldsymbol{X}$. Specifically, the representation of each solution in DE for optimizing the FN components is coded as:

$$X = [\overline{F}_R \quad \overline{F}_I \quad \overline{F}_C] \tag{4}$$

where F_R, F_I, and F_C are the resistors, inductors, and capacitors of the FN part. Also, we can encode the X as:

$$X = [x_1, x_2, \cdots, x_D] \tag{5}$$

where $D = \left|\overline{F}_R\right| + \left|\overline{F}_I\right| + \left|\overline{F}_C\right|$ is the number of the component, and is also regarded as the number of dimension of the problem.

B. Fitness Function

The fitness function definition for FN is according to the proposals in [9] whose main considerations include reducing the settling time and controlling the overshoot. The fitness function is a maximal optimization problem described as:

$$\Phi_{FN}(X) = \sum_{R_L = R_{L\_\min},\delta R_L}^{R_{L\_\max}} \sum_{v_{in} = v_{in\_\min},\delta v_{in}}^{v_{in\_\max}} [F_1(R_L, v_{in}, X) + F_2(R_L, v_{in}, X) + F_3(R_L, v_{in}, X)] + F_4(X) \tag{6}$$

where $R_{L\_\min}$ and $R_{L\_\max}$, $v_{in\_\min}$, and $v_{in\_\max}$ are the minimal and maximal values of R_L and v_{in}, respectively. δR_L and δv_{in} are the step length in varying the values of R_L and v_{in}.

The F_1, F_2, F_3, and F_4 are the four objective functions for the FN as designed in [9]. Specifically, F_1 is to measure the steady-state error of the output voltage v_o; F_2 is to measure the transient response of v_d, including the maximum overshoot and undershoot, and the settling time; F_3 is to control the steady-state ripple voltage on the output v_o; F_4 is to measure the dynamic behaviors during the large-signal change. For more details of the fitness function definitions, refer to [9].

C. Initialization

In the initialization, a set of N solutions are randomly generated as the population. Each solution $\boldsymbol{X_i} = [x_{i1}, x_{i2}, \ldots, x_{iD}]$ represents a potential PEC, where $1 \le i \le N$, N is the population size and D is the number of components in the PEC. Each dimension x_{id} ($1 \le d \le D$) is initialized uniformly and randomly within the search space as:

$$x_{id} = random\ (L_d, U_d) \tag{7}$$

where L_d and U_d are low bound and upper bound of the d^{th} dimension search space respectively. For each solution, DE evaluates its fitness value according to (6).

D. Evolutionary Process

After the initialization, DE goes to the evolutionary process. For each individual X_i in the population, it uses the following three operators named *mutation*, *crossover*, and *selection* to evolve new solutions generation by generation to approach the optimal solution.

Firstly, the mutation operator, X_i uses Eq. (1) to generate its responding mutant solution V_i. It should be noted that, if the value of any dimension v_{id} exceeds the search range $[L_d, U_d]$, it is reset to the corresponding bound.

Secondly, the X_i and V_i crossover according to Eq. (2) to generate a new solution U_i.

Thirdly, the new solution U_i is evaluated by Eq. (6). Then the X_i and U_i compete and is selected as the new solution according to Eq. (3). Herein, as we pursue the maximal fitness value to obtain best circuit performance, we select the better solution with large fitness value.

All the individuals perform the same three mutation, crossover, and selection operators to evolve the solutions better and better, until the termination condition met, e.g., the maximal generations have been reached.

E. The Whole Algorithm

According to the solution encoding scheme, fitness function, and evolutionary process described above, the whole approach flowchart of using DE to solve the PEC problem is shown as Fig. 2.

