Optimizing Power Electronic Circuit Design with Uniform Search Range:

An Orthogonal Learning Particle Swarm Optimization Approach with Predictive Solution Strategy

Ying Lin, Department of Psychology, Sun Yat-sen University, Guangzhou, China

Zhi-Hui Zhan, Department of Computer Science, Sun Yat-sen University, Guangzhou, China

Jun Zhang, Department of Computer Science, Sun Yat-sen University, Guangzhou, China

ABSTRACT

Power electronic circuit (PEC) design is a complicated optimization problem that calls for evolutionary computation (EC) algorithms. The existing EC algorithms for optimizing PEC have difficulty in practical applications because they demand users to carefully define very narrow search ranges for different circuit components. Aiming at this problem, this paper models PEC with uniform search range (USR), where the components' search ranges are set uniformly according to the commonly used ranges in industrial applications. In order to solve the complex PEC problem with USR, an efficient orthogonal learning particle swarm optimization with predictive solution (OLPSO/PS) is proposed. Firstly, OLPSO/PS uses an orthogonal learning strategy to construct a more promising and efficient exemplar to guide particles to fly towards better searching areas dynamically. Secondly OLPSO/PS utilizes the predictive solution (PS) strategy to help save computational burden. OLPSO/PS is compared not only with the well-studied genetic algorithm and PSO for optimizing PEC in the literature, but also with OLPSO without PS and some other well-performed EC algorithms. Results show that OLPSO/PS is more promising in the optimization of PEC with USR, outperforming other algorithms in terms of higher fitness quality, faster optimization speed, stronger reliability, and better simulation results on both voltage and current. Moreover, the effectiveness of OLPSO/PS is further validated on the experiments in a practical circuit.

Keywords: *Particle Swarm Optimization (PSO), Power Electronic Circuit (PEC), Orthogonal Experimental Design (OED), Orthogonal Learning Particle Swarm Optimization (OLPSO)*

DOI: 10.4018/ijsir.2014070103

1. INTRODUCTION

Power electronic circuit (PEC) design is a complicated optimization problem because the circuit components interact with each other to influence the circuit performance. Therefore, the circuit components such as resistors, capacitors, and inductors in the PEC have to be optimally designed in order to obtain better circuit performance (Ebrahimi, Babaei, & Gharehpetian, 2012). This has attracted great attentions in both engineering community and academic community for that PEC supports significant applications in industrial, commercial, residential, aerospace, military, and utility areas (Song & Wang, 2013).

For a long time, suitable components design and parameters tuning of PEC often challenge engineers (Emadi, 2004). Since the 1970s, various optimization approaches such as heuristic method (Sussman & Stallman, 1975), knowledge based method (Harjani, Rutenbar, & Carley, 1989), gradient descent or hill-climbing method (Huelsman, 1993), and simulated annealing method (Ochotta, 1996) 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 (Zhang, Chung, Lo, Hui, & Wu, 2001). Therefore, the obtained values for the circuit components may be sub-optimal, leading to low satisfaction when used in practical applications.

Recently, evolutionary computation (EC) algorithms have been fast developed and have been utilized as optimization techniques in many real-world applications (Li et al., 2014; Li, Zhan, Lin, Zhang, & Luo, 2015; Shen et al., 2014). Zhang et al. (2001) for the first time modeled fitness functions to describe the PEC and proposed a genetic algorithm (GA) to optimize the PEC component values. The PEC problem formulated in (Zhang et al., 2001) is later extensively studied, with ant colony optimization (ACO) (Zhang, Chung, Lo, & Huang, 2009) and particle swarm optimization (PSO) (Zhang, Shi, & Zhan, 2008) being successfully applied to solve the problem. Even though promising results have been obtained in previous studies, one disadvantage of using GA and ACO approaches is that they have to consume a lot of computation before obtaining the good component values while a traditional PSO is often premature convergent. Moreover, in these previous studies, the PEC model is with the circuit components ranges carefully pre-defined and elaborately determined by expert designers. For example, in the above studies, the search range for some resistors is from 470Ω to 47kΩ, but for some others is from 600Ω to 6kΩ, while the search range for some capacitors is from 0.33μF to 33μF, but for some others is from 0.18μF to 18μF. In fact, such specific search ranges are difficult to define for different components in different PECs. Therefore, it is difficult to apply the current approaches in practical applications. Moreover, an impractical issue of such an optimization model is that how do you know this component is within this range while another component should be within another range? By taking these into consideration, this paper goes further to model the PEC problem with uniform search range (USR) (Zhan & Zhang, 2011). That is, the components' search ranges are no longer pre-defined elaborately by expert designers, but are set according to commonly used ranges in practical industrial applications. For example, all the resistors are within a uniform search range from 100Ω to 100kΩ and all the capacitors are within a uniform search range from 0.1μF to 100μF, spanning the wide ranges which are commonly used in practice (Electronics, 2000).

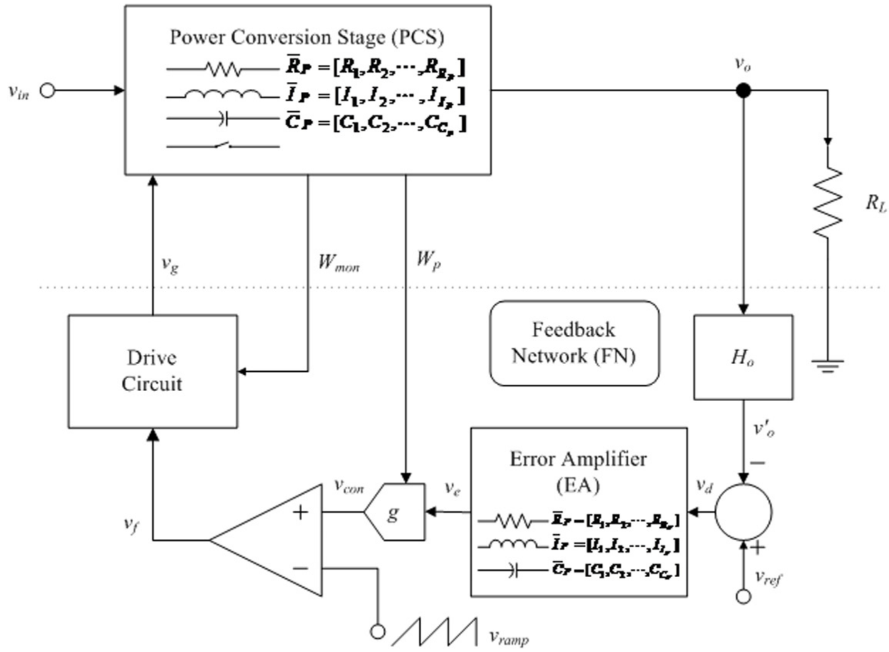
Nevertheless, setting the search range freely and uniformly will make the PEC search space large and complex. This challenges the efficiency of traditional approaches. In order to solve the PEC problem with USR effectively and efficiently, the optimization approach should have both the strong global search ability and fast optimization speed. PSO is a variant of EC paradigm, featured with its simple concept and fast convergence speed, yet a very efficient global optimization technique (Kennedy & Eberhart, 1995; Kennedy, Eberhart, & Shi, 2001). Even though PSO has been extensively studied (Lim & Isa, 2014; Zhan et al., 2013) and have been successfully applied to various practical problems (Kwok et al., 2013; Mazhoud. Hadj-Hamou, Bigeon, & Joyeux, 2013), the exploration of PSO to PEC design and optimization progresses at a slow pace. In some early studies, we applied a traditional PSO to design PEC (Zhang et al., 2008) and also adopted an orthogonal learning PSO (OLPSO) to optimize PEC because OLPSO has faster optimization speed and stronger global search ability than a traditional PSO (Zhan & Zhang, 2011). However, as the fitness evaluation of PEC is an extremely expensive computation process, the optimization approach should be capable to find the global optimum of PEC with as few fitness evaluations as possible.

