Comparison among Five Bio-inspired Optimization Techniques for Designing Hybrid Optimization Algorithms

Duc Hoang Nguyen
Faculty of Electrical and Electronics Engineering
HCMC University of Technology
Ho Chi Minh City, Vietnam

ABSTRACT

This paper proposes ideas to create hybrid optimization algorithms that combines strengths of SFLA or PSO with strengths of GA, DE or BA. While SFLA or PSO can find optimal solutions quickly because of directive searching and exchange of information, GA, DE or BA has higher random that make it easily escape from local optima to find global solutions. Thus, hybrid algorithms are able to find optimal solutions quickly like SFLA or PSO and escape from local optima like GA, DE or BA. A hybrid SFL-Bees algorithm has illustrated for these ideas. Numerical simulations carried out have shown the effectiveness of the proposed algorithm, its ability to achieve good quality solutions and processing time, which outperforms the SFLA and BA.

General Terms

Algorithms.

Keywords

Optimization, Hybrid, PSO, SFLA, GA, DE, BA.

1. INTRODUCTION

Optimization problems are very important in practice, especially in areas such as design engineering, scientific experiments and making decisions in business. Because of increasingly complexity of these real-world optimization problems (non-linear, the number of optimized variable is large, …), they can't be solved by traditional methods such as gradient-based methods, linear-quadratic, … These motivate other methods based on natural principles and heuristics. One of these methods are bio-inspired optimization algorithms. These are randomly searching algorithms, imitate biological evolution in nature and/or swarm social behaviors.

In this paper, the author uses some bio-inspired algorithms to optimally tune parameters of fuzzy logic controllers. Basing on the results obtained, the author will point out the strengths and weaknesses of each algorithm. From that, strengths of each algorithm are combined together to create hybrid algorithms being able to find solutions better than individual algorithms.

A number of bio-inspired algorithms are very large [1-2]. In [1], the authors listed about 40 different algorithms and continue to grow. Among these, the author only concentrates on five algorithms, that is, GA - Genetic Algorithm, DE - Differential Evolution (belong to evolutionary-based optimization algorithms) and PSO - Particle Swarm Optimization, SFLA - Shuffled Frog Leaping Algorithm, BA - Bees Algorithm (belong to swarm-intelligence-based optimization algorithms).

The rest of this paper is organized as follows: section 2 introduces the overview of the GA, DE, PSO, SFLA and BA algorithms, while section 3 describes how to design and tune parameters of the fuzzy controller to balance the rotary inverted pendulum system and simulation results also presented. Section 4 presents ideas which combines the strengths of individual algorithms to create hybrid algorithms, simulation results to illustrate strengths of the suggested algorithm is also presented and the final section is conclusions.

2. OVERVIEW OF BIO-INSPIRED OPTIMIZATION TECHNIQUES

2.1 Genetic Algorithm – GA

Genetic Algorithm is perhaps the most well-known class of algorithms belonging to evolutionary-based optimization algorithms. GA is essentially the search algorithm inspired by the principle of natural selection. The basic idea is to evolve a population of individuals (also called "chromosomes"), where each individual represents a candidate solution to a given problem. Each individual is evaluated by a fitness function, which measures the quality of its corresponding solution. At each generation (iteration) the fittest (the best) individuals of the current population survive and produce offspring resembling them, so that the population gradually contains fitter and fitter individuals - i.e., better and better candidate solutions to the underlying problem. In GA the population of individuals usually evolves via a selection method, which selects the best individuals to reproduce, and via genetic operators such as crossover and mutation, which produce new offspring out of the selected individuals [3]. In this paper, BLX-α crossover operator is used as (1):

$$X_c^k(j) = \text{random}\left(\underline{X}_c^k(j), \bar{X}_c^k(j)\right) \quad (1)$$

With,

$$\underline{X}_c^k(j) = min\left(X_{p1}^k(j), X_{p2}^k(j)\right) - \alpha\left|X_{p1}^k(j) - X_{p2}^k(j)\right|$$

$$\bar{X}_c^k(j) = max\left(X_{p1}^k(j), X_{p2}^k(j)\right) + \alpha\left|X_{p1}^k(j) - X_{p2}^k(j)\right|$$

Where, $X_c^k(j)$: *jth* gene of *kth* individual

$X_{p1}^k(j), X_{p2}^k(j)$: *jth* gene of *kth* parents

Flowchart of GA as in Fig. 1.