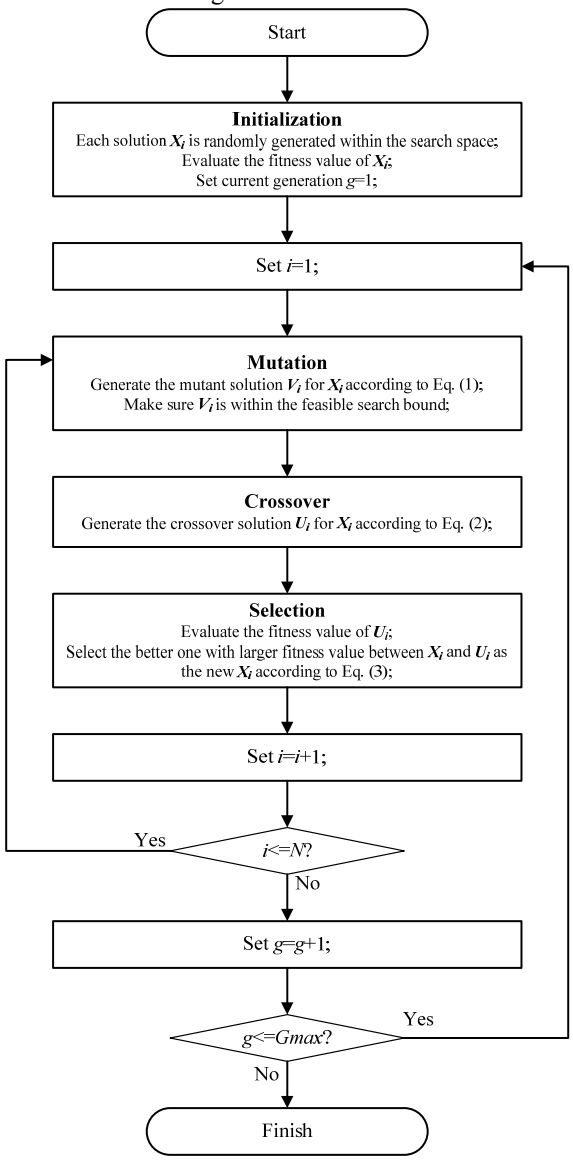

Figure 2. The whole approach flowchart of using DE to solve the PEC problem.

IV. Experimental Studies

A. Circuit Configuration Parameters

In this section, the performance of DE in solving the PEC design and optimization problem is evaluated. The PEC is the same as the one in [9] and [16] where the buck regulator is with overcurrent protection, as shown in Fig. 3. In this circuit, the PCS part is a classical buck converter and the FN part is a proportional-plus-integral controller. In this paper, we do not consider to optimize the components in PCS for that only the L and C are required to be optimized whilst the R_L, r_C, and r_E are assumed to be known as in a practical circuit. Moreover, the PCS part is always with static characteristics and the components L and C are relatively stable [9]. Therefore, the values for L and C are set as 200μH and 1000μF, respectively, according the proposals in [9] and by the considerations of available component values in industry. For FN, all component values are required to be optimized. That is, the components R_1, R_2, R_{C3}, R_4, C_2, C_3, and C_4 in the FN part are optimized by DE and the fitness function is as (2), with their search ranges as shown in Table I.

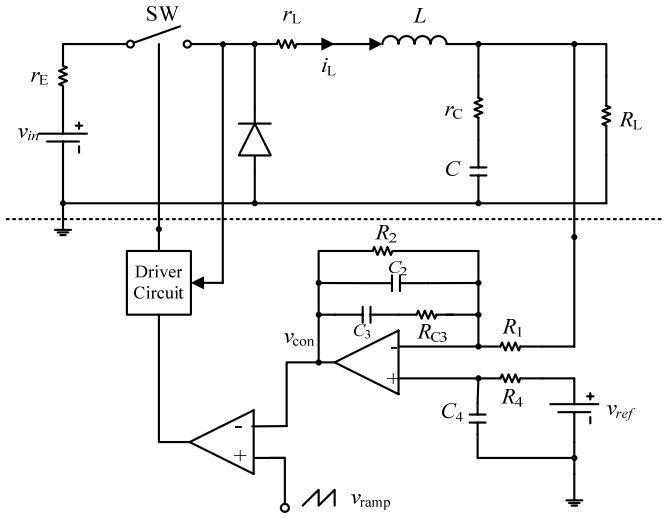

Figure 3. Circuit schematics of the buck regulator with overcurrent protection.

