Particle Swarm Optimization As a Tool For Electrical Circuit Analysis

Majid Toghyani Rizi¹, Mohammad Hadi Shahrokh Abadi², Ahmad Hajipour³
¹,²,³ Faculty of Electrical and Computer Engineering, Hakim Sabzevari University, Zip Code: 9617976487, Sabzevar, Iran

¹ toghyani.rizi@gmail.com
² mhshahrokh@hsu.ac.ir
³ a.hajipour@hsu.ac.ir

Abstract:
Application of metaheuristic algorithms has been of continued interest in the field of electrical engineering because of their powerful features. In this paper we used Particle Swarm Optimization (PSO) as a tool for optimizing a special battery charger circuit with three unknown variables in order to transfer maximum delivered power of main battery to the load batteries as a case study. The MATLAB simulation results are completely compatible with PSPICE software.

Keywords: metaheuristic algorithms, Particle Swarm Optimization (PSO), battery charger circuit, MATLAB

1. Introduction
Nature-inspired optimization algorithms have become increasingly popular in recent years, and most of these metaheuristic algorithms, such as particle swarm optimization (PSO) and firefly algorithms, are often based on swarm intelligence [1].

particle swarm optimization is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position but is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions. PSO algorithm has been used in various fields of electrical engineering for the aim of optimizing such as design of logic circuits, analog and digital filter design, integrated circuit design, Microwave circuit design and etc. [2-8].

Modeling and simulation are very important tools of systems engineering. They help us gain a better understanding of the functioning of the real world. Also they are important for the design of new systems to predict their behavior under varying operating conditions before they actually have been built [9].

In this work, a battery charger circuit as an applicable electrical circuit has been modeled and simulated. Classical PSO algorithm is described in section 1 then its use as a tool for optimizing the circuit variables in order to transfer a determined part of master battery power to other batteries (load batteries) is investigated in section 2. The validation of algorithm
responses was examined by PSPICE (orcad) software in section 3. Finally, section 4 presents concluding remarks and suggestions for future work.

2. Classical PSO Algorithm

The basic principles in “classical” PSO are very simple. A set of moving particles (the swarm) is initially “thrown” inside the search space. Each particle has the following features [10-11]:
1. It has a position and a velocity.
2. It knows its position, and the objective function value for this position.
3. It remembers its best previous position found so far (pbest).
4. It knows its neighbor s’ best previous position and objective function value (gbest).

Each particle tries to modify its position and velocity according to following equations:

\[ V_i^{k+1} = w V_i^k + c_1 \text{ rand} \times (pbest_i - s_i^k) + c_2 \text{ rand} \times (gbest - s_i^k) \]  

(1)

\[ w = w_{\text{Max}} - \left(\frac{w_{\text{Max}} - w_{\text{Min}}}{\text{max iter}}\right) \times \frac{\text{iter}}{\text{max iter}} \]  

(2)

\[ s_i^{k+1} = s_i^k + V_i^{k+1} \]  

(3)

Where
- \( V_i^k \): velocity of agent i at iteration k
- \( w \): weighting function
- \( c_i \): weighting factor
- \( \text{rand} \): uniformly distributed random number between 0 and 1
- \( s_i^k \): current position of agent i at iteration k
- \( pbest_i \): pbest of agent i
- \( gbest \): gbest of the group
- \( w_{\text{Max}} \): initial weight
- \( w_{\text{Min}} \): final weight
- \( \text{max iter} \): maximum iteration number
- \( \text{iter} \): current iteration number

The flowchart for PSO is given in Figure 1. The main steps of PSO algorithm are as follows:

Step1: Initialize the particles with initial random positions in search space and the velocities of particles in a given range randomly and define these as the best known positions (Pbest) of each particle.

Step2: Define the objective function (fitness) that needs to be optimized and evaluate it.

Step3: if particle fitness (fitness(p)) is better than fitness(pbest) then pbest=p.

These two steps will continue till all the particles exhaust.

Step4: Select best of pbests as gbest.

Step5: update particles position and velocity according to equations (1) and (3) respectively.

Above steps will be continue until maximum iteration reaches.
3. A battery charging circuit as a case study

The electric circuit shown in Figure 2 uses a 30 V battery (main battery) to charge three batteries of values 5, 10, and 20V (load batteries). The currents in the branches have the shown directions [13-14].

All currents $I_i$, $\forall i$ must be positive to prevent batteries from discharging. Suppose that the circuit fixed parameters have the following values: $R1 = 33\Omega$, $R3 = 5\Omega$, $V1 = 30$ volt, $V2 = 5$ volt, $V3 = 10$ and $V4 = 20$ volt. The other circuit parameters are variables.

The range of values of those parameters are:
- $R2 : 100\Omega$ to $1k\Omega$
- $R4 : 0.5k\Omega$ to $1.5k\Omega$
- $R5 : 1\Omega$ to $10\Omega$

We aim at determining the values of the three variables ($R2$, $R4$ and $R5$) so that maximum percent of delivered power by 30V battery would be consumed by other three batteries. In this case study we want to transfer 0.62 percent of the total power of 30v battery to the charged batteries by considering the power of resistors. To solve this problem, we use PSO.