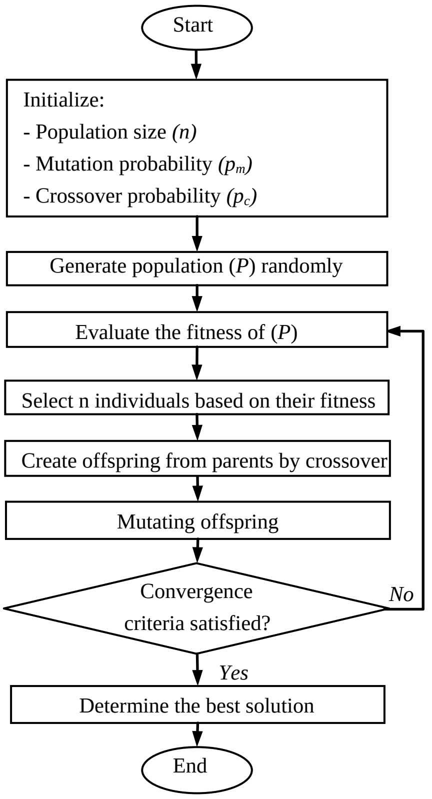

Fig. 1. Flowchart of the GA

2.2 Differential Evolution – DE

Differential Evolution grew out of Ken Price's attempts to solve the Chebychev Polynomial fitting problem that had been posed to him by Rainer Storn [4-5]. DE adopted for various optimization scenarios including constrained, large-scale, multi-objective, multimodal and dynamic optimization, hybridization of DE with other optimizers, and also the multi-faceted literature on applications of DE [6-11].

DE belongs to the class of evolutionary algorithms which use bio-inspired operations of crossover, mutation, and selection on a population in order to minimize an objective function. These operations will be briefly described in this section.

Mutation: Mutation operator is the prime operator of DE and it is the implementation of this operation that makes DE different from other evolutionary algorithms. The mutation process at each generation begins by randomly selecting three individuals in the population. There are many mutation strategies implemented in the DE, however in this paper the following strategy is used.

$$V_i^k = X_{r0}^k + F\left(X_{r1}^k - X_{r2}^k\right) \qquad (2)$$

Where X_{r0}^k, X_{r1}^k and X_{r2}^k are randomly selected and satisfy: $X_{r0}^k \neq X_{r1}^k \neq X_{r2}^k$;

Crossover: after the mutation phase is complete, the crossover process is applied to target vector X and mutated vector V in order to generate trial vector U by using the equation (3).

$$U_i^k = \langle U_i^k(j) \rangle = \begin{cases} \langle V_i^k(j) \rangle & if\ \left(rand_j(0,1) \le p_c\right)\ or\ j = rnbr(i) \\ \langle X_i^k(j) \rangle & otherwise \end{cases} \qquad (3)$$

Selection: The population for the next generation is selected from the individual in current population and its corresponding trial vector according to the rule (4).

$$X_i^{k+1} = \begin{cases} U_i^k & if\ \ f\left(U_i^k\right) \le f\left(X_i^k\right) \\ X_i^k & otherwise \end{cases} \qquad (4)$$

Where *f(.)* is the objective function.

The flowchart of the DE is illustrated in Fig.2. Further information about DE, refer to [7].

