International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161
©2018 INPRESSCO®, All Rights Reserved
Available at http://inpressco.com/category/ijcet

Research Article

Stability Enhancement of Transfer Function by Ant Colony Optimization and Particle Swarm Optimization Algorithm

Prabha Singh[#], Jaya Mishra[#], Chaman Yadav^*, Kushal Tiwari^ and Shashank Singh[$]

[#] SSEC (SSTC) Junwani, Bhila, India
^CREDA, Raipur, India
[$]Markfed Kanker, India

Received 16 April 2018, Accepted 18 June 2018, Available online 23 June 2018, **Vol.8, No.3 (May/June 2018)**

Abstract

This paper presents the concepts of three evolutionary algorithms i.e, ant colony optimization and particle swarm optimization algorithm. An evolutionary algorithm copies the way how evolution occurs in the nature. There are various types of evolutionary algorithms. This paper focuses on ACO and PSO algorithms. ACO provides solution to various optimization problems. It follows the principle of survival of the fittest. Various problems such as knapsack problem, TSP(travelling salesman problem) can be solved using genetic algorithm. Ant colony optimization is a heuristic algorithm which follows the behaviour of ants i.e., the way ants seek food in their environment by starting from their nest. Particle swarm optimization algorithm (PSO) is also an optimization algorithm which also uses a method of searching using some heuristics.

Keywords: *Ant colony optimization, Evolutionary Algorithm, Particle swarm optimization algorithm, Swarm intelligence.*

1. Introduction

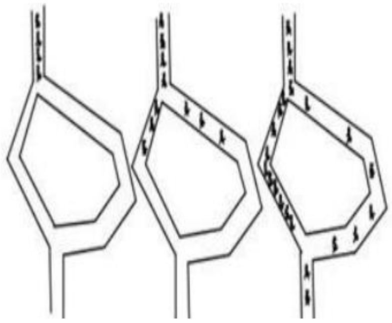
An evolutionary algorithm (EA) is an algorithm first given by Charles Darwin in 1859. It provides solution to various optimization problems. It copies the process of evolution that occurs in the nature i.e, mutation, recombination and natural selection. It starts from an initial population and by applying these operators finds the fittest generation of population. An ACO (Ant colony optimization) algorithm is also a heuristic algorithm, first found by Marco Dorigo, which provides an optimal solution by mimicking the way ants find their food. ACO is a type of SI (swarm intelligence) technique. PSO (Particle Swarm) algorithm, first found by Kennedy and Eberhart, also provides an optimal solution to problems and is inspired by the flock of birds, fishes and by herds of animals. It mimics the way they find food in their environment and follows the approach of information sharing. This paper constitutes the following sections. The origin of ACO . Section III illustrates basics of ACO (Ant Colony Optimization) algorithm. Section III Illustrates PSO (Particle swarm optimization) algorithm. Section IV illustrates simulation and Section V illustrates results conclusion. Section VI illustrates future work. Se illustrates references.

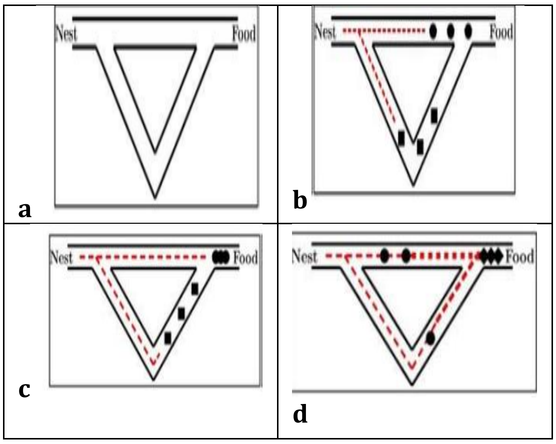
*Corresponding author's ORCID ID: 0000-0002-8808-3191
DOI: https://doi.org/10.14741/ijcet/v.8.3.35

2. Ant Colony Optimization Algorithm

ACO was initially given by Marco Dorigo. This algorithm is inspired by the behaviour of ants in the real world.