TABLE I. Search Ranges of the Components in the FN Part

Components	Search Range
R_1	[60Ω, 6kΩ]
R_2	[30kΩ, 3000kΩ]
R_{C3}	[470Ω, 47kΩ]
R_4	[100Ω, 10kΩ]
C_2	[0.2μF, 20μF]
C_3	[0.33μF, 33μF]
C_4	[0.18μF, 18μF]

B. Algorithm Parameters Configuration

The performance of DE in optimizing the PEC is evaluated and compared with not only the GA approach proposed in [9] and the PSO approach in [16] because they are two well-studied approaches that optimize PEC in continuous search space, but also with the NGBSO approach because it is a most recent approach for PEC optimization [20].

The parameters of GA, PSO, and BSO are set according to the configurations in their references. The crossover and mutation probabilities of the GA approach are set the same as in [9] where population size N=30, p_x=0.85, and p_m=0.25. The inertia weight ω in PSO linearly decreases from 0.9 to 0.4, while the acceleration coefficients c_1 and c_2 are both set to be 2.0 [16]. The population size of PSO is 30 according to [16]. For NGBSO, according to [20], the parameters N=100, cluster number M=5, $p\_replace$=0.2, p_r=0.005, $p\_center = N(0.4,\ 0.1)$, while other parameters are set according to [17]. For DE, the parameter N=100, F=0.5, and CR=0.1.

In order to make a fair comparison, all the algorithms use the same maximal fitness evaluations (FEs) of 1.5×10^4 as the termination criterion [9][16]. As the evaluation of the fitness function is usually the most expensive computational part in the optimization of PEC, the execution time of different algorithms will be almost the same if they use the same number of FEs. In order to make the comparisons in a statistical sense, the experiment is carried out 10 times independently with each approach and the average results are used for comparison.

C. Comparisons on Fitness Quality

The results of GA, PSO, NGBSO, and DE are compared in Table II where the "Mean" stands for the average fitness value of the 10 independent runs and "Std. Dev" is the standard deviation. Moreover, the "Median" fitness value and the "Best" fitness value among the 10 runs are given and compared in the Table II.

TABLE II. Result Comparisons of Different Approaches

Approach	GA	PSO	NGBSO	DE
Mean	131.205	147.417	158.826	**184.732**
Std. Dev	**2.391**	24.971	18.4707	11.018
Best	134.197	**194.863**	168.005	192.652
Median	131.699	135.249	167.834	**188.052**
Worst	127.332	134.897	123.647	**159.377**
Success #	0	2	8	**10**

The 'Success #' is the number of runs that can find final solution with a fitness larger than 150.

It can be observed from the table that DE achieves the best results among all the four approaches when measured by the mean fitness value. Moreover, although the "Best" fitness solution is slightly beaten by PSO, the "Best" fitness solution obtained by is still larger than 190, indicating the strong global search ability of DE. By comparing the "Median" fitness, DE also performs best among all the 4 approaches, indicating that DE can obtain high quality solution at most of the time. The obtained component values in the "Median" fitness solution optimized by different approaches are presented in Table III.

TABLE III. OPTIMIZED COMPONENT VALUES IN THE BEST RUN WITH DIFFERENT APPROACHES

Comp-onents	GA	PSO	NGBSO	DE
R_1	432.53 Ω	664.9622 Ω	60.0063 Ω	60 Ω
R_2	1651.37 kΩ	1587.174 kΩ	169.044 kΩ	30 kΩ
R_{C3}	45.6666 kΩ	9.9242 kΩ	492.272 Ω	470Ω
R_4	1.0291 kΩ	792.1815 Ω	6.51734 kΩ	407.465 Ω
C_2	18.3062 μF	10.859 μF	0.2005 μF	0.2 μF
C_3	14.8589 μF	0.33 μF	3.22305 μF	1.626 μF
C_4	7.7432 μF	9.427 μF	0.18026 μF	2.715 μF
Fitness Value	131.699	135.249	167.834	**188.052**

D. Comparisons on Reliability

Besides the high solution quality of DE, the fast optimization speed and strong algorithm reliability of DE are also supported by the comparisons in Table II and Fig. 4. By giving an acceptable fitness value of 150, DE can successfully obtain final solutions with fitness values larger than 150 in all the 10 runs whilst NGBSO and PSO can only succeeds in 8 runs and 2 runs, respectively. The GA approach even totally fails in obtaining solutions with fitness values larger than 150. Therefore, DE is the most reliable algorithm that can obtain high quality solutions to PEC constantly.