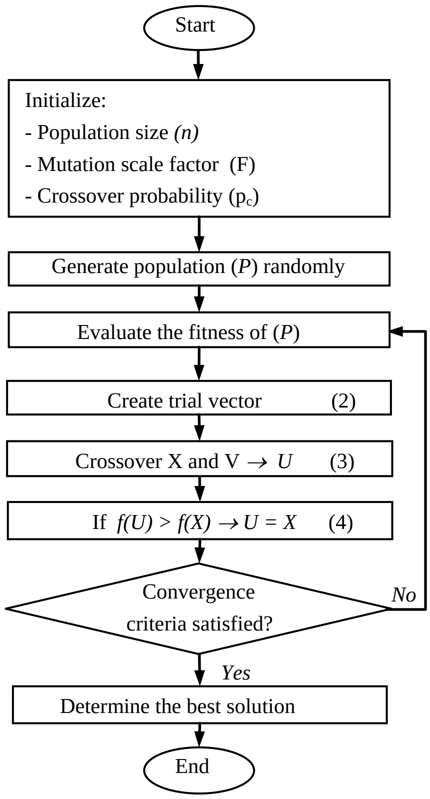

Fig. 2. Flowchart of the DE

2.3 Particle Swarm Optimization – PSO

PSO is a global optimization technique that has been developed by Eberhart and Kennedy in 1995 [12]. PSO is a population based search algorithm where each individual is referred to as particle and represents a candidate solution. Each particle in PSO flies through the search space with an adaptable velocity that is dynamically modified according to its own flying experience and also the flying experience of the other particles. In PSO each particle strives to improve itself by imitating traits from their successful peers. Further, each particle has a memory and hence it is capable of remembering the best position in the search space ever visited by it.

Velocity and position of individual particles updated as follows:

$$V_i^{k+1} = \omega V_i^k + c_1 rand_1 \left(X_{pi} - X_i^k \right) + c_2 rand_2 \left(X_g - X_i^k \right) \quad (5)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (6)$$

Where:

V_i^{k+1} : velocity of particle i at loop k+1.

X_i^{k+1} : position of particle i at loop k+1.

ω : inertia weight.

c_1, c_2 : cognitive and social parameters.

$rand_1, rand_2$: random numbers between 0 and 1.

X_{pi} : best "remembered" individual particle i position.

X_g : best "remembered" swarm position.

Flowchart of PSO as in Fig. 3.

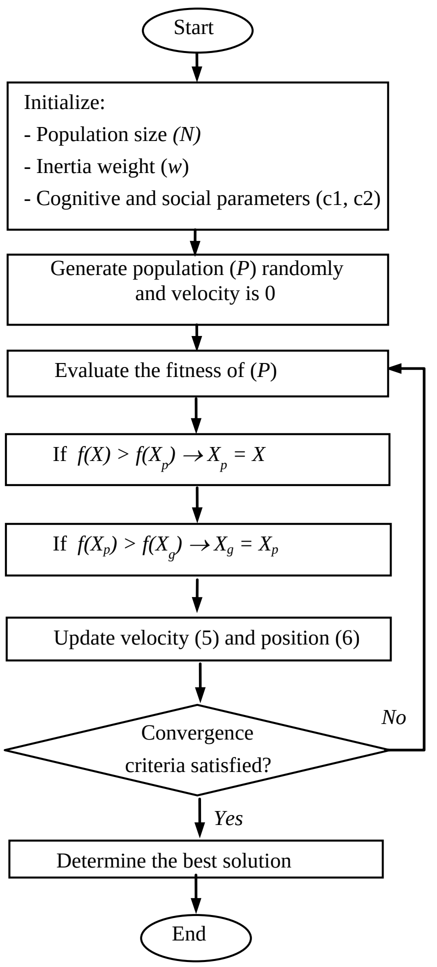

Fig. 3. Flowchart of the PSO

2.4 Shuffled Frog Leaping Algorithm – SFLA

The SFLA is a meta-heuristic optimization method that mimics the memetic evolution of a group of frogs when seeking for the location that has the maximum amount of available food. The algorithm contains elements of local search and global information exchange. The SFLA involves a population of possible solutions defined by a set of virtual frogs that is partitioned into subsets referred to as memeplexes. Within each memeplex, the individual frog holds ideas that can be influenced by the ideas of other frogs, and the ideas can evolve through a process of memetic evolution. The SFLA performs simultaneously an independent local search in each memeplex using a particle swarm optimization-like method. To ensure global exploration, after a defined number of memeplex evolution steps (i.e. local search iterations), the virtual frogs are shuffled and reorganized into new memeplexes in a technique similar to that used in the shuffled complex evolution algorithm.