The solution space will be 3 dimensional that is often referred to as hyper space. The range of the values of the parameters defines the solution region. Each position of the particle is denoted by 3 co-ordinates. They represent $R2$, $R4$, and $R5$. To solve the problem, at first we need to determine the currents in the branches. We use Mesh current analysis. The mesh equations in matrix form are given by:

$$
\begin{bmatrix}
R_1 + R_2 & -R_2 & 0 \\
-R_2 & R_2 + R_3 + R_4 & -R_4 \\
0 & -R_4 & R_4 + R_5
\end{bmatrix}
\begin{bmatrix}
I_1 \\
I_3 \\
I_5
\end{bmatrix}
= \begin{bmatrix}
V_1 - V_2 \\
V_2 - V_3 \\
V_3 - V_4
\end{bmatrix}
$$

(4)
This equation is comparable to $A_i = V$. Using matrix inversion, we can solve $i$ (the current matrix $I(1), I(2), I(3)$). Using KCL, one can obtain other branches currents as follows:

$$
I_1 = I(1) \\
I_2 = I(2) \\
I_3 = I(3) \\
I_4 = I_1 - I_3 \\
I_5 = I_2 - I_5
$$ (5)

If we put values of all the parameters, the mesh current can be solved using simple computer code. From the currents we can calculate the power delivered by the 30 V voltage source ($P_T$) and the power that are consumed by the other batteries ($P_B$) and resistors ($P_R$):

$$
P_T = V_1 I_1 \\
P_R = \sum_{j=1}^{5} P_{R_j} \\
P_B = \sum_{j=2}^{4} P_{B_j} \\
P_{R_j} = R_j I_j^2 \\
P_{B_j} = V_j I_j
$$ (6)

Here $V_1$ is 30 V or master battery. Now, we need to define the fitness function. The function that we are trying to optimize is called the fitness function and the value of the fitness function corresponding to the position of a particle is known as the fitness value of that particle. We require that ($P_T$) and ($P_B$) have the relation: $0.62 P_T = P_B$. We define fitness function as: $Fitness = -abs(0.62 P_T - P_B)$. Here abs denotes absolute value. We note that when our condition is satisfied, the fitness function will have zero value. In other cases, we will a negative fitness value. So, maximizing this fitness function will give us the desired results.

3. Simulation Results of MATLAB

The values of variable parameters in the battery charger circuit i.e $R_2, R_4$ and $R_5$ and also the amounts of other parameters such as ($P_T$), ($P_B$), ($P_R$), the ratio of $\left(\frac{P_B}{P_T}\right)$ and $gbest_val$ are given in the table 1. Simulation results of MATLAB are illustrated in figure 3. It is important to note that the values of variable parameters are changed after each running of algorithm because of random selection of particles (resistance values).

<table>
<thead>
<tr>
<th>Parameters</th>
<th>First Run (100 iteration)</th>
<th>Second Run (100 iteration)</th>
<th>Third Run (100 iteration)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_2(\text{k}\Omega)$</td>
<td>0.95185</td>
<td>0.94495</td>
<td>0.98700</td>
</tr>
<tr>
<td>$R_4(\text{k}\Omega)$</td>
<td>1.2214</td>
<td>1.0713</td>
<td>1.3968</td>
</tr>
<tr>
<td>$R_5(\Omega)$</td>
<td>2.5752</td>
<td>1.4917</td>
<td>4.3080</td>
</tr>
<tr>
<td>$P_B(\text{W})$</td>
<td>4.6554</td>
<td>4.7701</td>
<td>4.4818</td>
</tr>
<tr>
<td>$P_R(\text{W})$</td>
<td>2.8533</td>
<td>2.9236</td>
<td>2.7469</td>
</tr>
<tr>
<td>$P_T(\text{W})$</td>
<td>7.5087</td>
<td>7.6938</td>
<td>7.2288</td>
</tr>
<tr>
<td>Ratio (percent)</td>
<td>62.0</td>
<td>62.0</td>
<td>62.0</td>
</tr>
<tr>
<td>$gbest_val$</td>
<td>-5.3021e-8</td>
<td>-2.3194e-8</td>
<td>-3.8130e-8</td>
</tr>
</tbody>
</table>
Figure 3. Simulation results after first run (100 iteration) a) R2  b) R4  c) R5  d) fitness

The validation of resulted simulation of MATLAB, is examined by PSPICE (orcad) software for the first run of algorithm. The battery charger circuit along with simulation result of branch currents are illustrated in figure 4:
The power that is consumed by V1 ($P_T$) is -7.5087 W, that means the power is delivered to the circuit is 7.5087 W. Also, the power that is consumed by other batteries V2, V3 and V4 is 4.6554 W. The consumed power by resistors is 2.8533 W and this result confirms that the total generated power in the circuit by master battery is equal to the total consumed power by load batteries and resistors i.e $P_T = P_B + P_R$. These results are completely compatible with MATLAB simulation results. One can investigate the validation of other runs by changing the circuit resistances.
4. Conclusion

In this work, a battery charger circuit design as a case study is done using Particle Swarm Optimization. The aim of problem was optimizing three unknown variables of circuit to transfer maximum percent of delivered power by the main battery to the other batteries. The results are so useful for a circuit designer to choose optimum circuit elements to achieve a special goal in design. As a further work, studies will be carried out for using this evolutionary algorithm to optimize the circuits in frequency domain by employing phasor or laplace transform methods.

5. References


3. Xuesong YAN, Chengyu HU, Hong YAO, Yuanyuan FAN, Qingzhong LIANG, Chao LIU, Electronic Circuit Optimization Design Algorithm based on Particle Swarm Optimization, 2013, pages:50-52


9. Devendra k. chaturvedi, modeling and simulation of systems using MATLAB and Simulink ,CRC press, 2010


Majid Toghyani Rizi, Mohammad Hadi Shahrokh Abadi, Ahmad Hajipour


12. www23.homepage.villanova.edu/.../PSO_meander-line.ppt, access on: 1 Jan 2015

13. Mohamed Bakr, Nonlinear Optimization in Electrical Engineering with Applications in MATLAB, Published by The Institution of Engineering and Technology, 2013.