Ants follow the principle of survival of colonies rather than survival of the individual as they always stay in their colonies. Ants are known for doing very difficult tasks in a very simple way. They search the path which is the shortest one between a food source and their nest. Initially, they wander randomly in their environment. While exploring the path, they deposit a chemical i.e, pheromone on their path which they visit. If an ant finds food, it again goes back to its nest by again depositing that pheromone trail on the ground. Other ants smell these pheromone trails. The chances of an ant choosing a path depends on the amount of pheromone deposited on that path. Higher the pheromone concentration on the path , probability of selecting that path by other ant also gets higher. So, the probability of choosing a path and density of pheromone are directly proportional.





The above figure shows how ants find their food in the environment. In fig.(a), no ant is searching for food. They are in their nest. So, no chemical is there on the path. In fig.(b), ants have started exploring the path to find the food. Different ants will follow different paths and they will keep on depositing the pheromone on the path which they will follow. Some ants will follow the path shown by circle and some by rhombus. In fig.(c), the ants that follow the path depicted by circle will reach the food source earlier as compared to ants that follow the path depicted by rhombus. It shows that path shown by circles was the shorter one. So, when the ants will go back to their nest, their chances of choosing the path shown by circle will increase. In fig.(d), density of pheromone on the path depicted by circle will be more. The ACO algorithm is as follows:- While conditions for termination do not met {Activites Schedule Solution Construction Based On Ants() Update Pheromone() Daemon-Actions() } In the above algorithm Daemon-Actions() part is optional. ACO is an iterative algorithm in which the while loop controls the run time of this algorithm. In Solution Construction Based On Ants(), a colony of ants build a set of solution. These solutions are then evaluated based on an objective function. In Update Pheromone (), two opposite mechanisms are used i.e, pheromone deposit and pheromone evaporation. In pheromone deposit, artificial ants increase pheromone values on visited components connections. In pheromone evaporation, ants decrease the values of pheromone trails on all connections by a same value. In Daemon-Actions(), the daemon may decide to deposit extra pheromone on the solution components that belong to the best solution found so far.

3. Particle Swarm Optimization

Particle Swarm Optimization (PSO) was originally found by Dr. Kennedy and E berhart in 1995. Like ACO, it is also based on swarm intelligence. It is inspired by the bird and fish flock movement behavior. Birds search for food by scattering or going together in groups. While searching, there is always a bird who has better food resource information, and who is perceptible of the place where food can be found and it can smell the food very well. Because Birds transmit the information about food source to each other, that's why they flock at the place where food can be found. This algorithm is very simple and has many advantages. That's why, it can be applied to various fields such as, neural network training, machine study etc. In basic PSO, there are n particles, and their position gives a potential solution in Ddimensional space. The most optimist position during the movement of a particle and the position of the most optimist particle in the surrounding, both affect the position of the particle. Algorithm for PSO:- Initialize the particle position and velocities randomly. While termination condition not met do

For each particle j Evaluate Yj , which is the fitness, at Xj which is the current position If Yj is better than Pbest j the n update Pbest j and Pj If Yj is better than Gbestj then update Gbest j and Gj For each particle j:- Xj is a vector which depicts the position and Yj denotes value of objective function. Vj is a vector which denotes velocity of the particle. Pj is the best position found so far and Pbest j is objective function score. Gj is the best position found so far in its neighbourhood and Gbest j is objective function score of Gj. In updating velocity, new velocity is created by three terms:-

1) Inertia Term, because of which particle go in direction same as before by applying adjustments to old velocity.
2) Cognitive Term(Personal best), because of which particle moves back to position which was best previously.
3) Social Learning Term, because of which particle goes back to neighbour's previous best position.

PSO can be applied to both engineering problems and scientific research as it is based on intelligence. There are no mutation calculations in PSO like genetic so, it is very simple.

4. Simulation and Results

From the above discussion it is clear that by using the algorithm the stability criteria of the plant is achieved.

In a power plant, both active and reactive power demands continually vary the rising or falling trend. Power input must therefore be continuously regulated to match the active power demand; otherwise the machine speed will change with consequent change in frequency, which may be highly undesirable. Also the excitation of generators must be continuously regulated to match the reactive power demand with reactive generation, failing which the voltage at various system buses may go beyond the prescribed limits. Consider a block diagram of a plant which is in series with the controller to maintain the stability limit for the plant.