In addition, to provide the opportunity for random generation of improved information, random virtual frogs are generated and substituted in the population if the local search cannot find better solutions. The local searches and the shuffling processes continue until defined convergence criteria are satisfied. The flowchart of the SFLA is illustrated in Fig 4.

The idea updating frog leaping rule which is expressed as:

$$D = r.c(X_b - X_w) \quad (7)$$

$$X_w(new) = X_w + D \quad (8)$$

where X_b and X_w are identified as the frogs with the best and the worst fitness respectively; r is a random number between 0 and 1; c is a constant chosen in the range between 1 and 2. [13-17]

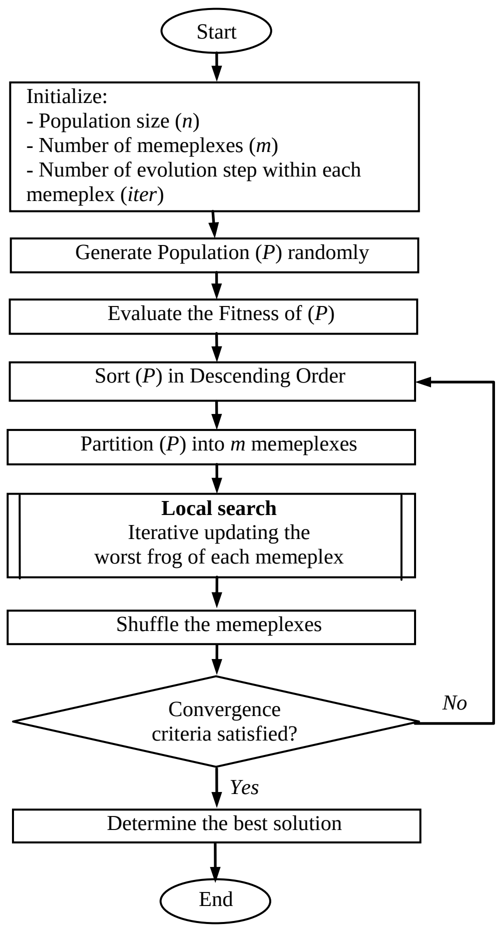

Fig 4: Flowchart of the SFLA

2.5 Bees Algorithm – BA

The Bees Algorithm is an optimization algorithm inspired by the natural foraging behavior of honey bees to find the optimal solution. The algorithm requires a number of parameters to be set, namely: number of scout bees (n), number of sites selected out of n visited sites (m), number of best sites out of m selected sites (e), number of bees recruited for best e sites (n2), number of bees recruited for the other (m-e) selected sites (n1), initial size of patches (ngh) which includes site and its neighborhood and stopping criterion. The algorithm starts with the n scout bees being placed randomly in the search space. The fitness of the sites visited by the scout bees are evaluated. Bees that have the highest fitness are chosen as "selected bees" and sites visited by them are chosen for neighborhood search. Then, the algorithm conducts search in the neighborhood of the selected sites, assigning more bees to search near to the best e sites. The bees can be chosen directly according to the fitness associated with the sites they are visiting. The flowchart of the BA is illustrated in Fig. 5. [18-19].