The OLPSO algorithm was originally designed by Zhan, Zhang, Li, and Shi (2011) where the authors argued that in a traditional PSO, the information of a particle's best experience and its neighborhood's best experience was not efficiently used by a simple summation of the two experiences. In order to discover more useful information lying in the two exemplars and thus combine the information to construct an efficient guidance exemplar in PSO, Zhan et al. (2011) designed an orthogonal learning (OL) strategy for the particles to enhance the learning ability, thus OLPSO is developed to improve the PSO's performance. Advantages of the OL strategy and the OLPSO algorithm have been demonstrated by comparing with PSO using a traditional learning strategy and some other state-of-the-art algorithms on mathematical benchmark functions (Zhan et al. 2011). More importantly, the advantages of OLPSO have attracted great interests and attentions from researchers and engineers to apply it to various real-world applications (Ghosh, Das, & Zafar, 2012; Hu, Ding, Hao, & Ren, 2014). The primary results of using OLPSO to optimize PEC have been presented in (Zhan & Zhang, 2011), showing that OLPSO has strong search ability to find global optimum of PEC. However, one important problem comes out: as calculating the fitness value in PEC is extremely expensive, the OL procedure to construct the guidance exemplar seems to be consumptive because it has to generate a few solutions combining from the personal and neighborhood's experiences, even though the solutions number is very small due to the orthogonal experimental design (OED) method. To sufficiently utilize the solutions generated in the OL procedure, we can think that these solutions are predictive solutions (PS) according to the personal and neighborhood's experiences. Each PS is compared with a randomly selected particle, if the PS has better fitness than the compared particle, then the particle position is replaced by this corresponding PS. This should be much promising not only because that this PS are not discarded during the process, which saves lots of computational burden, but also because that this PS can give the particle another way to improve its solution quality, which helps the particle fly to the global optimum faster.

In order to demonstrate the advantages of OLPSO with PS, termed as OLPSO/PS in this paper, the GA (Zhang et al., 2001) and the traditional PSO (Zhang et al., 2008) approaches, which are well-studied approaches for PEC optimization, are extended to solve PEC with USR and are compared with OLPSO/PS. Moreover, two well-performed EC algorithms named comprehensive learning PSO (CLPSO) (Liang, Qin, Suganthan, & Baskar, 2006) and adaptive differential evolution (JADE) (Zhang & Sanderson 2009) are applied to optimize the PEC design and are compared with OLPSO/PS. The OLPSO algorithm without PS is also compared. Moreover, experiments on a practical circuit are further conducted to test the circuit performance obtained by OLPSO/PS.

The rest of this article is organized as follows. In Section 2, the description of PEC is presented. Section 3 proposes to use OLPSO/PS to solve the PEC problem. Section 4 verifies the performance of OLPSO/PS in optimizing the PEC by comparing not only with some well-studied algorithms in the literature, but also with some well-performed EC algorithms. Also, experiments on a practical circuit are

Figure 1. A block diagram of PEC



conducted to validate the OLPSO/PS-optimized circuit performance. Finally, conclusions are summarized and future work is highlighted in Section 5.

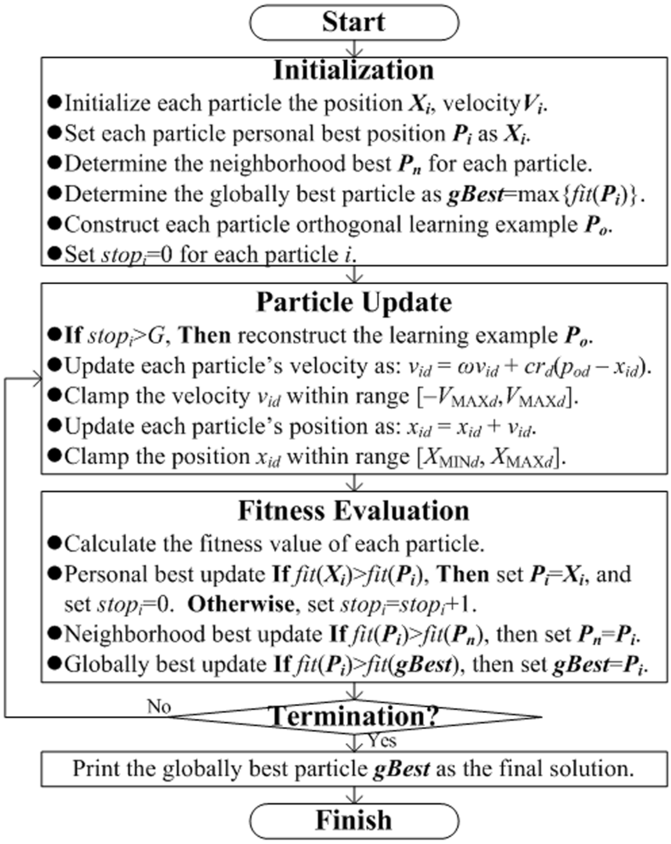
2. PEC

PEC is a circuit that contains a number of components such as resistors, capacitors, and inductors. Figure 1 shows the basic block diagram of a PEC. In this PEC, the circuit can be decoupled into two parts where the first part is the power conversion stage (PCS) and the second part is the feedback network (FN) (Zhang et al., 2001).

The function of PCS is to transfer the power from the input source v_{in} to the output load R_L. It consists of R_P resistors, I_P inductors, and C_P capacitors. On the other hand, FN is the control part that consists of R_F resistors, I_F inductors, and C_F capacitors. There is a signal conditioner H in the FN circuit to convert the PCS output voltage 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. This output is combined with the feedback signals W_p from the PCS part to give 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.

In order to optimize the component values of PEC, the components are coded as the variables and are optimized by the optimization process. As the interactions between the two parts in the optimization are relatively low during the training process, the components in the two parts can be decomposed (Zhang et al. 2001). Since the PCS part is always with static characteristics and the component values are relative stable. Therefore, the components in PCS are not optimized in this paper, but the components in the FN part which are crucial

Figure 2. The flowchart of OLPSO/PS for PEC



to the circuit performance are optimized in this paper.

3. OLPSO/PS FOR PEC

The implementation of using OLPSO/PS to search the optimal component values for a PEC is illustrated in Figure 2 and is described as follows.

3.1. Initialization

PSO is introduced as a swarm intelligence (SI) (Kennedy, Eberhart, & Shi, 2001) algorithm that emulates the swarm behaviors of birds flocking and fish schooling. When searching in a D-dimensional hyperspace, each particle i keeps a position vector $\boldsymbol{X_i} = [x_{i1}, x_{i2}, \ldots, x_{iD}]$ and a velocity vector $\boldsymbol{V_i} = [v_{i1}, v_{i2}, \ldots, v_{iD}]$ to indicate its current state, where i is a positive integer indexing the particle in the swarm and D is the dimensionality of the problem. Moreover, particle i keeps its personal historically best position vector $\boldsymbol{P_i} = [p_{i1}, p_{i2}, \ldots, p_{iD}]$. The best position of all the particles in the i^{th} particle's neighborhood is denoted as $\boldsymbol{P_n} = [p_{n1}, p_{n2}, \ldots, p_{nD}]$.

Therefore, in using OLPSO/PS for PEC optimization, the components in the FN part can be represented with the use of an vectors $X(FN)$. Specifically, the representation of each particle in OLPSO/PS for optimizing the FN components is coded as

$$X(FN) = [\overline{R}_F \quad \overline{I}_F \quad \overline{C}_F] \tag{1}$$

where $\overline{R}_F = [R_1, R_2, \cdots, R_{R_F}]$ are the resistors, $\overline{I}_F = [I_1, I_2, \cdots, I_{I_F}]$ are the inductors, and $\overline{C}_F = [C_1, C_2, \cdots, C_{C_F}]$ are the capacitors.

For each particle i, initialize its position $\boldsymbol{X}_i$ (the circuit components) and its velocity $\boldsymbol{V}_i$ with random values in their corresponding search ranges $[X_{\text{MIN}d}, X_{\text{MAX}d}]$ and $[-V_{\text{MAX}d}, V_{\text{MAX}d}]$, respectively. Set its personal best position $\boldsymbol{P}_i$ as $\boldsymbol{X}_i$. Calculate all the particles' fitness and determine the neighborhood best $\boldsymbol{P}_n$ for each particle. Determine the globally best particle as ***gBest***=max$\{fit(\boldsymbol{P}_i)\}$.

