Lecture Series on

Intelligent Control

Lecture 29
Artificial Bee Colony Algorithm

Kwang Y. Lee
Professor of Electrical and Computer Engineering
Baylor University
Waco, TX 76798, USA
Kwang_Y_Lee@baylor.edu

1

Artificial Bee Colony Algorithm

- Artificial bee colony (ABC) algorithm is an optimization technique that simulates the foraging behavior of honey bees, and has been successfully applied to various practical problems.
- ABC belongs to the group of swarm intelligence algorithms and was proposed by Karaboga in 2005.

- Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
- Karaboga, D. & Basturk, B. On the performance of artificial bee colony (ABC) algorithm, Applied Soft Computing, volume 8, Issue 1, January 2008, pp 687–697.
- ABC Algorithm Homepage: https://abc.erciyes.edu.tr/

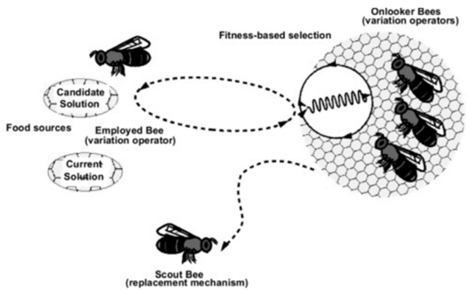
2

Artificial Bee Colony Algorithm

- A set of honey bees, called swarm, can successfully accomplish tasks through social cooperation. In the ABC algorithm, there are three types of bees: employed bees, onlooker bees, and scout bees.
- The employed bees search food around the food source in their memory; meanwhile they share the information of these food sources to the onlooker bees.
- The onlooker bees tend to select good food sources from those found by the employed bees. The food source that has higher quality (fitness) will have a large chance to be selected by the onlooker bees than the one of lower quality.
- The scout bees are translated from a few employed bees, which abandon their food sources and search new ones.

3

Artificial Bee Colony Algorithm



4

Artificial Bee Colony Algorithm

- In the ABC algorithm, the first half of the swarm consists of employed bees, and the second half constitutes the onlooker bees.
- The number of employed bees or the onlooker bees is equal to the number of solutions in the swarm. The ABC generates a randomly distributed initial population of *SN* solutions (food sources), where *SN* denotes the swarm size.

5

Artificial Bee Colony Algorithm

- Let $X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,n}\}$ represent the *i*-th solution in the swarm, where *n* is the dimension size.
- Each employed bee X_i generates a new candidate solution V_i in the neighborhood of its present position as equation below:

$$V_{i_k} = X_{i_k} + \Phi_{i_k} \times (X_{i_k} - X_{j_k})$$

- Where X_j is a randomly selected candidate solution, $i \neq j$,
- *k* is a random dimension index selected from the set $\{1, 2, \dots, n\}$
- Φ_{i_k} is a random number within $[-1, 1]$

6

Artificial Bee Colony Algorithm

$X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,n}\}$ $V_{i_k} = X_{i_k} + \Phi_{i_k} \times (X_{i_k} - X_{j_k})$

- Once the new candidate solution V_i is generated, a greedy selection is used. If the fitness value of V_i is better than that of its parents X_i, then update X_i with V_i; otherwise keep X_i unchanged.
- After all employed bees complete the search process; they share the information of their food sources with the onlooker bees through waggle dances.

7

Artificial Bee Colony Algorithm

- An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount.
- This probabilistic selection is really a roulette wheel selection mechanism which is described as equation below:

$$P_i = \frac{fit_i}{\sum_j fit_j}$$

- Where fit_i is the fitness value of the *i*-th solution in the swarm.
- As seen, the better the solution *i*, the higher the probability of the *i*-th food source selected.

8

Artificial Bee Colony Algorithm

- If a position cannot be improved over a predefined number (called limit) of cycles, then the food source is abandoned. Assume that the abandoned source is X_i, and then the scout bee discovers a new food source to be replaced with as equation below:

$$X_{i_k} = lb_j + rand(0,1) \times (ub_j - lb_j)$$

- Where $rand(0,1)$ is a random number within $[0,1]$ based on a normal distribution and lb, ub are lower and upper boundaries of the *j*-th dimension, respectively.

9

Artificial Bee Colony Algorithm

The ***main steps*** of the algorithm:[1]

- Initial food sources are produced for all employed bees
- REPEAT
  - Each employed bee goes to a food source in her memory and determines a neighbour source, then evaluates its nectar amount and dances in the hive
  - Each onlooker watches the dance of employed bees and chooses one of their sources depending on the dances, and then goes to that source. After choosing a neighbour around that, she evaluates its nectar amount.
  - Abandoned food sources are determined and are replaced with the new food sources discovered by scouts.
  - The best food source found so far is registered.
- UNTIL (requirements are met)
- https://www.youtube.com/watch?v=3qQr1eZwz5E

10

Improved Artificial Bee Colony Based on Orthogonal Learning for Optimal Power Flow

Wenlei Bai+, Ibrahim Eke++, Kwang Y. Lee+
+ Baylor University, U.S.A.
++ Kirikkale University, Turkey

ISAP 2015, Proto, Portugal, August 13-17, 2015

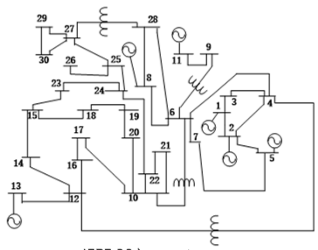
11

Outlines

- Optimal Power Flow (OPF)
- Traditional Artificial Bee Colony (ABC) Algorithm
- Improved ABC Based on Orthogonal Learning
- Simulation Results

12

What is Optimal Power Flow?



