VOL. 6, NO. 10, OCTOBER 2011 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

SOFT COMPUTING OPTIMIZATION TECHNIQUES FOR SOLAR PHOTOVOLTAIC ARRAYS

Ramaprabha R. and Mathur B. L.
Department of Electrical and Electronics Engineering, SSN College of Engineering, Chennai, Tamilnadu, India
E-Mail: ramaprabhasuresh@gmail.com

ABSTRACT

This paper presents the soft computing optimization techniques to address the Maximum Power Point Tracking (MPPT) of Solar Photovoltaic (SPV) array under partial shaded conditions. Partial shaded SPV modules produce several local maximum power points, which makes the tracking of the Global Maximum Power Point (GMPP) a difficult task. Most of conventional tracking methods fail to work properly under partial shaded conditions. Methods proposed by some authors track the GMPP with some limitations. In this paper, three different soft computing techniques like Genetic algorithm (GA), Differential evolution (DE) and Particle Swarm optimization (PSO) techniques have been applied for GMPP tracking. The performances of these techniques are compared in respect of their tracking time and accuracy.

Keywords: solar photovoltaic array, soft computing methods, optimization, global maximum power point tracking, DE, PSO.

INTRODUCTION

In recent years intelligence techniques have been used widely in the Maximum Power Point Tracking (MPPT) process of SPV systems. Especially under non uniform and partially shading conditions, where there is a difficulty to track true GMPP in the presence of local MPPs. (Chen *et al*., 2010). Therefore, for satisfactory results, all environmental conditions (especially instantaneous climate changes and partial shading) must be taken into account in the design process of MPPT. Artificial intelligence can produce appropriate solutions for these conditions. Artificial intelligence-based MPPT algorithms are the most advantageous systems in terms of electrical efficiency. In this paper three different optimization technique viz; Genetic Algorithm (GA), Differential Evolution (DE) and Particle Swarm Optimization (PSO) techniques have been used for GMPP tracking. These techniques are compared in respect of their tracking time and accuracy.

PROBLEM FORMULATION

Necessity of optimization algorithm

The standard one diode or 5 parameter model used to represent the SPV module is shown in Figure-1. The modeling and simulation of SPV array under partial shaded conditions discussed by Ramaprabha and Mathur, 2009; Patel and Agarwal, 2008a; Patel and Agarwal, 2008 is used in this paper. The partial shade has more impact on series connected modules. To avoid the stress on low illuminated cells, bypass diodes are connected in anti parallel with a module/group of cells. The introduction of bypass diodes introduces multiple peaks in P-V characteristics. The simulation of series connected SPV array characteristics under partial shaded condition with bypass diode is shown in Figure-2. The model is developed using MATLAB M-file. The detailed explanation of the effect of bypass diodes in the characteristics has been discussed by Ramaprabha and Mathur, 2009 and Silvestre *et al*., 2009. Figure-2 shows the electrical characteristics of SPV array with bypass diodes under partial shaded conditions.

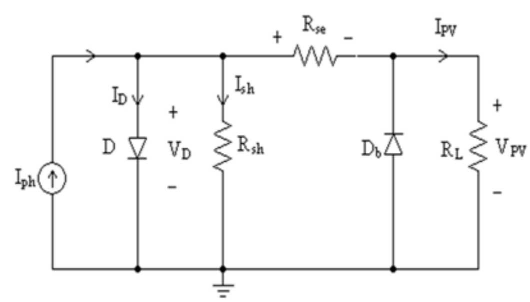

Figure-1. Five parameter model of SPV cell with bypass diode.

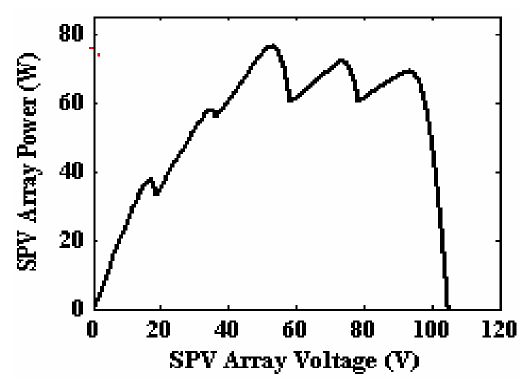

Figure-2. Characteristics of series connected SPV modules under partial shaded condition.