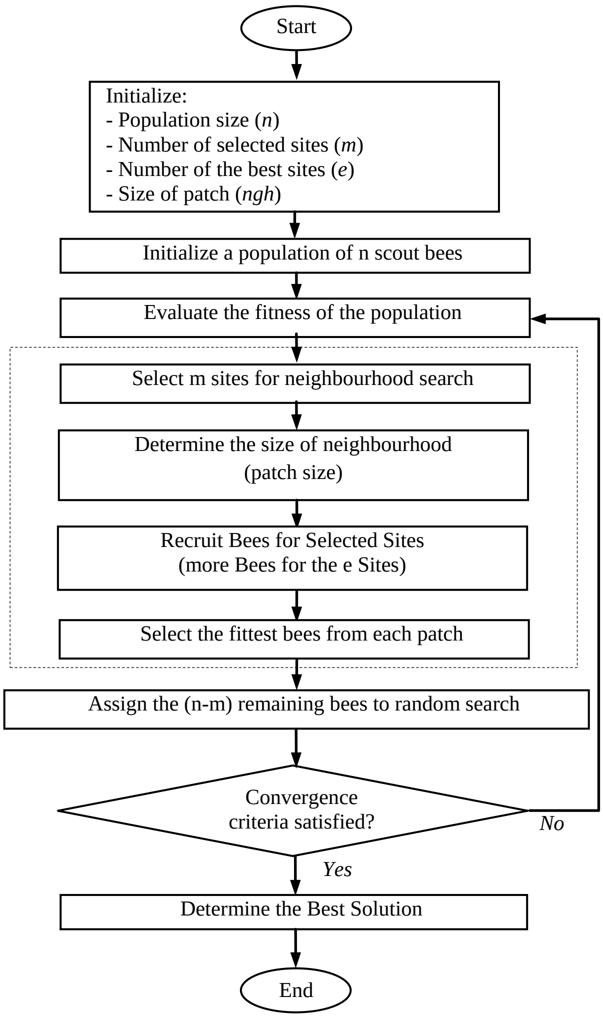

Fig 5: Flowchart of the BA

3. TUNING PARAMETERS OF FUZZY LOGIC CONTROLLER

This section only presents simulation results while details of design of a fuzzy logic controller for balancing the rotary inverted pendulum in the upright position refers [20]. Evolution of quadratic performance index in case of tuning 5 and 12 parameters are presented in Fig. 6 and 7.

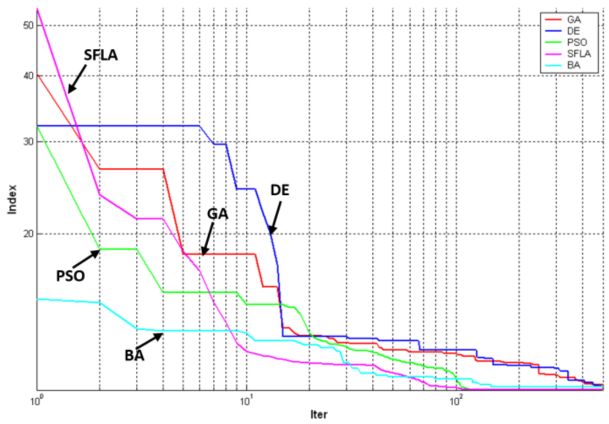

Fig 6: Evolution of index in case of tuning 5 parameters

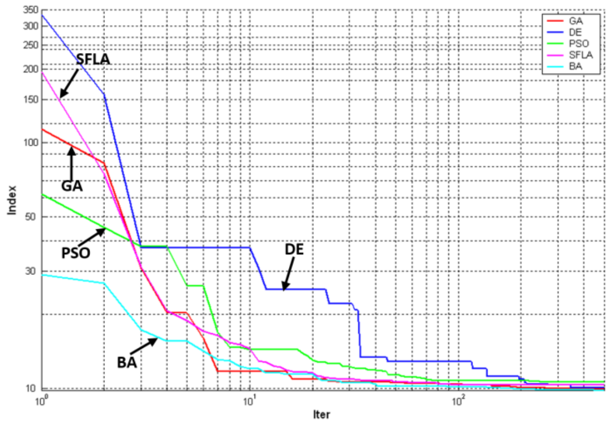

Fig 7: Evolution of index in case of tuning 12 parameters

The following observations can be drawn from the above objective function plots:

✓ In case of tuning 5 variables: PSO and SFLA algorithms give better results in terms of convergent rate and objective function value (faster convergent rate and smaller objective function value compared to the remaining algorithms). Convergent rate of GA and DE algorithms is rather slow. BA has faster convergent rate and smaller objective function value compared to GA and DE.