IEEE 30 bus system

Minimize total generating cost: $\min_{u} \sum_{i=1}^{N_G} C_i(P_i)$

13

BAYLOR UNIVERSITY — SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

- Optimization problem

$$\min_{u} f(u)$$
$$G(x,u,y)=0$$
$$H(x,u,y)\geq 0$$

- Equality constraints
  - Power balance at each node - power flow equations
- Inequality constraints
  - Network operating limits (line flows, voltages)
  - Limits on control variables

14

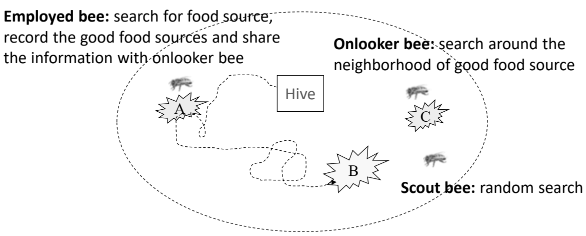
- Size of the problem
  - 1000's of lines, hundreds of controls
- Problem is non-linear
- Problem is non-convex
- Some of the variables are discrete
  - Position of transformer and phase shifter taps
  - Status of switched capacitors or reactors

Difficult to handle

15

Artificial Bee Colony (ABC)

- Population-based searching algorithm



16

- Each initial solution $X_i = \{X_{i,1}, X_{i,2}, \ldots, X_{i,D}\}$ is generated randomly and D is dimension of the problem, given in a vector form:

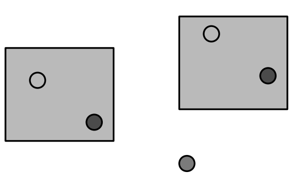
$$X_{i,j} = X_{\min,j} + \varphi(X_{\max,j} - X_{\min,j}) \quad i \in [1, SN] \quad j \in [1, D] \tag{1}$$

- After initialization, a candidate solution vector V_i is updated:

$$V_{i,j} = X_{i,j} + \Phi(X_{k,j} - X_{i,j}) \quad k \neq i \in [1, SN] \quad j \in [1, D] \tag{2}$$

- Where φ is a random number in (0,1), Φ is a random number in [-1,1] and SN is the size of onlooker or employed bees. Equation (2) means that the solution is updated around its neighborhood.

17



- The Employed Bee:
  It stays on a food source (feasible solutions) and provides the neighborhood of the source in its memory.
- The Onlooker Bee:
  It gets the information of food sources (feasible solutions) from the employed bees in the hive, select one of the food sources (high fitness values of solutions) and search better solutions around that food source.
- The Scout Bee:
  It is responsible for finding new food, the new nectar, sources.

18

Step 1) Initialization:
1.1) Randomly generate *SN* points in the search space as feasible solution X_i by (1).
1.2) Run Load Flow and evaluate the fitness function.

Step 2) For all employed bees (i = 1,…,*SN*):
2.1) Generate a candidate solution V_i by (2).
2.2) Run Load Flow and evaluate the fitness function.
2.3) Choose a solution (from X_i and V_i) with better fitness value.

Step 3) For all onlooker bees (will only be executed under certain probability p):
3.1) Generate a new candidate solution by V_i (2).
3.2) Run Load flow and evaluate the fitness function.
3.3) Choose a solution (from X_i and V_i) with better fitness value.

Step 4) For all scout bees (will be executed only after maximum trails):
Replace X_i with a new randomly produced solution X_i by (1).

19

Improved ABC

- The original ABC has poor efficiency on exploitation
- To overcome these issues, the *Orthogonal Learning* strategy (OL) is proposed to find an efficient candidate solution.
- *Orthogonal Array*

20

- An $N \times k$ array A is defined as an **orthogonal array** (OA) which has strength t on $0 \le t \le k$ when each $N \times t$ sub-array of A contains t-tuple exactly the same times as a row

$$L_9(3^3) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 3 & 3 \\ 2 & 1 & 2 \\ 2 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 3 \\ 3 & 2 & 1 \\ 3 & 3 & 2 \end{bmatrix}$$

- tuples (1,1) (1,2) (1,3) (2,1) (2,2) (2,3) (3,1) (3,2) (3,3) appear in any two columns one time

21

Orthogonal Experiment Design (OED)

OED: Orthogonal array is used to obtain the best candidate solution with fewer combinations

experiment with 3 levels and 3 factors:

Chemical Reaction Experiment

Factors / Levels	A Temp. °C	B Time (Min)	C Water (%)
1 L1	80	90	5
2 L2	85	120	6
3 L3	90	150	7