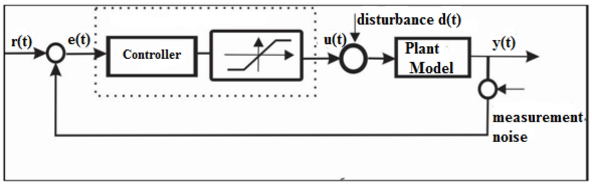


Fig 4.1 Block diagram of Plant model with controller

In these block diagram if the controller is eliminated and the system is run without any controller then the system gives unstable results.

Matlab Modelling of the plant

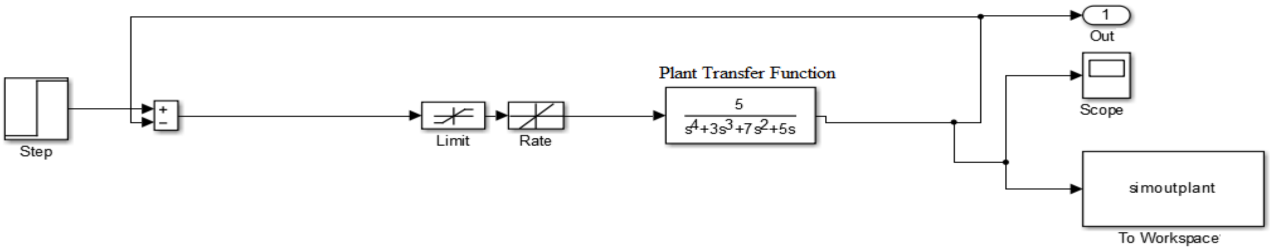


Fig 4.2 Matlab modelling for plant without controller

As the theory specified that each and every system has their transfer function, so considering in this matlab model that the plant is having transfer function

$$Tf = \frac{5}{s^4+3s^3+7s^2+5s}$$

In this model there are two factors which are connected in series with the plant i.e, the limit and the rate. The rate limiter block limits the first derivative of the signal passng through it. The output do not changes faster than that of the limitation criteria by the use of such type of blocks.

After simulating theblock diagram in Matlab it gives the required result.

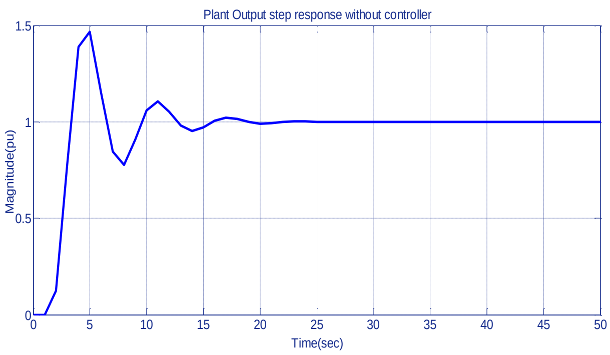


Fig. 4.3 Output response of plant without controller

The above response shows that the plant transfer function is unstable; to make it stable a controller is used. The block diagram shown below is containing a controller in series with the plant transfer function. The controller is of PID type i.e. Proportional Integral and Derivative which helps the system to become stable. The proportional controller helps to improve the transient response by adding finite zero to open loop transfer function, whereas the integral controller reduces the steady state error by adding pole at the origin of the system transfer function.

The system is not stable hence the values of the required factor are as follows:

RiseTime: 1.4011
SettlingTime: 17.2781
SettlingMin: 0.7767
SettlingMax: 1.4665
Overshoot: 46.6527
Undershoot: 0
Peak: 1.4665
PeakTime: 5

But the problem is that how to tuned the PID controller to achieve stability limit? The answer is that by using algorithm that is introduced in the above chapters. By using the concept of both PSO (Particle Swarm Optimization) and ACO(Bacteria Foraging Optimization) it would gives the result of the controller gain after the several iteration which is accurate to provide the system stability.

Hence the system model containing the controller is shown below which is tuned by using the PSO and ACO algorithm.