✓ In case of tuning 12 variables: GA, SFLA and BA have better convergent rate while GA, DE and BA have better objective function value. BA has the smallest objective function value.

From these remarks, it can be concluded that when the number of optimized variables is small, PSO and SFLA algorithms find solutions better than GA, DE and BA in terms of convergent rate and quality of solutions. When the number of optimized variables is large, GA, DE and BA algorithms find objective function value better than PSO and SFLA. The reason is that PSO and SFLA are less random than the remaining algorithms in searching for optimal solutions. Worse agents (worse solutions) in PSO and SFLA always follow better agents (better solutions) to update its position. Particularly, for PSO method, particles in population always fly to the best particle (best solution); for SFLA method, worse frogs always jump to better frog to search for more food (better solution). However, when agents in population are closer, they can't escape from their position and result is that they are trapped into a position in the solution space. Hence, PSO and SFLA is premature convergence. Whereas, update of individuals of GA, DE and BA is more random. So, these algorithms have slower convergence. For instance, in GA method, two parents are selected randomly to create offspring using crossover operator; in DE method, trial vector is created from 3 randomly selected vectors and crossover with fourth vector; and, in BA method, the majority of new bees in population are created randomly. Therefore, as the number of variables that need to be optimized are large, these algorithms (GA, DE and BA) have ability of finding better solutions, they easily escape from local optima compared to PSO and SFLA methods. However, these don't mean that GA, DE and BA methods are completely random. They are also deterministic. For example, in GA method, only parents having the best fitness are selected to crossover; or, in BA, only searching for around the best bees many times to find the possibly best solution.

4. IDEAS TO DESIGN HYBRID BIO-INSPIRED OPTIMIZATION METHODS

4.1 Remarks

As presented in section 2, bio-inspired optimization algorithms are methods which imitate biological evolution of the creatures in nature or their behavior. Although these methods have different strategies to solve optimization problems, they have many similarities. Two features considered here are local search and global search. As can be seen from the above overview, all algorithms have local search and global search operators, however each has distinct ways to update new individuals. This leads to different convergent rate and quality of solution. In particular, PSO and SFLA are less random than the remaining algorithms in searching for optimal solutions. Bad agents always follow good agents to update their positon. This makes PSO and SFLA have fast convergent rate. However, as agents' position is closer (*Xw* and *Xb* or *Xg* for SFLA; *Xi* and *Xg* or *Xp* for PSO), it's almost unchanged. That means that they are trapped into a position in the solution space and can't escape from their position. Hence, PSO and SFLA is premature convergence. On the other hand, GA, DE and BA are more random as finding solutions. This makes GA, DE and BA have slower convergent rate. However, due to more random as updating position, these algorithms easily escape from local optima, especially as the number of individuals in population is large.

Simulation results in section 3 has proven that these remarks are right. So, the author has ideas to design hybrid optimization methods as follows.

4.2 Ideas

From the above remarks, it's able to see that algorithms have the faster convergent rate, the easier getting stuck in local optima, such as PSO and SFLA. On the other hand, algorithms have slow convergent rate, they're able to escape from local optimal, such as GA, DE and BA.
From this conclusion, the author proposed an idea that combining fast convergent rate of algorithms PSO or SFLA with ability of escaping from local optima of algorithms GA, DE or BA to create hybrid algorithms that have ability to compromise between convergent rate and quality of solution as solving optimization problems.
To illustrate this idea, the author proposed a hybrid SFLA - Bees algorithm which combines fast convergent rate of algorithm SFLA with ability of escaping from local optima of algorithm BA.