Moreover, construct the orthogonal learning example $\boldsymbol{P}_o = [p_{o1}, p_{o2}, \ldots, p_{oD}]$ via OL strategy for each particle as

$$\boldsymbol{P}_o = \boldsymbol{P}_i \oplus \boldsymbol{P}_n \tag{2}$$

where the symbol ⊕ stands for the OED operation.

OED is one of the most important experimental design methods and has been used to solve many classes of real-world problems successfully. The OED method has been developed for the purpose of tracking experimental problems with many kinds of *factors* and many different choices per factor, namely *levels*. The efficiency of OED lies in its ability of a *general-design* test by using an *orthogonal array* (OA) to construct representative combinations, and its ability of using a *factor analysis* (FA) to find out or predict the best combination by testing only a limited number of cases. The OA for a problem with N factors and Q levels per factor is always denoted by $L_M(Q^N)$, where L denotes the OA and M is the number of combinations of test cases.

In OLPSO, the OED is based on a two levels multiply factors OA because each dimension is regarded as a factor and the choices from P_{id} or P_{nd} of the the d^{th} dimension are regarded as two levels. When using OLPSO to optimize a problem with D dimensions, the construction process of $\boldsymbol{P}_o$ for a particle i based on the OED is described as the following six steps.

Step 1: An OA is generated as $L_M(2^D)$, using the approach detailed in (Zhan et al. 2011), where D is the number of dimensions (factors) and $M = 2^{\lceil \log_2(D+1) \rceil}$. Table 1 is an example using an OA with 7 factors and each factor with 2 levels (denoted as level 1 and level 2).

Step 2: Make up M tested solutions $\boldsymbol{X}_j$ ($1 \le j \le M$) according to the OA by selecting the corresponding value from $\boldsymbol{P}_i$ or $\boldsymbol{P}_n$. Here, for each dimension d of each solution $\boldsymbol{X}_j$, if OA[j][d] is 1, then $x_{jd}=p_{id}$; otherwise, selects $x_{jd}=p_{nd}$.

Step 3: Evaluate each tested solution $\boldsymbol{X}_j$ ($1 \le j \le M$), and record the best solution as $\boldsymbol{X}_b$. The fitness of each combination is listed in the last column of Table 1. For example, the best combination $\boldsymbol{X}_b$ is $\boldsymbol{X}_5$ for this maximization problem.

Step 4: Calculate the effect of each level on each factor and determine the best level for each factor using the FA process.

The FA is to calculate the contributions of different levels on each factor. The process works as follows. Let f_m denote the experimental result of the m^{th} ($1 \le m \le M$) combination and S_{dq} denote the effect of the q^{th} ($1 \le q \le Q$) level in the d^{th} ($1 \le d \le D$) factor. The calculation of S_{dq} is to add up all the f_m in which the level is q in the d^{th} factor, and then divide the total count of z_{mdq}, as shown in (3) where z_{mdq} is 1 if f_m is the q^{th} level in the d^{th} factor, otherwise, z_{mdq} is 0.

$$S_{dq} = \frac{\sum_{m=1}^{M} f_m \times z_{mdq}}{\sum_{m=1}^{M} z_{mdq}} \tag{3}$$

In this way, the effect of each level on each factor can be calculated and compared, as shown in Table 1. For example, when we calculate the effect of level 1 on factor A, denoted by element

Table 1. An Example of orthogonal experimental design with seven factors and two levels

OED	Factors	A	B	C	D	E	F	G	Fitness
Combinations According to OA	$\boldsymbol{X}_1$	1	1	1	1	1	1	1	f_1
	$\boldsymbol{X}_2$	1	1	1	2	2	2	2	f_2
	$\boldsymbol{X}_3$	1	2	2	1	1	2	2	f_3
	$\boldsymbol{X}_4$	1	2	2	2	2	1	1	f_4
	$\boldsymbol{X}_5$	2	1	2	1	2	1	2	$\boldsymbol{f_5}$ (max)
	$\boldsymbol{X}_6$	2	1	2	2	1	2	1	f_6
	$\boldsymbol{X}_7$	2	2	1	1	2	2	1	f_7
	$\boldsymbol{X}_8$	2	2	1	2	1	1	2	f_8
FA for Contributions	L_1	$\boldsymbol{S}_{\mathbf{A1}}$	S_{B1}	S_{C1}	S_{D1}	$\boldsymbol{S}_{\mathbf{E1}}$	$\boldsymbol{S}_{\mathbf{F1}}$	S_{G1}	
	L_2	S_{A2}	$\boldsymbol{S}_{\mathbf{B2}}$	$\boldsymbol{S}_{\mathbf{C2}}$	$\boldsymbol{S}_{\mathbf{D2}}$	S_{E2}	S_{F2}	$\boldsymbol{S}_{\mathbf{G2}}$	
Best	$\boldsymbol{X}_p$	L_1	L_2	L_2	L_2	L_1	L_1	L_2	f_p
Make up $\boldsymbol{P}_o$	$\boldsymbol{P}_o$	L_2	L_1	L_2	L_1	L_2	L_1	L_2	if $f_5>f_p$
		L_1	L_2	L_2	L_2	L_1	L_1	L_2	if $f_p>f_5$

A1, the experimental results of $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, $\boldsymbol{X}_3$, and $\boldsymbol{X}_4$ are summed up for Eq. (3) because only these combinations are involved in level 1 of factor A. Then the sum divides the combination number (4 in this case) to yield S_{dq} (S_{A1} in this case). With all the S_{dq} calculated, the best combination of the levels can be determined by selecting the level of each factor that provides the highest-quality S_{dq}. For example, for a maximization problem, if $S_{\mathrm{A1}}>S_{\mathrm{A2}}$, then the better level of factor A is level 1.

Step 5: Derive a predictive solution $\boldsymbol{X}_p$ with the levels determined in Step 4 and evaluate $\boldsymbol{X}_p$.

Step 6: Compare $f(\boldsymbol{X}_b)$ and $f(\boldsymbol{X}_p)$ and the level combination of the better solution is used to construct the vector $\boldsymbol{P}_o$. In the example of Table 1, f_5 is compared with $f(\boldsymbol{X}_p)$, if f_5 is better than $f(\boldsymbol{X}_p)$, then $\boldsymbol{P}_o$ is constructed with the level combination of $\boldsymbol{X}_5$; otherwise, $\boldsymbol{P}_o$ is constructed with the level combination of $\boldsymbol{X}_p$.

After the initialization, set $stop_i$=0 for each particle i, and then go to the following evolutionary process.

3.2. Particle Update

In a traditional PSO, the particle i is updated guided by the linear sum of personal historically best position $\boldsymbol{P}_i$ and its neighborhood best position $\boldsymbol{P}_n$. The difference between OLPSO and the traditional PSO is that OLPSO use an OL strategy to combine the information of $\boldsymbol{P}_i$ and $\boldsymbol{P}_n$ to form a better guidance vector $\boldsymbol{P}_o$, as shown in (2). Then the particle's flying velocity is thus adjusted as

$$v_{id} = \omega\times v_{id} + c\times r_d\times(p_{od} - x_{id}) \tag{4}$$

where ω is the inertia weight to control the exploration and exploitation abilities of the algorithm. Parameters c is the acceleration coefficient which is set to 2.0 (Zhan et al. 2009). The r_d is a randomly generated value within range [0, 1] for the d^{th} dimension. In order to control the flying velocity within a reasonable range, a positive value $V_{\mathrm{MAX}d}$ (can be set to 20% of the search range of the corresponding dimension (Zhan et al. 2009)) is used to clamp the updated velocity. If $|v_{id}|$ exceeds $V_{\mathrm{MAX}d}$, then it is set to $sign(v_{id})V_{\mathrm{MAX}d}$.

After the velocity update, the particle updates its position as

$$x_{id} = x_{id} + v_{id} \tag{5}$$

Similarly, if the updated position x_{id} is out of the search range [$X_{\text{MIN}d}$, $X_{\text{MAX}d}$], the position is set to the corresponding bound.