22

Deciding the Best Combination Levels by OED

$$L_9(3^3) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 3 & 3 \\ 2 & 1 & 2 \\ 2 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 3 \\ 3 & 2 & 1 \\ 3 & 3 & 2 \end{bmatrix}$$

Comb.	A: Temp. (°C)	B: Time (min)	C: Water (%)	Results
Cb1	(1) 80	(1) 90	(1) 5	$f_1 = 31$
Cb2	(1) 80	(2) 120	(2) 6	$f_2 = 54$
Cb3	(1) 80	(3) 150	(3) 7	$f_3 = 38$
Cb4	(2) 85	(1) 90	(2) 6	$f_4 = 53$
Cb5	(2) 85	(2) 120	(3) 7	$f_5 = 49$
Cb6	(2) 85	(3) 150	(1) 5	$f_6 = 42$
Cb7	(3) 90	(1) 90	(3) 7	$f_7 = 57$
Cb8	(3) 90	(2) 120	(1) 5	$f_8 = 62$
Cb9	(3) 90	(3) 150	(2) 6	$f_9 = 64$
levels	Factor Analysis			
L1	$(f_1+f_2+f_3)/3=41$	$(f_1+f_4+f_7)/3=47$	$(f_1+f_6+f_8)/3=45$	
L2	$(f_4+f_5+f_6)/3=48$	$(f_2+f_5+f_8)/3=55$	$(f_2+f_4+f_9)/3=57$	
L3	$(f_7+f_8+f_9)/3=61$	$(f_3+f_6+f_9)/3=48$	$(f_3+f_5+f_7)/3=48$	
OED Results	AL3	BL2	CL2	

[3 2 2] is not listed in the orthogonal array $L_9(3^3)$, but it is predicted as the best combination for the experiment.

23

In each iteration the OL is implemented once in Onlooker bee phase to generate deep search of candidate solutions (24 Control variables):

First, and Orthogonal Array for 2 level and 24 factors needs to formed and denoted as

$\boxed{L_{32}(2^{24})}$

Then

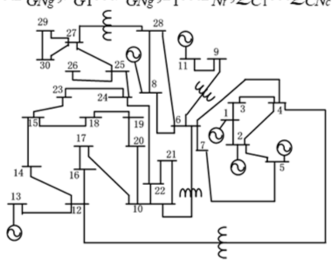
$$\begin{matrix} V_{i,1}, V_{i,2}, V_{i,3} \ldots\ldots V_{i,24} \\ V_{k,1}, V_{k,2}, V_{k,3} \ldots\ldots V_{k,24} \end{matrix} \xrightarrow{\text{Predict the best combination solution by OED}} V_{b,1}, V_{b,2}, V_{b,3} \ldots\ldots V_{b,24}$$

24

Simulation results

- IEEE 30-bus system
- 24 Control variables:

$$u^T = \left[P_{G2} \ldots P_{GN_g}, V_{G1} \ldots V_{GN_g}, T_1 \ldots T_{N_t}, Q_{C1} \ldots Q_{CN_c}\right]$$

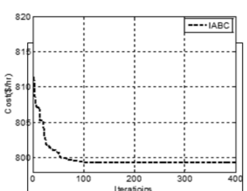


25

Case 1: Quadratic objective function $f_i(P_{Gi}) = a_i + b_i P_{Gi} + c_i P_{Gi}^2$

COMPARISON FOR FUEL COST MINIMIZATION IN IEEE 30-BUS

Method	Fuel cost ($/h)		
	Min	Avg.	Max
IABC	799.332	799.445	799.970
ABC	800.660	800.871	801.867
LDI-PSO	800.734	801.557	803.869
GSA	805.175	812.194	827.459
Parallel PSO	800.640	N/A	N/A
MSFLA	802.287	802.414	802.509
MDE	802.376	802.382	802.404
PSO	814.203	814.610	815.675
EGA	802.060	N/A	802.140
Gradient Method	804.853	N/A	N/A

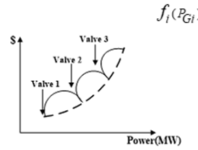


Convergence for case 1

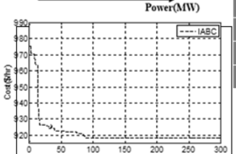
26

Case 2: Effect of valve point loading

$$f_i(P_{Gi}) = a_i + b_i P_{Gi} + c_i P_{Gi}^2 + \left|d_i \sin(e_i(P_{Gi,\min} - P_{Gi}))\right|$$



Method	Fuel cost ($/h)		
	Min	Avg.	Max
IABC	918.167	919.567	921.458
ABC	945.450	960.565	973.599
GSA	929.724	930.925	932.049
MDE	930.793	942.501	954.073
PSO	946.877	947.532	947.977



Convergence for case 2

27

Conclusions

- OPF is a non-smooth and non-convex problem
- An improved ABC (IABC) optimization technique is developed based on the orthogonal learning strategy and successfully applied to the OPF problem.
- The effectiveness of the IABC has been tested on IEEE-30 bus system and obtained promising results.

28

Questions?

29