From Figure-2, it is observed that P-V characteristic has multiple peaks due to partial shading (Patel and Agarwal, 2008a). Among the multiple peaks one is GMPP and others are local peak power points. In this situation the conventional MPPT algorithm could fail to determine the actual GMPP or even traps into one of the local peaks. Therefore, considerable amount of possible

ARPN Journal of Engineering and Applied Sciences

SPV power is not utilized. Hence the power should be optimized to harvest the maximum power produced by SPV array.

Definition of objective function

A Nonlinear optimization problem can be stated in mathematical terms as given in equation (1).

$$\text{Find } Y = (y_1, y_2, \ldots.. y_n) \quad (1)$$

such that F(Y) is minimum or maximum subject to the constraint and bounds are given by equation (2).

$$g_j(Y) \geq 0,\ j = 1,2,..m \text{ and } y_j^L \leq y_i \leq y_j^U,\ j = 1,2...n \quad (2)$$

where

F is the objective function to be minimized or maximized, y_j's are variables, g_j is constraint function, y_j^L and y_j^U are the lower and upper bounds on the variables.

In this work the objective function considered is F(Y) = Maximization of SPVA power, P_{PV}

The variable y_i = SPVA current, I_{PV}. The constraint is $I_{PVmax} \geq I_{PV} \geq I_{PVmin}$. Here, $y_j^U = I_{PVmax.}$

Genetic algorithm

GA based optimization (Goldberg, 1989) is an adaptive heuristic search technique that involves generation, systematic evaluation and enhancement of potential design solution until a stopping criterion is met. There are three fundamental operators involved in the search process of a genetic algorithm: selection, crossover and mutation. Selection is a process which chooses a chromosome from the current generation's population for inclusion in the next generation's population according to their fitness. Crossover operator combines two chromosomes to produce a new chromosome (offspring). Mutation operator maintains genetic diversity from one generation of population to the next and aims to achieve some stochastic variability of GA in order to get a quicker convergence.

Differential evolution algorithm

DE algorithm is a population based algorithm like genetic algorithms using crossover, mutation and selection operators. DE uses the differences of randomly sampled pairs of object vectors to guide the mutation operation instead of using the probability distribution function as other evolutionary algorithms (Price at al., 2005). DE based optimization process is described below:

A. Initialization

DE starts with a population of M_P M-dimensional search variable vectors. The ith vector of the population at the current generation is given by

$$\vec{Y_i}(t) = [y_{i,1}(t), y_{i,2}(t), y_{i,3}(t) \ldots\ldots.. y_{i,M}(t)] \quad (3)$$

There is a feasible numerical range for each search-variable, within which value of the parameter should lie for better search results. Initially the problem parameters or independent variables are initialized in their feasible numerical range. If the jth parameter of the given problem has its lower and upper bound as y_j^L and y_j^U respectively, then the jth component of the ith population members is initialized as given by equation (4).

$$y_{i,j}(0) = y_j^L + \text{rand}(0,1) \cdot (y_j^U - y_j^L) \quad (4)$$

B. Mutation

In each iteration, to change the population member $\vec{Y_i}(t)$, a Donor vector $\vec{V}_i(t)$ is created. To create $\vec{V}_i(t)$ for each ith member, three other parameter vectors (r_1, r_2, r_3 vectors) are selected in random fashion from the current population. A scalar number F scales the difference of any two of the three vectors and the scaled difference is added to the third one to obtain the donor vector $\vec{V}_i(t)$. The mutation process for jth component of each vector is expressed by equation (5).

$$v_{i,j}(t+1) = y_{r1,j}(t) + F \cdot (y_{r2,j}(t) - y_{r3,j}(t)) \quad (5)$$

The method of creating donor vector demarcates between various DE schemes. Price and storn (2005) have suggested ten different mutation strategies. The above mutation strategy is referred as DE/rand/1. This scheme uses a randomly selected vector $\vec{Y}_{r1}$ and only one weighted difference vector $F \cdot (\vec{Y}_{r2} - \vec{Y}_{r3})$ is used to perturb it. In this work mutation strategy DE/best/1 is used. In this scheme the vector to be perturbed is the best vector of the current population and the perturbation is caused by single difference vector as given by equation (6).