The details of designing this hybrid algorithm are described in the paper [21]. The author here only presents results.

4.3 Illustration

Below are the results when using the hybrid SFL-Bees algorithm to find optimal solutions of F7 function. Results show that mean objective function value of hybrid SFL-Bees algorithm is greater than value of BA but smaller than value of SFLA (hybrid SFL-Bees algorithm has strength of BA, i.e. be able to globally search for solution) and average processing time of hybrid SFL-Bees algorithm is smaller than time of BA but greater than time of SFLA (has strength of SFLA, i.e. can find optimal solution quickly) as demonstrated in Fig. 8 and Fig.9. Results for F8 function also show strengths of hybrid SFL-Bees algorithm.

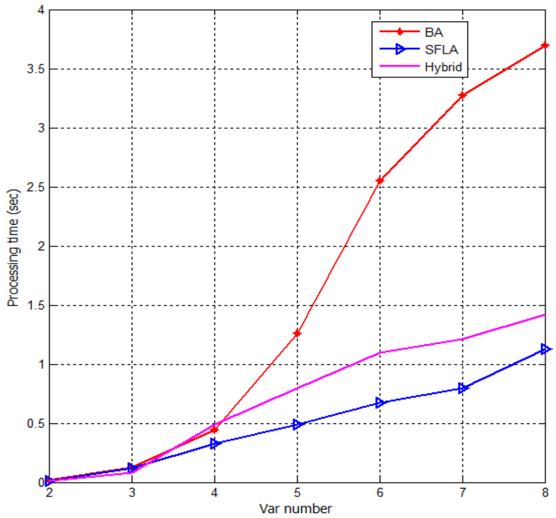

Fig 8: Plot of Processing Time

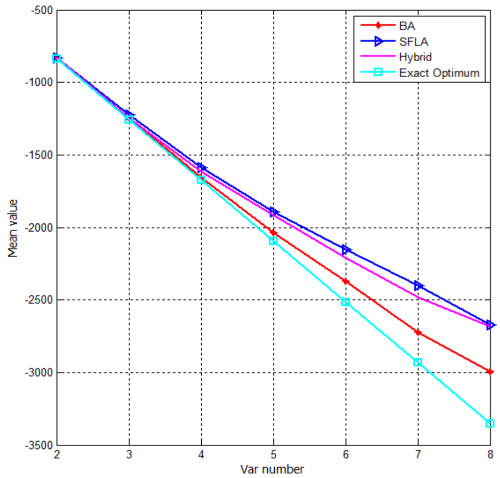

Fig 9: Plot of Mean value

5. CONCLUSION

In this paper, five bio-inspired optimization techniques have been overviewed. Strengths and weaknesses of each algorithm are presented. Based on these results, the author proposed ideas to create hybrid methods. That is to combine the strengths of each algorithm together. The author also illustrates these ideas by proposing a novel algorithm called Hybrid SFL-Bees Algorithm that combine strengths of SFLA and BA, namely ability to find global optimal solution quickly. Simulation results show that hybrid algorithm outperforms each individual algorithm. The future work is to combine these algorithms to create more the hybrid algorithms and apply them for solving other kinds of optimization problems, for instance, tuning parameters of fuzzy controller.

6. REFERENCES

[1] Iztok Fister Jr., Xin-She Yang, Iztok Fister, Janez Brest, Duˇsan Fister, "A Brief Review of Nature-Inspired Algorithms for Optimization", Elektrotehniˇski Vestnik 80(3): pp.1–7, 2013.

[2] E. Elbeltagi, T. Hezagy & D. Grierson, "Comparison among five evolutionary-based optimization algorithms", Advanced Engineering Informatics, vol.19, 43-53, 2005.

[3] Evangelos Triantaphyllou and Giovanni Felici, "Data mining and knowledge discovery approaches based on rule induction techniques" – Springer, Chapter 12, 2006.