It should be noted that before the particle update in every generation, the particle i will first check whether it has stagnated for a long time. If its fitness has not been improved for more than G generation, i.e., $stop_i>G$, then the particle should reconstruct the orthogonal learning example $\boldsymbol{P_o}$ via OL strategy (Eq. (2)) before the velocity and position update. If the $\boldsymbol{P_o}$ has been reconstructed, we should reset $stop_i=0$.

3.3. Fitness Evaluation

The fitness function definition for FN is according to the proposals in (Zhang et al. 2001), whose main considerations includes reducing the settling time and controlling the overshoot. The fitness function is described as:

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

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 (Zhang et al. 2001). 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 (Zhang et al. 2001).

Therefore, in OLPSO/PS, after the particle velocity and position update, the new position $\boldsymbol{X_i}$ is evaluated. Moreover, the personal historically best position is updated if the new position has a better fitness value. That is, if $fit(\boldsymbol{X_i})>fit(\boldsymbol{P_i})$, then set $\boldsymbol{P_i}=\boldsymbol{X_i}$, and set $stop_i=0$; otherwise, set $stop_i=stop_i+1$. The particle's neighborhood best position $\boldsymbol{P_n}$ and the population's globally best position ***gBest*** are also updated if necessary. That is, if $fit(\boldsymbol{P_i})>fit(\boldsymbol{P_n})$, then set $\boldsymbol{P_n}=\boldsymbol{P_i}$; if $fit(\boldsymbol{P_i})>fit(\boldsymbol{gBest})$, then set $\boldsymbol{gBest}=\boldsymbol{P_i}$.

The new position is evaluated and can replace the $\boldsymbol{P_i}$ or $\boldsymbol{P_n}$ if it has a better fitness value. Then the algorithm goes on for the next generation until termination.

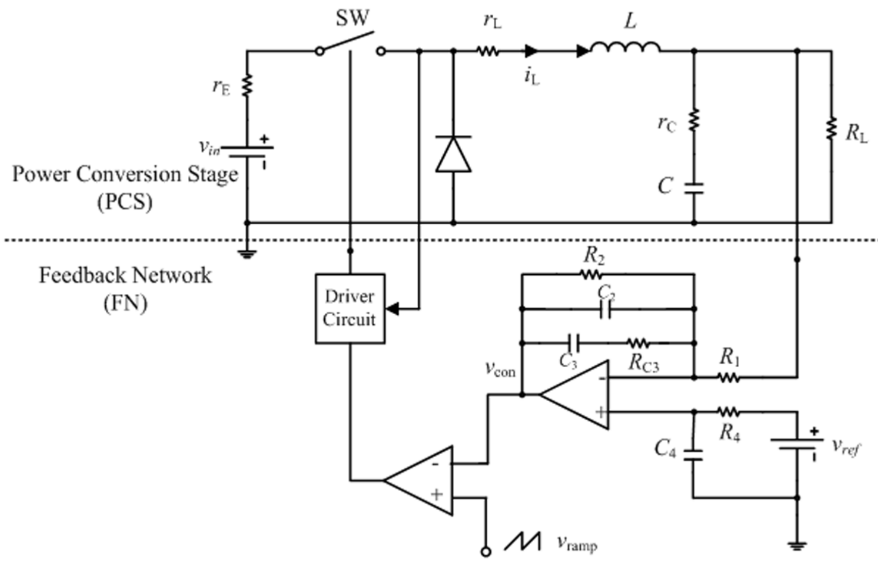
The operations in the "*Particle Update*" and "*Fitness Evaluation*" are performed on each particle in every generation. If the number of iterations or the number of fitness evaluations reaches the pre-defined maximum, then output the globally best particle ***gBest*** as the final solution. The program terminates.

3.4. PS Strategy to Save Computational Burden

As calculating the PEC solution fitness value is an extreme expensive process, the OL strategy used in OLPSO seems to be consumptive because it has to generate a number of solutions before it can determine the best combination of the personal and neighborhood's experiences. In this paper, a novel strategy is proposed to efficiently utilize these generated solutions to save the computational burden.

The strategy is very simple. In the process of OLPSO using OED to construct the guidance exemplar for a particle i, as described in Section 3.1, Step 3 is to generate a number of solutions by combining the personal and neighborhood's experiences. However, in the original OLPSO algorithm, these solutions are just utilized for FA to help determine the promising level of each factor. Although this is very important for constructing the good guidance exemplar, the information in these generated solutions is not sufficiently used. From another point of view, we can regard these solutions as predictive solutions

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



(PS) because each of them is a prediction from the corresponding personal and neighborhood's experiences. Moreover, these generated solutions can be regarded as new solutions that can be added into the population if they have good fitness values, which can increase the diversity of the population.

By considering the motivations that the PS solutions can be used to compete for surviving in the population to increase the diversity and also can save the computational burden, in our OLPSO/PS algorithm, when generate a test solution $\boldsymbol{X}_j$ ($1 \le j \le M$) according to the OA, the fitness value of $\boldsymbol{X}_j$ is compared with the fitness value of a random selected particle r in the population, where $r \ne i$. If $\boldsymbol{X}_j$ has a better fitness than particle r does, the position and fitness of particle r are replaced by those of $\boldsymbol{X}_j$.

We call this strategy the PS strategy that all the predictive solutions in the OL procedure are not only used for combining a promising guidance exemplar as in original OLPSO, but also are used as new solutions to compete for surviving into the population. Such a strategy can make full use of the PS information to save lots of computational burden.

4. PEC DESIGN AND RESULTS COMPARISONS

4.1. Circuit Configurations

In this section, the OLPSO/PS algorithm is applied to the PEC design and optimization problem. The PEC is the same as the one in (Zhang et al., 2001) and (Zhang et al., 2008) where the buck regulator is with overcurrent protection, as shown in Figure 3.

In this circuit, the PCS part is a classical buck converter and the FN part is a proportional-plus-integral controller. For PCS, 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 (Zhang et al. 2009). Therefore, the components

Table 2. Search ranges of the components in the FN part

Components	Search Ranges
R_1	[100Ω, 100kΩ]
R_2	[100Ω, 100kΩ]
R_{C3}	[100Ω, 100kΩ]
R_4	[100Ω, 100kΩ]
C_2	[0.1μF, 100μF]
C_3	[0.1μF, 100μF]
C_4	[0.1μF, 100μF]

in PCS are not optimized in this paper, but the values are set as 200μH and 1000μF for L and C, respectively, according to the proposals in (Zhang et al., 2001) 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 OLPSO/PS and the fitness function is as (6). Moreover, the required specifications of the whole PEC are listed as follows:

1. Input voltage range v_{in}: 20 ~ 40 V
2. Output load range R_L: 5 ~ 10 Ω
3. Nominal output voltage: 5 V±1%
4. Switching frequency: 20 kHz
5. Maximum settling time: 20 ms

As argued in this paper that in the previous studies, the components' search ranges are carefully pre-defined by expert designers in order to make the optimization problem easy for optimization approaches to search. However, such specific search ranges are difficult to define for different PECs. In this paper, we use the USR to set the components search ranges freely according to uniform and commonly used ranges. That is, the search range for resistors R_1, R_2, R_{C3}, and R_4 are set to be [100Ω, 100kΩ] and the search range for capacitors C_2, C_3, and C_4 are set to be [0.1μF, 100μF], as given in Table 2.

4.2. Algorithm Configurations

The performance of OLPSO/PS in optimizing the PEC is evaluated and compared with both the GA approach proposed in (Zhang et al., 2001) and the PSO approach in (Zhang et al., 2008) because they are two well-studied approaches that optimize PEC in continuous search space. Moreover, the OLPSO algorithm (Zhan & Zhang, 2011) and two other well-performed optimization algorithms named CLPSO (Liang et al., 2006) and JADE (Zhang & Sanderson, 2009) are used for comparisons.