$$v_{i,j}(t+1) = y_{best}(t) + F \cdot (y_{r1,j}(t) - y_{r2,j}(t)) \quad (6)$$

C. Crossover

To increase the potential diversity of the population a crossover operator is used. DE uses two kinds of cross over schemes namely "Exponential" and "Binomial". In this work binomial crossover is used. In this crossover scheme, the crossover is performed on each of the Q variables whenever a randomly picked number between 0 and 1 is within the crossover (CR) value. The scheme may be outlined as given by equation (7).

$$u_{i,j}(t) = v_{i,j}(t) \quad \text{if}(\text{rand}(0,1)) < CR \quad (7)$$

$$= y_{i,j}(t) \quad \text{else}$$

In this way for each trial vector $\vec{Y}_i(t)$ an offspring vector $\vec{U}_i(t)$ is created.

D. Selection

Selection operator is used to determine which one of the target vector and the trial vector will survive in the next generation. DE involves the Darwinian principle of "Survival of the fittest" in its selection process. The

VOL. 6, NO. 10, OCTOBER 2011 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

selection process may be outlined as given by equation (8).

$$\vec{Y}_i(t+1) = \vec{U}_i(t) \text{ if } f(\vec{U}_i(t)) \le f(\vec{Y}_i(t) \qquad (8)$$

$$= \vec{Y}_i(t) \text{ if } f(\vec{Y}_i(t) < f(\vec{U}_i(t))$$

where f is the function to be minimized. If the new trial vector yields a better value of the fitness function, it replaces its target in the next generation; otherwise the target vector is retained in the population.

Particle swarm optimization (PSO)

PSO is developed by Kennedy and Eberhart (1995). It was found to be reliable in solving non-linear problems with multiple optima. In PSO, a number of particles form a ''swarm" that evolve or fly throughout the feasible hyperspace to search for fruitful regions in which optimal solution may exist. Each particle has two vectors associated with it, the position (Z_i) and velocity (V_i) vectors. In N-dimensional search space, $Z_i = [z_{i1}, z_{i2} \ldots z_{iN}]$ and $V_i = [v_{i1}, v_{i2}, \ldots v_{iN}]$ are the two vectors associated with each particle i. During their search, members of the swarm interact with each others in a certain way to optimize their search experience. There are different variants of particle swarm paradigms but the most commonly used one is the gbest model where the whole population is considered as a single neighborhood throughout the flying experience (Kennedy and Eberhart 1995). In each iteration, particle with the best solution shares its position coordinates (gbest) information with the rest of the swarm. Each particle updates its coordinates based on its own best search experience (pbest) and gbest according to the equations (9) and (10).

$$v_i^{k+1} = wv_i^k + c_1 rand_1\left(pbest_i^k - z_i^k\right) + c_2 rand_2\left(gbest_i^k - z_i^k\right) \qquad (9)$$

$$z_i^{k+1} = z_i^k + v_i^{k+1} \qquad (10)$$

where c_1 and c_2 are two positive acceleration constants, they keep balance between the particle's individual and social behavior when they are set equal; $rand_1$ and $rand_2$ are two randomly generated numbers with a range of [0, 1] added in the model to introduce stochastic nature in particle's movement; and w is the inertia weight (Equation 11) and it keeps a balance between exploration and exploitation. In our case, it is a linearly decreasing function of the iteration index.

$$w(k) = w_{max} - \left(\frac{w_{max} - w_{min}}{iter_{max}}\right) \times iter \qquad (11)$$

where $iter_{max}$ is the maximum number of iteration, 'iter' is the current iteration number, w_{max} is the initial weight and w_{min} is the final weight. In conclusion, an initial value of *w* around 1, with a gradual decline toward 0 is considered as a proper choice. The most important factor that governs the PSO performance in its search for optimal solution is to maintain a balance between exploration and exploitation. Exploration is the PSO ability to cover and explore different areas in the feasible search space while exploitation is the ability to concentrate only on promising areas in the search space and to enhance the quality of potential solution in the fruitful region. Exploration requires bigger step sizes at the beginning of the optimization process to determine the most promising areas then the step size is reduced to focus only on that area. This balanced is usually achieved through proper tuning of PSO key parameters (Chaturvedi *et al*., 2009). Just like in the case of other evolutionary algorithms, PSO has many key features that attracted many researchers to employ it in different applications in which conventional optimization algorithms might fail such as:

- It only requires a fitness function to measure the ''quality" of a solution instead of complex mathematical operations like gradient, Hessian, or matrix inversion. This reduces the computational complexity and relieves some of the restrictions that are usually imposed on the objective function.
- It is less sensitive to a good initial solution since it is a population based method.
- It can be easily incorporated with other optimization tools to form hybrid ones.
- It has the ability to escape local minima since it follows probabilistic transition rules.