[4] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces," Tech. Report, International Computer Science Institute (Berkeley), 1995.

[5] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces", Journal of Global Optimization, vol. 11, Dec. 1997, pp. 341-359.

[6] Neri, F. & Tirronen, "Recent advances in differential evolution: A survey and experimental analysis", Artificial Intelligence Review 33(1-2): 61-106. V. 2010.

[7] Swagatam Das, Sankha Subhra Mullick, and P. N. Suganthan, "Recent Advances in Differential Evolution – An Updated Survey", Swarm and Evolutionary Computation, Volume 27, April 2016, Pages 1–30.

[8] K. V. Price, R. M. Storn, and J. A. Lampinen, "Differential Evolution: A Practical Approach to Global Optimization", Springer-Verlag, Berlin, Heidelberg, second edition, 2006

[9] M.-F. Han et al,"Differential evolution with local information for neuro-fuzzy systems optimization", Knowledge-Based Systems 44, 78–89, Elsevier 2013.

[10] Cheng-Jian Lin, Chih-Feng Wu, Hsueh-Yi Lin & Cheng-Yi Yu, "An Interactively Recurrent Functional Neural Fuzzy Network with Fuzzy Differential Evolution and Its Applications", Sains Malaysiana 44(12)(2015): 1721–1728.

[11] Patricia Ochoa, Oscar Castillo and José Soria, "Differential Evolution with Dynamic Adaptation of Parameters for the Optimization of Fuzzy Controllers", Recent Advances on Hybrid Approaches for Designing Intelligent Systems Studies in Computational Intelligence 547, Springer 2014.

[12] J. Kennedy and R. C. Eberhart.: Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ,5(3), 1942–1948, (1995).

[13] Muzaffar Eusuff, Kevin Lansey and Fayzul Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization", Engineering Optimization Vol. 38, No. 2, March 2006, 129–154.

[14] A. Darvishi, A. Alimardani, B. Vahidi , S.H. Hosseinian, "Shuffled Frog-Leaping Algorithm for Control of Selective and Total Harmonic Distortion", Journal of Applied Research and Technology, Volume 12, Issue 1, February 2014, Pages 111–121.

[15] Yi Han, et al,"Shuffled Frog Leaping Algorithm for Preemptive Project Scheduling Problems with Resource Vacations Based on Patterson Set", Journal of Applied Mathematics,Volume 2013 (2013), Article ID 451090.

[16] Daniel Mora-Melia, Pedro L. Iglesias-Rey, F. Javier Martínez-Solano and Pedro Muñoz-Velasco, "The Efficiency of Setting Parameters in a Modified Shuffled Frog Leaping Algorithm Applied to Optimizing Water Distribution Networks", Water 2016, 8, 182, MDPI.

[17] Dina M. Said, Nabil M. Hamed, Almoataz Y. Abdelaziz, "Shuffled Frog Leaping Algorithm for Economic Dispatch with Valve Loading Effect", International Electrical Engineering Journal, Vol 7 No 3, 30 JUL, 2016.

[18] D.T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri , S. Rahim and M. Zaidi, "The Bees Algorithm – A Novel Tool for Complex Optimization Problems", Manufacturing Engineering Centre, Cardiff University, Cardiff CF24 3AA, UK.

[19] D.T. Pham, A.Haj Dqrwish, E.E. Eldukhri, and S. Otri.: Using the Bees Algorithm to tune a fuzzy logic controller for a robot gymnast. Proceedings of the 3rd Virtual International Conference on Intelligent Production Machines and Systems, (2007).

[20] Duc-Hoang Nguyen and Manh-Dung Ngo, "Comparing convergence of PSO and SFLA optimization algorithms in tuning parameters of fuzzy logic controller", AETA 2015.

[21] Duc-Hoang Nguyen, "A Hybrid SFL-Bees Algorithm", International Journal of Computer Applications 128(5):13-18, October 2015. Published by Foundation of Computer Science (FCS), NY, USA.