The parameters of GA and PSO are set according to the configurations in their references. The crossover and mutation probabilities of the GA approach are set the same as that in (Zhang et al., 2001) where p_x=0.85 and p_m=0.25. The inertia weight ω in PSO, OLPSO, and OLPSO/PS linearly decreases from 0.9 to 0.4, while their acceleration coefficients are all set to be 2.0 (Zhang et al., 2008). The parameter G in OLPSO and OLPSO/PS is set to be 5 (Zhan et al., 2011). The parameters of CLPSO and JADE are set according to their original proposals in (Liang et al., 2006) and (Zhang & Sanderson, 2009), respectively. For the population size and the maximal generation number, they are set to be 30 and 500 respectively in both GA and PSO as proposed in (Zhang et al., 2001) and (Zhang et al., 2008). The population size of OLPSO, OLPSO/PS, CLPSO, and JADE is set to 40, 40, 40, and 30, respectively, according to

Table 3. Experimental result comparisons of different approaches

Approaches	Mean	Std. Dev	Best	Median	Worst	errFEs	Success #
GA	109.636	9.0058	127.494	109.502	97.191	×	0
PSO	152.110	21.2900	192.304	137.879	137.699	3817	8
CLPSO	155.902	13.9257	191.638	149.147	138.919	8588	14
JADE	135.711	25.1273	183.208	135.361	95.973	8681	8
OLPSO	183.748	14.1892	192.962	192.425	137.924	3675	28
OLPSO/PS	**190.532**	**10.1865**	**192.992**	**192.858**	**153.288**	**2577**	**30**

the proposals in (Zhan et al., 2011; Liang et al., 2006; Zhang & Sanderson, 2009). However, 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 (Zhang et al., 2008). 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 30 times independently with each approach and the average results are used for comparison.

4.3. Comparisons on Fitness Quality

The results of GA, PSO, CLPSO, JADE, OLPSO, and OLPSO/PS are compared in Table 3 where the "Mean" stands for the average fitness value of the 30 independent runs and "Std. Dev" is the standard deviation. Moreover, the "Median" fitness value and the "Best" fitness value among the 30 runs are given and compared in the Table 3. It can be observed from the table that OLPSO/PS achieves better results than not only GA and PSO, but also OLPSO and the state-of-the-art CLPSO and JADE when measured by the mean fitness value. Therefore, OLPSO/PS has the capacity to obtain good solutions consistently. Moreover, OLPSO/PS can obtain the highest "Best" fitness solution among all the approaches, indicating the strongest global search ability of OLPSO/PS. By comparing the "Median" fitness, OLPSO/PS also performs the best among all the five approaches, indicating that OLPSO/PS can obtain high quality solution at most of the time.

It should be noticed that these results are obtained in the new large search space which is set freely according to commonly used ranges. The total failure of GA indicates that the GA approach is not efficient enough to make sufficient search in the large space to find good solution, even though it could be used for optimizing PEC with specific pre-defined search ranges (Zhang et al., 2001). Moreover, the large search space challenges the search ability of the traditional PSO algorithm, CLPSO, and JADE. The OLPSO/PS algorithm is still promising and its results are demonstrated to be much better than the others. The obtained component values in the "Best" fitness solution optimized by different approaches are presented in Table 4. The results show that the components values obtained by GA are much different from those obtained by OLPSO and OLPSO/PS. Although OLPSO and OLPSO/PS obtain similar values, the fitness still shows that OLPSO/PS performs better than OLPSO, indicating that the PS strategy makes OLPSO better in the PEC optimization problem.

4.4. Comparisons on Optimization Speed and Reliability

Besides the high solution quality of OLPSO/PS, the fast optimization speed and strong algorithm reliability of OLPSO/PS are also supported by the comparisons in Table 3. By giving an acceptable fitness value of 150, OLPSO/PS can

Table 4. Optimized component values in the best run with different approaches

Components	GA	PSO	CLPSO	JADE	OLPSO	OLPSO/PS
R_1	356.099 Ω	100 Ω	100 Ω	100 Ω	100 Ω	100 Ω
R_2	60.4418 kΩ	71.7442 kΩ	36.4732 kΩ	82.223 kΩ	13.1202 kΩ	12.3865 kΩ
R_{C3}	98.6189 kΩ	831.532 Ω	960.373 Ω	136.366 Ω	1.04713 kΩ	1.02 kΩ
R_4	2.07867 kΩ	11.4945 kΩ	115.497 Ω	100 Ω	11.1206 kΩ	104.208 Ω
C_2	19.6276 µF	0.1 µF	0.1 µF	0.1 µF	0.1 µF	0.1 µF
C_3	28.0941 µF	1.72671 µF	1.47557 µF	6.5245 µF	1.11032 µF	1.11977 µF
C_4	3.38356 µF	0.1 µF	9.85044 µF	14.1778 µF	0.1 µF	10.670 µF
Fitness	127.494	192.304	191.638	183.208	192.962	**192.992**

successfully obtain final solutions with fitness values larger than 150 in all of the 30 runs whilst OLPSO succeeds in 28 runs, CLPSO succeeds in 14 runs, PSO and JADE can only succeeds in 8 runs, and GA totally fails in obtaining solutions with fitness values larger than 150. Therefore, OLPSO/PS is the most reliable algorithm that can obtain high quality solutions to PEC constantly. Moreover, among the successful runs in each algorithm, the mean FEs needed to reach the acceptable value of 150 compared in Table 3 further shows that OLPSO/PS is the fastest algorithm among the six contenders. Therefore, the results confirm that the PS strategy not only makes OLPSO/PS more reliable, but also makes OLPSO/PS converge faster. The mean convergence characteristics of different approaches are plotted in Figure 4. The curves show that both GA and JADE fall into a poor local optimum quite early whilst OLPSO and OLPSO/PS can improve the fitness value steadily for a long time. More importantly, OLPSO/PS has faster

Figure 4. Mean convergence characteristics of different approaches

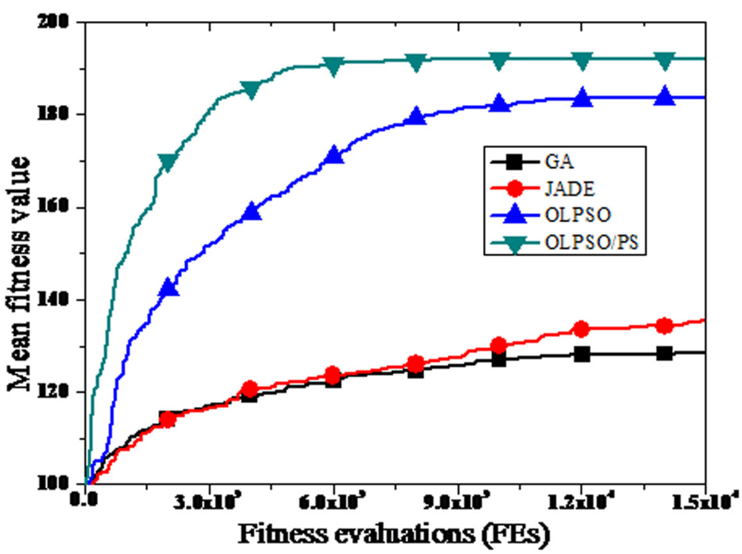
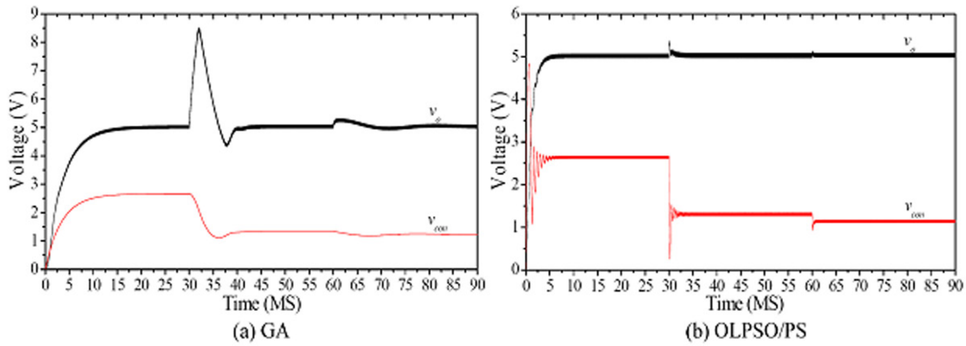


Figure 5. Simulated voltage responses from 0 ms to 90 ms. From 0 ms to 30 ms, v_{in} is 20 V and R_L is 5 Ω; on the 30 ms, v_{in} is suddenly changed from 20 V to 40 V while R_L is still 5 Ω; on the 60 ms, R_L is suddenly changed from 5 Ω to 10 Ω while v_{in} is still 40 V



speed than OLPSO to convergent to the global optimal region.