The mean convergence characteristics of different approaches are plotted in Fig. 4. The curves show that GA has very weak search ability and falls into very poor local optima quite early. Although PSO and NGBSO can obtain some good fitness in early stage, they stay stagnate for the long time during the late stage. Therefore, they are not efficient enough because of the premature convergence. On the other hand, the figure shows that DE has strong global search ability to avoid local optima. That is, the DE algorithm can improve the fitness value during the whole evolutionary process. This indicates that DE is much reliable and promising for the PEC and may also be promising for other complex real-world application.

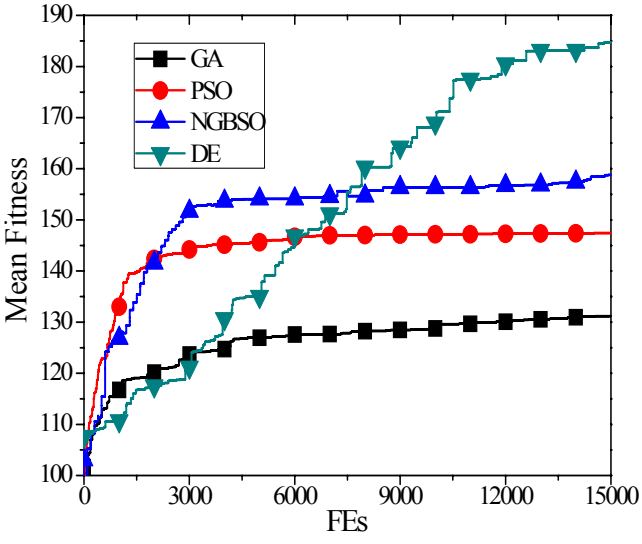

Figure 4. Mean convergence characteristics of different approaches in optimizing the PEC.

E. Comparisons on Simulation Results

Simulations are conducted in this sub-section. In order to make the comparisons clearer, the component values of the PEC are set as the optimized results of the median solutions obtained by GA and DE. The simulation results are plotted and compared in Fig. 5 and Fig. 6. In the simulation results comparison, Fig. 5 gives the results of voltage and Fig. 6 gives the results of current.

The simulation lasts for 90 milliseconds (ms). The input voltage v_{in} is 40 V and the output load R_L is 10 Ω. The simulated startup transients can be compared in the first 30 ms of the figures. It is observed that the circuit with DE-optimized component values has better performance, giving faster settling time. The buck with component values optimized by DE uses only about 5 ms to reach the steady state, while the ones with component values optimized by GA uses about 10 ms.

Fig. 5 and Fig. 6 also show the simulated transient responses under large signal disturbances. On the 30 ms, when the regulator is in steady state, the input voltage is suddenly changed from 40 V to 20 V, with the load still fixed as 10 Ω. As the responses to this change, the output voltage v_o, the control voltage v_{con}, and the inductor current i_L are all disturbed. However, the circuit optimized by DE has much smaller disturbance and shorter response time (less than 5 ms) than the one optimized by GA (more than 10 ms), confirming the advantages of the DE algorithm. Moreover, the overshoot of v_o of the GA-optimized circuit is much larger than that of the DE-optimized circuit.

Similar tests on load disturbances are also studied when the system has reverted a steady state with v_{in} equals 20 V and R_L equals 5 Ω. In this disturbance, R_L is suddenly changed from 10 Ω to 5 Ω on the 60 ms, with the v_{in} being still fixed as 20 V. The simulation results in the figures also show that the DE-optimized circuit has a smaller disturbance response to the change and a shorter time to revert the steady state when compared with GA. Therefore, the proposed DE algorithm can optimize the circuit component values and make the circuit exhibit better dynamic performance.

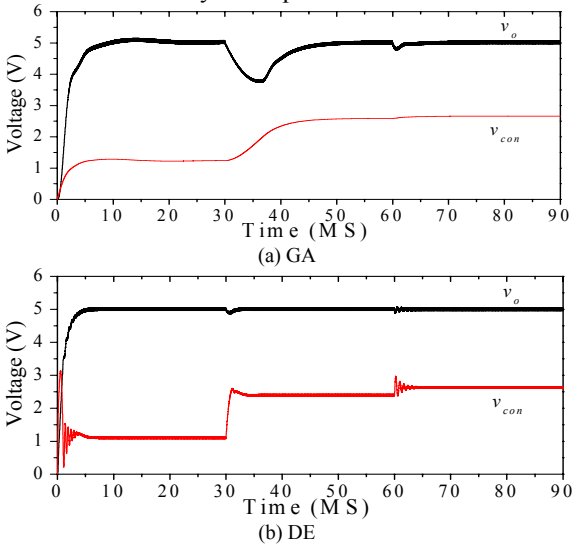

Figure 5. Simulated voltage responses from 0 ms to 90 ms.