More interesting PSO advantages can be emphasized when compared to other members of evolutionary algorithms like:

- It can be easily programmed and modified with basic mathematical and logic operations. It is inexpensive in terms of computation time and memory.
- It requires less parameter tuning.
- It works with direct real valued numbers that eliminates the need to do binary conversion of classical canonical genetic algorithm.

SIMULATION AND COMPARISON OF ALGORITHMS TO ADDRESS GMPPT

The genetic algorithm implementation steps are given below:

Step 1: Read number of modules connected, insolation pattern and temperature for each module.
Step 2: Define objective function (Equation 1) and identify the parameters.
Step 3: Generate initial population.
Step 4: Evaluate the population by objective function.
Step 5: Test convergence. If satisfied then stop else continue.
Step 6: Start reproduction process by applying genetic operators: Selection, Crossover and Mutation.
Step 7: Evolve new generation. Go to step 3.

The DE algorithm implementation process is given below:

Step 1: Read number of modules connected, insolation pattern and temperature for each module.

Step 2: Initialize DE parameters like M, CR, M_P, F and Gen_{max}.
Step 3: Randomly generate initial population.
Step 4: Evaluate the population by objective function (equation-1) and determine best fit vector.
Step 5: For every vector in the population find the vector difference of two randomly selected vectors and mutate with the best vector of the current population to obtain donor vector using equation (6).
Step 6: Obtain the trial vector based on preset crossover constant using equation (7).
Step 7: For the entire population, evaluate the objective function value of trial vector and create a new population by selecting the target or trial vector based on the value of objective function.
Step 8: Test convergence. If satisfied then stop else go to step3.

The PSO algorithm implementation (Miyatake *et al*., 2007; Azab, 2009) process is given below:

Step 1: Read number of modules connected, insolation pattern and temperature for each module.
Step 2: Initialize PSO parameters such as w_{max}, w_{min}, c_1, c_2 and $Iter_{max}$.
Step 3: Generate initial population of N particles (design variables) with random positions and velocities.
Step 4: Compute objective value, current and power.
Step 5: Measure the fitness of each particle.
Step 6: Update personal best: Compare the fitness value of each particle with its pbests. If the current value is better than pbest, then set pbest value to the current value.
Step 7: Update global best: Compare the fitness value of each particle with gbest. If the current value is better than gbest, set gbest to the current particle's value.
Step 8: Update velocities: Calculate velocities V^{k+1} using equation (9).
Step 9: Update positions: Calculate positions Z^{k+1} using equation (10).
Step10: Return to step 4 until the current iteration reaches the maximum iteration number.
Step11: Output the optimal value of SPVA current and corresponding SPVA power in the last iteration.

All the algorithms have been written in M-file coding. The parameter settings for different algorithms are shown from Tables 1 to 3.

Table-1. GA Parameters.

Number of design variables	**1**
Population size, M_P	20
Crossover rate, CR	0.8
Mutation rate, F	0.10
Maximum generations, Gen_{max}	50
Selection scheme	Roulette wheel
Crossover	Two point
Mutation	Uniform

Table-2. DE Parameters.

Number of design variables	**1**
Population size M_P	20
Crossover constant, CR	0.8
Scaling factor for mutation, F	0.10
Maximum Generations, Gen_{max}	50

Table-3. PSO Parameters.

Number of design variables		**1**
Number of particles		20
Acceleration constants		$c_1 = 1.5$
		$c_2 = 1.5$
Inertia weight		$w_{max} = 0.9$
		$w_{min} = 0.4$
Maximum iterations, $Iter_{max}$		50

The performance of the optimization technique in terms of convergence with GA, PSO and DE is shown in Figure-3. From Figure-3, it is clear that PSO method converges earlier than the GA and DE. In order to verify the robustness of the algorithms, simulations were carried out for 30 independent runs (Yin *et al*., 2010). From the results in Table-4 it is evident that the PSO method is more robust than the GA and DE as the standard deviation of the fitness values for 20 runs is very low in the PSO method.