4.5. Comparisons on Simulation Results

Simulations are conducted in this sub-section. The component values of the PEC are set as the optimized results of GA and OLPSO/PS. In order to make the comparisons clearer, the median solution obtained by each approach is used. The simulation results are plotted and compared in Figure 5 and Figure 6. In the simulation results comparison, Figure 5 gives the results of voltage and Figure 6 gives the results of current.

The simulation lasts for 90 milliseconds (ms). The input voltage v_{in} is 20 V and the output load R_L is 5 Ω. The simulated startup transients can be compared in the first 30 ms of the figures. It is observed that the circuit with OLPSO/PS-optimized component values has better performance, giving faster settling time. The buck with component values optimized by OLPSO/PS uses only about 5 ms to reach the steady state, while the one with component values optimized by GA uses more than 10 ms.

Figure 6. Simulated current responses from 0 ms to 90 ms. From 0 ms to 30 ms, v_{in} is 20 V and R_L is 5 Ω; on the 30 ms, v_{in} is suddenly changed from 20 V to 40 V while R_L is still 5 Ω; on the 60 ms, R_L is suddenly changed from 5 Ω to 10 Ω while v_{in} is still 40 V

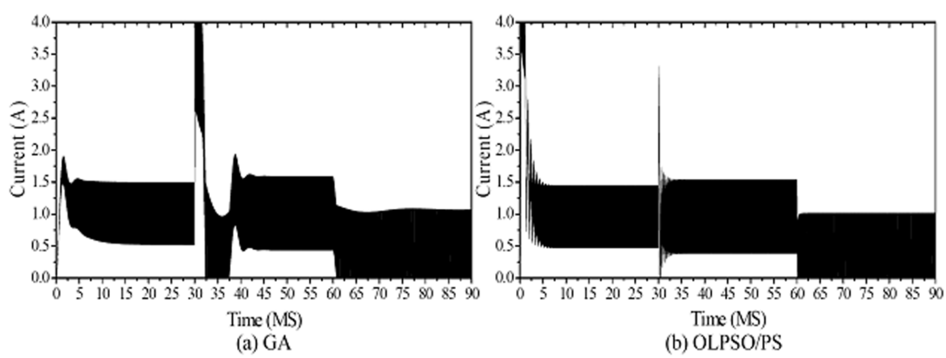
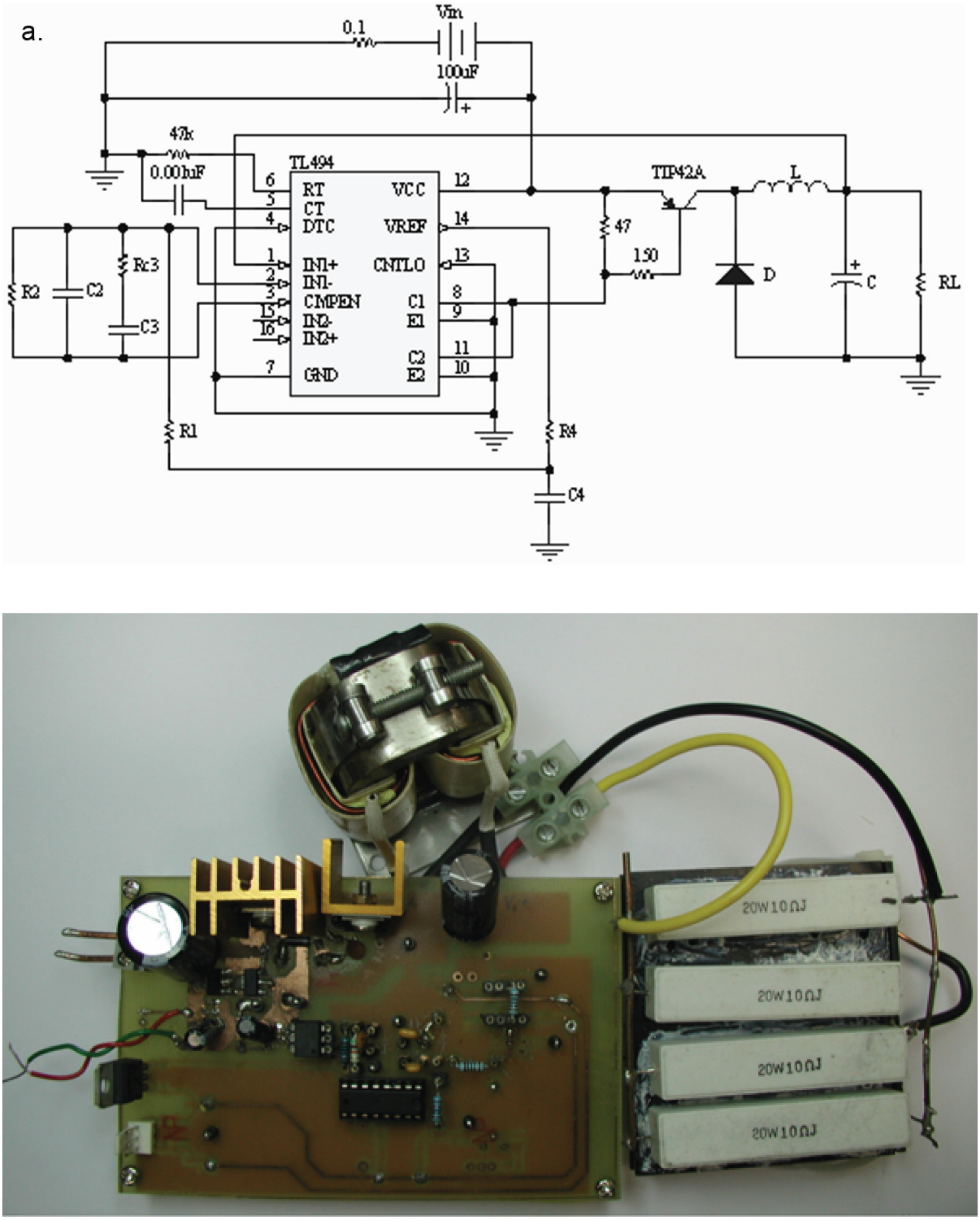


Figure 7. (a). The practical circuit with component values optimized by OLPSO/PS- Circuit diagram of the practical circuit. (b). The practical circuit with component values optimized by OLPSO/PS- Actual picture of the OLPSO-optimized circuit



b.