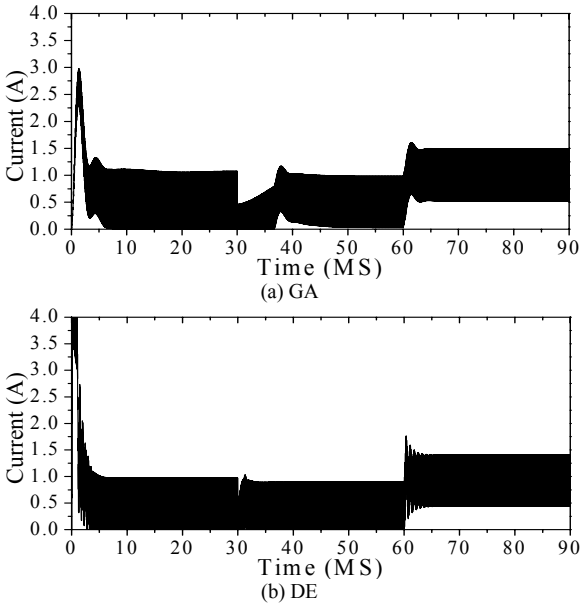

Figure 6. Simulated current responses from 0 ms to 90 ms.

V. CONCLUSIONS

In this paper, we propose to directly apply the simple DE algorithm to solve the PEC problem and have gained encouraging results. By comparing DE with GA, PSO, and NGBSO, the DE algorithm show promising performance in solve this kind of complex optimization.

It should be noted that the DE algorithm adopted in this paper is the traditional DE without any modifications. This is an advantage that the approach is kept to be simple and easy for used. However, in our future work, we will pay attentions to some recent enhanced adaptive DE variants [27][28] and try to obtain better results.

REFERENCES

[1] B. K. Bose, "Recent advance in power electronics," *IEEE Trans. Power Electron.*, vol. 7, no. 1, pp. 2-16, Jan. 1992.

[2] Z. Chen, J. M. Guerrero, and F. Blaabjerg, "A review of the state of the art of power electronics for wind turbines," *IEEE Trans. Power Electron.*, vol. 24, no. 8, pp. 1859-1875, Aug. 2009.

[3] J. G. Kassakian, M. F. Schlecht, and G. C. Verghese, *Principles of Power Electronics*, Addison-Wesley, 1991.

[4] G. J. Sussman and R. M. Stallman, "Heuristic techniques in computer aided circuit analysis," *IEEE Trans. Circuits Syst.*, vol. CAS-22, no. 11, pp. 857–865, Nov. 1975.

[5] R. Harjani, R. A. Rutenbar, and L. R. Carley, "OASYS: A framework for analog circuit synthesis," *IEEE Trans. Comput.-Aided Design*, vol. 8, no. 6, pp. 1247–1266, Jun. 1989.

[6] L. P. Huelsman, "Optimization—A powerful tool for analysis and design," *IEEE Trans. Circuits Systs. I*, vol. 40, no. 7, pp. 431–439, Jul. 1993.

[7] R. E. Massara, *Optimization Methods in Electronic Circuit Design*. New York: Longman, 2000.

[8] E. S. Ochotta, R. A. Rutenbar, and L. R. Carley, "Synthesis of high-performance analog circuits in ASTRX/OBLX," *IEEE Trans. Comput.-Aided Design*, vol. 15, no. 3, pp. 273–294, Mar. 1996.

[9] J. Zhang, H. Chung, W. L. Lo, S. Y. R. Hui, and A. Wu, "Implementation of a decoupled optimization technique for design of switching regulators using genetic algorithm," *IEEE Trans. Power Electron.*, vol. 16, no. 6, pp. 752–763, Nov. 2001.