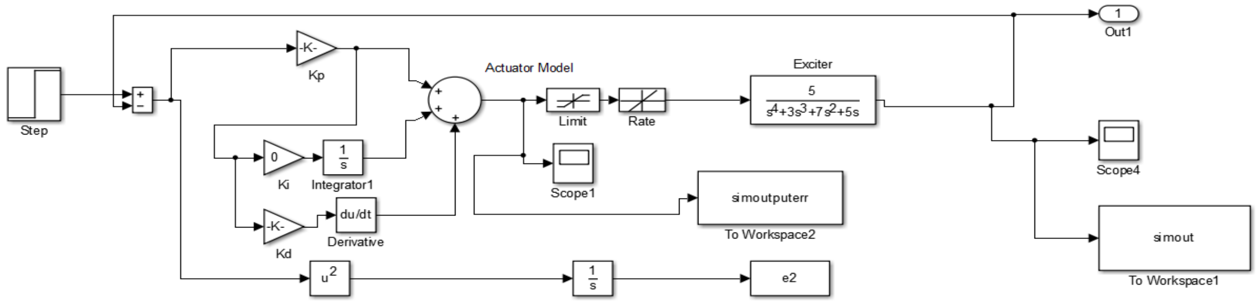


Fig. 4.4 Matlab modelling for Plant with PID controller

As the figure shows above that manipulating the values of gain by PSO PID technique or PSO ACO technique it gives the required results in both PSO as well as in ACO. After the execution of algorithm (in both ACO as well as in PSO) the required values of proportional gain, integral gain and derivative gain are saved in workspace and from the work space the required values are taken up by the PID controller.

Conclusions

In this paper, a review to three evolutionary algorithms, i.e, Particle Swarm Optimization (PSO)and Ant colony optimization(ACO) is given. Genetic Algorithm copies the way evolution occurs in the nature, while ACO is based on the behaviour of ants and PSO is based on the behaviour of flock of birds and fishes. All these algorithms can be used to solve various optimization problems.

Future Work

This paper provides a review to Genetic, ACO and PSO algorithms. In the future, enhancements can be done in these algorithms to provide better solutions to optimization problems. Also these algorithms together can be compared with each other to find most effective algorithm. These algorithms can also be combined together to make a hybrid algorithm to solve various problems.

References

Rijwan Khan, Dr. Mohd. Amjad (Dec 2014),A review on automated test case generation using genetic algorithms, in International Journal of Advanced research in Computer Science and software Engineering, Volume 4, Issue 12, pp-201-204.

Qinghai Bhai (Feb 2010),Analysis of Particle Swarm Optimization Algorithm, in CCSE journal, Volume 3, Issue 1, pp- 180-184.

ZAr Chi Su SuHlaing and May Aye Khine (Dec 2011), Solving travelling Salesman Problem by Using Improved Ant Colony Optimization Algorithm, International Journal Of Information and Education Technology, Vol No. 1, Issue No. 5, pp- 404-409.

Christian Blum, Ant Colony Optimization- Introduction and recent trends, science direct.

UtkarshJaiswal, Shweta Aggarwal (July 2011), Ant Colony Optimization, in International Journal of Scientific & Engineering Research , Volume 2, Issue 7, pp- 201- 208.

Ruchika Malhotra, Chand Anand, Nikita Jain ApporvaMittal (June 2014),Comparison of search based techniques for automated test data generation, in International Journal of Computer Applications, Volume 95, Issue 23.

V. Selvi, Dr. R. Umarani (Aug 2010), Comparative Analysis of Ant Colony and Particle Swarm Optimization Techniques, in International Journal of Computer Applications (0975 – 8887), Volume 5,Issue4.

SapnaKatiyar (Dec 2013), A comparative Study of Genetic Algorithm and Particle Swarm Optimization, in AKGEC International Journal of Technology, Volume 2, Issue 2, pp-21-24.

Krishna H. Hingrajiya, Ravindra Kumar Gupta, Gajendra Singh Chandel (Aug 2012), An ant colony optimization algorithm for solving travelling salesman problem, International Journal of scientific and research publications, Volume 2, Issue 8