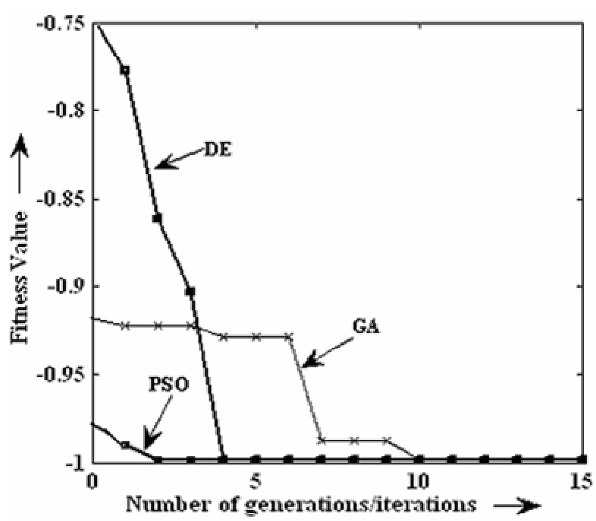

Figure-3. Convergence characteristics of GA, DE and PSO based methods.

VOL. 6, NO. 10, OCTOBER 2011 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Table-4. Comparison of different algorithms for a particular irradiance pattern (30 independent runs for a particular irradiance pattern $G_1 = 1000 W/m^2$; $G_2 = 600 W/m^2$; $G_3 = 300 W/m^2$).

Algorithm	**GA**	**DE**	**PSO**
Best solution	-0.9987	-0.9987	-0.9988
Worst solution	-0.9223	-0.7778	-0.9901
Average value	-0.9993	-0.9897	-0.9986
Standard deviation	0.0239	0.0385	0.0012

A comparison among different algorithms like Fibonacci search method, binary serach method, GA, DE and PSO for different set of irradiance levels is presented in Table-5.

Table-5. Comparison of different algorithms for different set of irradiation levels.

Shading pattern				Para-meters	Methods for GMPP tracking				
Label	G_1 (W/m^2)	G_2 (W/m^2)	G_3 (W/m^2)		Binary Search	Fibonacci Search	GA	DE	PSO
A	1000	1000	1000	P_m	111.3	114.8	112.42	111.8	111.27
				I_m	2.26	2.24	2.26	2.26	2.26
				V_m	49.23	51.25	49.74	49.47	49.23
B	1000	1000	800	P_m	97.54	96.69	98.83	97.83	97.54
				I_m	1.91	1.9	1.9	1.9	1.91
				V_m	51.07	50.8	52.02	51.49	51.34
C	1000	800	800	P_m	92.78	93.8	94.43	93.43	92.8
				I_m	1.85	1.8	1.85	1.85	1.85
				V_m	50.15	52.1	51.04	50.50	50.2
D	1000	1000	200	P_m	74.17	75.26	74.95	74.5	74.2
				I_m	2.26	2.3	2.26	2.26	2.26
				V_m	32.82	32.72	33.16	32.96	32.84
E	1000	500	500	P_m	59.4	60.23	61.08	61.01	59.36
				I_m	1.16	1.15	1.17	1.17	1.16
				V_m	51.21	52.37	52.21	52.14	51.17
F	1000	200	200	P_m	37.08	36.37	37.47	37.17	37.11
				I_m	2.26	2.28	2.26	2.26	2.26
				V_m	16.41	15.95	16.58	16.44	16.42
G	1000	800	600	P_m	75.2	76.1	76.64	75.84	75.23
				I_m	1.44	1.42	1.44	1.44	1.44
				V_m	52.22	53.6	53.22	52.6	52.24
H	1000	500	200	P_m	41.22	41.02	42.13	41.93	41.21
				I_m	1.18	1.16	1.18	1.18	1.18
				V_m	34.93	35	35.7	35.3	34.92
I	1000	600	300	P_m	48.95	49.01	49.94	49.04	48.93
				I_m	1.42	1.4	1.42	1.42	1.42
				V_m	34.48	34.96	35.17	34.54	34.46