Figure 8. Experimental transient responses of v_o (500 mV/division) and i_L (1 A/division) with the values optimized by OLPSO (Timebase: 400 µs/division) when the input voltage v_{in} is suddenly changed from 20 V to 40 V

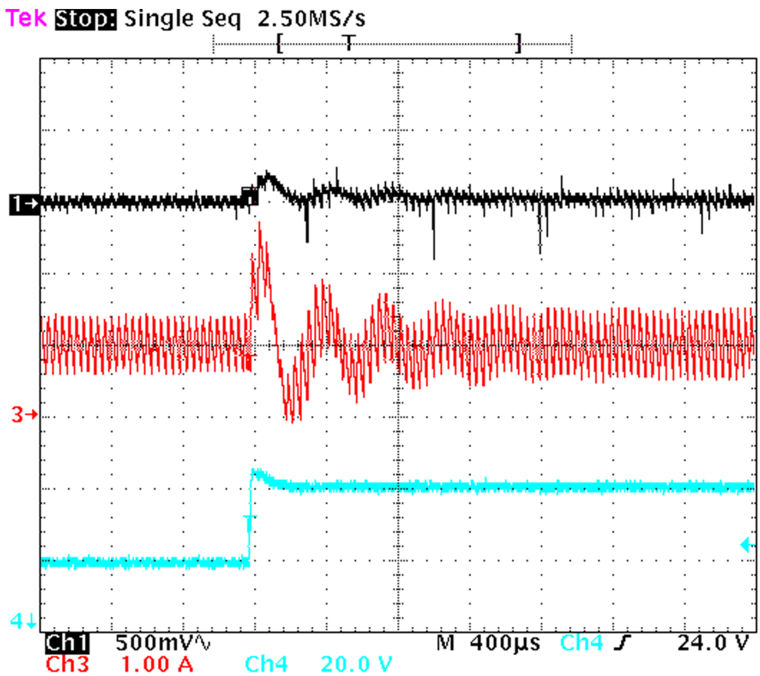


Figure 5 and Figure 6 also show the simulated transient responses under large signal disturbances. On the 30 ms, when the regulator is in steady state, the input voltage is suddenly changed from 20 V to 40 V, with the load still fixed as 5 Ω. As the responses to this change, the output voltage v_o, the control voltage v_{con}, and the inductor current i_L are all disturbed. However, the circuit optimized by OLPSO/PS has much smaller disturbance and shorter response time (about 2 ms) than the one optimized by GA (about 12 ms), confirming the advantages of the OLPSO/PS algorithm. Moreover, the output ripple voltage v_o in the OLPSO/PS optimized circuit is less than 1%, satisfying the required specification very well. However, for the GA optimized circuit, the ripple of the output voltage is very large.

Similar tests on load disturbances are also studied when the system has reverted a steady state with v_{in} equals 40 V and R_L equals 5 Ω. In this disturbance, R_L is suddenly changed from 5 Ω to 10 Ω on the 60 ms, with the v_{in} being still fixed as 40 V. The simulation results in the figures also show that the OLPSO/PS-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 OLPSO/PS algorithm can optimize the circuit component values and make the circuit exhibit better dynamic performance.

4.6. Validation on Practical Circuit

Experiments on the practical circuit as shown in Figure 7 are conducted in this sub-section to validate the performance of the OLPSO/PS-optimized circuit. The circuit is designed with the components optimized by OLPSO/PS, as the components values given in Table 4.

Figure 9. Experimental transient responses of v_o (200 mV/division) and i_L (1 A/division) with the values optimized by OLPSO (Timebase: 1 ms/division) when the output load R_L is suddenly changed from 10 Ω to 5 Ω

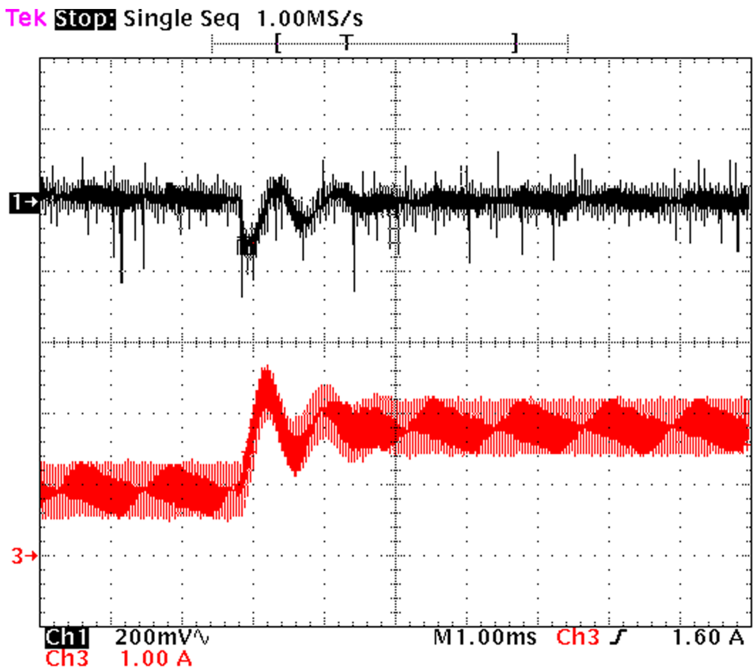


Figure 8 shows the experimental transient responses of v_o and i_L when the input voltage v_{in} is suddenly changed from 20 V to 40 V. The figure shows that the overshoot on the current i_L is smaller than 2 A and the overshoot on v_o is smaller than 200 mV. The OLPSO/PS-optimized circuit is demonstrated to yield good performance for that its output voltage v_o returns to the steady value of 5 V in about 1.2 ms settling time, which is much smaller than the maximal tolerant time in the design requirements.

Similar disturbance is also performed on the output load R_L, which is suddenly changed from 10 Ω to 5 Ω, and the experimental transient responses are shown in Figure 9. These experimental results also show that the circuit with component values optimized by OLPSO/PS gives very good performance, with the output voltage returning to 5 V in about only 2 ms. These experimental results indicate that the OLPSO/PS-optimized circuit not only meets the practical design requirements, but also yields better performance on the settling time than those in the literature (Zhang et al., 2001; Zhang et al., 2009).

5. CONCLUSION

This paper presented an OLPSO/PS algorithm for optimizing the component values in designing PEC. The challenge of the problem is that the components interact with each other and make the search space complex. Moreover, this paper argued that the search range of the components is not easy to be determined within certain specific range, and proposed to set a USR for the components according to the commonly used range in practical industrial community. To optimize PEC with such complex search space, the distinct feature of the proposed OLPSO/PS algorithm is that it uses a novel orthogonal learning strategy that can discover useful information in a particle's personally best position and its neighborhood's best position. This efficient OL strategy helps a particle construct a more promising and efficient guidance exemplar to improve its flying velocity and direction, and therefore is promising in a large search space like the PEC problem with USR. Moreover, the efficient PS strategy has been used in the OLPSO/PS algorithm to help increase diversity to the population and to help save the computational burden.

The effectiveness and efficiency of the OLPSO/PS algorithm in optimally designing PEC have been evaluated with the design of a buck regulator with overcurrent protection. In order to demonstrate the advantages of the proposed OLPSO/PS algorithm, results obtained by GA and by PSO using the traditional learning strategy were compared with the ones obtained by OLPSO/PS. Moreover, the OLPSO without using PS and the well-performed CLPSO and JADE algorithms were also used for comparisons. The results showed that OLPSO/PS outperforms the other algorithms not only with higher quality fitness value, but also with faster optimization speed, and stronger algorithm reliability. The comparisons with OLPSO further indicated that the PS strategy helps OLPSO/PS save lots of computational burden and also enhance the global search ability. Moreover, simulations on the circuits optimized by GA and OLPSO/PS were conducted and compared. The simulation results demonstrate the advantages of the OLPSO/PS algorithm by showing that the circuit optimized by OLPSO/PS exhibits both shorter startup time and shorter settling time in the transient responses, than the circuit optimized by GA. The disturbance of output voltage and current in the OLPSO/PS-optimized circuit is also slighter than those in the circuit optimized by GA. Further, the OLPSO/PS-optimized component values are applied to a practical circuit. The experimental results confirmed the effectiveness and efficiency of the OLPSO/PS-optimized circuit.

The good performance of OLPSO/PS on buck regulator optimization indicates that it is applicable and promising in solving the PEC or other controllers with complex structure. Therefore, it is interesting for the future work to apply the OLPSO/PS algorithm to optimize other problems in power electronics filed, and better performance may be expected. Moreover, we are currently extending the PEC problem to multi-objective optimization model. Therefore, our future work also includes applying multi-objective algorithms for PEC optimization.

The 'errFEs' is the mean fitness evaluations to reach an acceptable fitness of 150 among all the successful runs.

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

For code details of this paper, please contact the corresponding author Zhi-Hui Zhan (zhanzhh@mail.sysu.edu.cn).

ACKNOWLEDGMENT

This work was supported in part by the National High-Technology Research and Development Program (863 Program) of China No.2013AA01A212, in part by the National Natural Science Foundation of China (NSFC) with No. 61402545 and No. 61309003, the NSFC Key Program with No. 61332002, the Natural Science Foundation for Distinguished Young Scholars of Guangdong Province, China, and the Project for Pearl River New Star in Science and Technology, Guangzhou, China.