[10] M. Shen, Z. H. Zhan, W. N. Chen, Y. J. Gong, J. Zhang, and Y. Li, "Bi-velocity discrete particle swarm optimization and its application to multicast routing problem in communication networks," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 12, pp. 7141-7151, Dec. 2014..

[11] Z. H. Zhan, J. Zhang, Y. Li, O. Liu, S. K. Kwok, W. H. Ip, and O. Kaynak, "An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem," *IEEE Trans. Intell. Transport. Syst.*, vol. 11, no. 2, pp. 399-412, Jun. 2010.

[12] J. Garcia-Nieto, A. C. Olivera, and E. Alba, "Optimal cycle program of traffic lights with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 6, pp. 823-839, Dec. 2013.

[13] Z. H. Zhan, J. Zhang, K. J. Du, and J. Xiao, "Extended binary particle swarm optimization approach for disjoint set covers problem in wireless sensor networks," in *Proc. Conf. Technologies and Applications of Artificial Intelligence*, 2012, pp. 327-331.

[14] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: A brief survey," *IEEE Trans. Syst., Man, and Cybern. C*., vol. 41, no. 2, pp. 262-267, March. 2011.

[15] J. Zhang, S. H. Chung, W. L. Lo, and T. Huang, "Extended ant colony optimization algorithm for power electronic circuit design," *IEEE Trans. Power Electron.*, vol. 24, no. 1, pp. 147-162, Jan. 2009.

[16] J. Zhang, Y. Shi, and Z. H. Zhan, "Power electronic circuits design: A particle swarm optimization approach," in *Proc. The 7th International Conference on Simulated Evolution And Learning*, 2008, pp. 605–614.

[17] Z. H. Zhan, J. Zhang, Y. H. Shi, and H. L. Liu, "A modified brain storm optimization," in *Proc. IEEE Congr. Evol. Comput.*., 2012, pp. 1-8.

[18] Z. H. Zhan and J. Zhang, "Orthogonal learning particle swarm optimization for power electronic circuit optimization with free search range," in *Proc. IEEE Congr. Evol. Comput.*, 2011, pp.2563-2570.

[19] Z. H. Zhan, J. Zhang, Y. Li, and Y. H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, pp. 832-847, Dec. 2011.

[20] G. W. Zhang, Z. H. Zhan, K. J. Du, and W. N. Chen, "A normalization group brain storm optimization for power electronic circuit optimization," in *Proc. Genetic Evol. Comput. Conf.*, 2014, pp. 183-184.

[21] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[22] Y. L. Li, Z. H. Zhan, Y. J. Gong, W. N. Chen, J. Zhang, and Y. Li, "Differential evolution with an evolution path: A DEEP evolutionary algorithm," *IEEE Trans. on Cybernetics*, vol. 45, no. 9, pp. 1798-1810, Sept. 2015.

[23] X. F. Liu, Z. H. Zhan, and J. Zhang, "Dichotomy guided based parameter adaptation for differential evolution," in *Proc. Genetic Evol. Comput. Conf*., Madrid, Spain, Jul. 2015, pp. 289-296.

[24] Z. H. Zhan and J. Zhang, "Enhance differential evolution with random walk," in *Proc. Genetic Evol. Comput. Conf.*, Philadelphia, America, Jul. 2012, pp. 1513-1514.

[25] P. P. Menon, J. Kim, D. G. Bates, and I. Postlethwaite, "Clearance of nonlinear flight control laws using hybrid evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 689–699, Dec. 2006.

[26] Y. L. Li, Z. H. Zhan, Y. J. Gong, J. Zhang, Y. Li, and Q. Li, "Fast micro-differential evolution for topological active net optimization," *IEEE Trans. Cybern.*, DOI:10.1109/TCYB.2015.2437282, 2015.

[27] Z. H. Zhan and J. Zhang, "Co-evolutionary differential evolution with dynamic population size and adaptive migration strategy," in *Proc. Genetic Evol. Comput. Conf.*, Dublin, Ireland, Jul., 2011, pp. 211-212.

[28] Z. H. Zhan and J. Zhang, "Self-adaptive differential evolution based on PSO learning strategy," in *Proc. Genetic Evol. Comput. Conf.*, Portland, America, Jul., 2010, pp. 39-46.