It ensures the effectivess of PSO as compared with other algorithms. The performance of the PSO is validated graphically (Figure-4) by comparing its output (marked in green color) with that of the binary search method (marked in red color). In all the cases, PSO gives the optimum power (global peak) which is matched with the result of binary search. In PSO initially, the particles are randomly initialized. Therefore, the initial power is always high. This initial power corresponds to the 0^{th} iteration. As the algorithm progresses, the convergence is drastic and it finds a global maxima very quickly. The number of iterations needed for the convergence is seen to be 5-10, for this application environment.

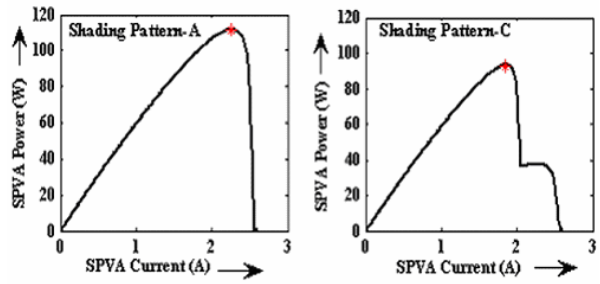

Figure-4. Validation of PSO for GMPP tracking of partial shaded SPVA.

CONCLUSIONS

This paper describes the optimization procedure to find GMPP of partial shaded SPVA using three different optimization techniques with the objective of maximizing the power. From the results it is seen that in terms of global exploration DE and PSO outperform GA. The results show that the convergence characteristics of PSO algorithm are better as compared to DE and GA. The PSO method has been found to be more robust as it gives minimum standard deviation than the other methods. The results show PSO algorithm is superior in terms of solution quality, global exploration and statistical soundness.

REFERENCES

Chaturvedi K.T., Pandit M. and Srivastava L. 2009. Particle Swarm Optimization with Time Varying Acceleration Coefficients for Non- Convex Economic Power Dispatch. International Journal of Electrical Power and Energy Systems. 31(6): 249-257.

Chen L.R, Tsai C.H., Lin Y.L. and Lai Y.S. 2010. A biological swarm chasing algorithm for tracking the PV maximum power point. IEEE Transactions on Energy Conversion. 25(2): 484-493.

Goldberg D. E. 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, Massachusetts, Harlow, England.

Kennedy J. and Eberhart R. 1995. Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks (ICNN'95). 4: 1942-1948.

Miyatake M., Inada T., Hiratsuka I., Zhao H., Otsuka H. and Nakano M. 2004. Control characteristics of a Fibonacci-search-based maximum power point tracker when a photovoltaic array is partially shaded. Proceedings of the 4th International Conference on Power Electronics and Motion Control. 2: 816-821.

Patel H. and Agarwal V. 2008a. MATLAB-Based Modeling to Study the Effects of Partial Shading on PV

Array. IEEE Transactions on Energy Conversion. 23(1): 302-310.

Patel H. and Agarwal V. 2008b. Maximum Power Point Tracking Scheme for PV Systems Operating Under Partially Shaded Conditions. IEEE Transactions on Industrial Electronics. 55(4): 1689-1698.

Price K., Storn R. and Lampinen J. 2005. Differential Evolution - A Practical Approach to Global Optimization. Springer, Berlin Heidelberg, New York, USA.

Ramaprabha R. and Mathur B.L. 2009. MATLAB based Modelling to Study the Influence of Shading on Series Connected SPVA. 2nd International Conference on Emerging Trends in Engineering and Technology, ICETET-09. pp. 30-34, December.

Ramaprabha R. and Mathur B.L. 2008. Modelling and Simulation of Solar PV Array under Partial Shaded Conditions. ICSET-2008. 7-11. Retrieved on May 6, 2009 from IEEE explore.

Silvestre S., Boronat A. and Chouder. A. 2009. Study of bypass diodes configuration on PV modules. Applied Energy 86. pp. 1632-1640.

Yin J. J., Tang W. and Man K. F. 2010. A Comparison of Optimization Algorithms for Biological Neural Network Identification. IEEE Transactions on Industrial Electronics. 57(3): 1127 -1131.

http://www.mathworks.com.