REFERENCES

Ebrahimi, J., Babaei, E., & Gharehpetian, G. B. (2012). A new multilevel converter topology with reduced number of power electronic components. *IEEE Transactions on Industrial Electronics*, *59*(2), 655–667. doi:10.1109/TIE.2011.2151813

Electronics. From http://www.electronics2000.co.uk/data/itemsmr/res_val.php, last assessed Feb. 2014.

Emadi, A. (2004). Modeling of power electronic loads in AC distribution systems using the generalized state-space averaging method. *IEEE Transactions on Industrial Electronics*, *51*(5), 992–1000. doi:10.1109/TIE.2004.834950

Ghosh, P., Das, S., & Zafar, H. (2012). Adaptive-differential-evolution-based design of two-channel quadrature mirror filter banks for sub-band coding and data transmission. *IEEE Transactions on Systems, Man and Cybernetics. Part C, Applications and Reviews*, *42*(6), 1613–1623. doi:10.1109/TSMCC.2012.2212007

Harjani, R., Rutenbar, R. A., & Carley, L. R. (1989). OASYS: A framework for analog circuit synthesis. *IEEE Transactions on Computer-Aided Design*, *8*(6), 1247–1266. doi:10.1109/43.44506

Hu, Y., Ding, Y., Hao, K., Ren, L., & Han, H. (2014). An immune orthogonal learning particle swarm optimisation algorithm for routing recovery of wireless sensor networks with mobile sink. *International Journal of Systems Science*, *45*(3), 337–350. doi:10.1080/00207721.2012.723053

Huelsman, L. P. (1993). Optimization—A powerful tool for analysis and design. *IEEE Transactions on Circuits Systems, I, 40*(7), 431–439.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In *Proc. IEEE Int. Conf. Neural Networks* (pp. 1942-1948).

Kennedy, J., Eberhart, R. C., & Shi, Y. H. (2001). *Swarm Intelligence*. San Mateo, CA: Morgan Kaufmann.

Kwok, N. M., Shi, H. Y., Ha, Q. P., Fang, G., Chen, S. Y., & Jia, X. (2013). Simultaneous image color correction and enhancement using particle swarm optimization. *Engineering Applications of Artificial Intelligence*, *26*(10), 2356–2371. doi:10.1016/j.engappai.2013.07.023

Li, Y. H., Zhan, Z. H., Lin, S., Zhang, J., & Luo, X. N. (2015). Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems. *Information Sciences*, *293*(1), 370–382. doi:10.1016/j.ins.2014.09.030

Li, Y. L., Zhan, Z. H., Gong, Y. J., Chen, W. N., Zhang, J., & Li, Y. (2014). *Differential evolution with an evolution path: A DEEP evolutionary algorithm*. IEEE Transactions on Cybernetics; doi:10.1109/TCYB.2014.2360752

Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, *10*(3), 281–295. doi:10.1109/TEVC.2005.857610

Lim, W. H., & Isa, N. (2014). Particle swarm optimization with increasing topology connectivity. *Engineering Applications of Artificial Intelligence*, *27*, 80–102. doi:10.1016/j.engappai.2013.09.011

Mazhoud, I., Hadj-Hamou, K., Bigeon, J., & Joyeux, P. (2013). Particle swarm optimization for solving engineering problems: A new constraint-handling mechanism. *Engineering Applications of Artificial Intelligence*, *26*(4), 1263–1273. doi:10.1016/j.engappai.2013.02.002

Ochotta, E. S., Rutenbar, R. A., & Carley, L. R. (1996). Synthesis of high-performance analog circuits in ASTRX/OBLX. *IEEE Transactions on Computer-Aided Design*, *15*(3), 273–294. doi:10.1109/43.489099

Shen, M., Zhan, Z. H., Chen, W. N., Gong, Y. J., Zhang, J., & Li, Y. (2014). Bi-velocity discrete particle swarm optimization and its application to multicast routing problem in communication networks. *IEEE Transactions on Industrial Electronics*, *61*(12), 7141–7151. doi:10.1109/TIE.2014.2314075

Song, Y., & Wang, B. (2013). Survey on reliability of power electronic systems. *IEEE Transactions on Power Electronics*, *28*(1), 591–604. doi:10.1109/TPEL.2012.2192503

Sussman, G. J., & Stallman, R. M. (1975). Heuristic techniques in computer aided circuit analysis. *IEEE Transactions on Circuits and Systems*, *CAS-22*(11), 857–865. doi:10.1109/TCS.1975.1083985

Zhan, Z. H., Li, J., Cao, J., Zhang, J., Chung, H., & Shi, Y. H. (2013). Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems. *IEEE Transactions on Cybernetics*, *43*(2), 445–463. doi:10.1109/TSMCB.2012.2209115

Zhan, Z. H., & Zhang, J. (2011). Orthogonal learning particle swarm optimization for power electronic circuit optimization with free search range. In Proc. IEEE Congr. Evol. Comput. (pp.2563-2570). doi:10.1109/CEC.2011.5949937

Zhan, Z. H., Zhang, J., Li, Y., & Chung, H. (2009). Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, B., 39*(6), 1362-1381.

Zhan, Z. H., Zhang, J., Li, Y., & Shi, Y. H. (2011). Orthogonal learning particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, *15*(6), 832–847. doi:10.1109/TEVC.2010.2052054

Zhang, J., Chung, H., Lo, W. L., Hui, S., & Wu, A. (2001). Implementation of a decoupled optimization technique for design of switching regulators using genetic algorithm. *IEEE Transactions on Power Electronics*, *16*(6), 752–763. doi:10.1109/63.974373

Zhang, J., Chung, S. H., Lo, W. L., & Huang, T. (2009). Extended ant colony optimization algorithm for power electronic circuit design. *IEEE Transactions on Power Electronics*, *24*(1), 147–162. doi:10.1109/TPEL.2008.2006175

Zhang, J., Shi, Y., & Zhan, Z. H. (2008). Power electronic circuits design: A particle swarm optimization approach. In *Proc. the 7th International Conference on Simulated Evolution and Learning* (pp. 605–614). doi:10.1007/978-3-540-89694-4_61

Zhang, J. Q., & Sanderson, A. C. (2009). JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, *13*(5), 945–958. doi:10.1109/TEVC.2009.2014613

Ying Lin received the Bachelor's degree and the Ph. D degree in 2007 and 2012, respectively, from the Department of Computer Science of Sun Yat-sen University, Guangzhou, China. She is currently a lecturer with the Department of Psychology, Sun Yat-sen University. She has published over 20 technical papers in her resarch areas. Her current research interests include genetic algorithms, differential evolution, tribal ecosystem inspired algorithm, ant colony optimization, particle swarm optimization, and their real-world applications, especially in the fields of engineering, psychological measurements, and social network analysis.

Zhi-Hui Zhan received the Bachelor's degree and the Ph. D degree in 2007 and 2013, respectively, from the Department of Computer Science of Sun Yat-Sen University, Guangzhou, China. He is currently a lecturer in Sun Yat-sen University. His current research interests include particle swarm optimization, differential evolution, brain strom optimization, ant colony optimization, genetic algorithms, and their applications in real-world problems. Dr. Zhan's doctoral dissertation was awarded the China Computer Federation (CCF) Outstanding Dissertation in 2013. He was a recipient of the Natural Science Funds for Distinguished Young Scientists from the Natural Science Foundation of Guangdong Province, China in 2014 and Pearl River New Star in Science and Technology award in 2014.

Jun Zhang received the Ph.D. degree from the City University of Hong Kong, Kowloon, Hong Kong, in 2002. He is currently a Changjiang Chair Professor with the School of Advanced Computing, Sun Yat-sen University. He has published over 100 technical papers in his research areas. His current research interests include computational intelligence, cloud computing, high performance computing, data mining, wireless sensor networks, operations research, and power electronic circuits. Dr. Zhang was a recipient of the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011 and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the IEEE Transactions on Evolutionary Computation, the IEEE Transactions on Industrial Electronics, and the IEEE Transactions on Cybernetics.