Ant Colony Based Load Flow Optimisation Using Matlab

[1]Kapil Upamanyu, [2]Keshav Bansal, [3]Miteshwar Singh
Department of Electrical Engineering
Delhi Technological University, Shahbad Daulatpur, Main Bawana road, New Delhi-110042, India
[1]kapil.upamanyu@gmail.com, [2]keshav.dtu@gmail.com, [3]miteshwar@yahoo.com

Abstract: This paper presents a solution to the network constrained optimisation problem using Ant Colony Optimisation (ACO) algorithm. This algorithm consists of artificial agents, called ants, which cooperate among themselves to find an optimal solution to the Constrained Load Flow (CLF) problem. These ants communicate with each other using pheromone matrix, pheromone being a chemical released by real ants for finding the shortest path from food source to their nest. The study revolves around finding the optimum settings of three tap changing transformers, which gives minimum power losses while maintaining a constant demand. The above method is applied to the standard IEEE 30-bus system and results are compared with the solutions obtained from conventional methods. The MATLAB code is developed for the same, and compared with the conventional approach.

Index Terms – Ant colony Optimisation (ACO), Constrained Load Flow (CLF), Genetic Algorithms (GA)

I. INTRODUCTION

The CLF problem deals with the offline adjustment of the power system control variables in order to satisfy physical and operating constraints [1]. Every year an immense amount of power is lost due to transmission, generation and distribution imperfections and non-idealities. The objective is to minimise the transmission losses by changing the tap setting of switching transformers. This gives us a constrained load flow problem to which ACO optimisation is applied.

Various methods can be followed for this purpose, conventional being testing all possible combinations of tap settings of transformers and finding the combination which corresponds to minimum power losses. But this method is associated with a large computational time and a need for a large computational memory.

Genetic Algorithms (GA) offer a great advantage of less computational time. GA has been used to evolve computer programs for specific tasks, and to design other computational structures [2]. GA involves heuristic methods such as swarm particle optimisation and ACO.

Heuristic methods do not guarantee the optimal solution to the given problem but the algorithm can be fine tuned so as to obtain a near optimum solution with a high accuracy and low computational time. The significant part of heuristic algorithms comprises metaheuristic methods, which differ from the classical methods in that they combined the stochastic and deterministic composition. It means that they are focused on global optimization, not only for local extremes [5]. The big advantage of metaheuristic is that they are built not only for solving a concrete type of problem, but they describe general algorithm in that, they show only the way, how to apply some procedures to become solution of the problem.

The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posterior information about the structure of previously obtained good solutions [3]. This metaheuristic algorithm is developed on the natural behaviour of ants of transporting their food from the source to their nest through shortest path. To find the shortest path, moving ants lay some pheromone on the ground, so an ant encountering a previously trail can detect it and decide with high probability to follow it. As a result, the collective behaviour that emerges is a form of a positive feedback loop where the probability with which an ant chooses a path increases with the number of ants that previously chose the same path [4].

In power systems, the ACS has been applied to solve the optimum generation scheduling problems [6], [7] and the optimum switch relocation and network reconfiguration problems for distribution systems [8], [9]. It is rather difficult to find a single search space, configuration, and a parameter set of an ACS algorithm that can satisfy every optimization problem. The ACS algorithm proposed in this paper formulates the CLF problem as a combinatorial optimization problem, implying the settings of the three tap changing transformers are combined to obtain an optimum solution to the CLF.

II. CLF PROBLEM

The load flow problem can be expressed by two sets of non linear equations:

$$Y=g(X, U)$$
$$Z=f(X, U) \qquad [10]$$

Where

Y Vector of nodal power injections

Z Vector of constrained variables (power flows, reactive

Powers of PV buses, etc.);
X State vector (voltage angles and magnitudes);
Control vector (transformer tap settings)
U Voltage and power at PV buses

The objective of the Constrained Load Flow problem is to maintain some or all elements of vectors X & Z within given operating limits and hence determine the combination of the control variables which satisfies such limits and provides the system with minimum power losses.

III. BASIC CONCEPTS OF ACS

Ant Colony Algorithms are based on the principle of stimulating the behaviour of real ants. [11][12] ACS finds its basis on the fact that as a group the ants are able to find the shortest path to their destination using simple communication methods. For real life ants the communication medium is the chemical, pheromone which they leave as a trail marker on the ground. The pheromone evaporates over time unless an additional amount is deposited, thus indicating that a greater number of ants prefer this path. In other words the trail with the greatest pheromone levels is indicative of high degree of optimisation.

Consider the minimisation of the CLF Power Loss Problem (S, P, C) where S is the matrix of all possible solutions, P is the objective function to be minimised and C is the matrix of constraints to be adhered to[13]. For the solution of the CLF problem artificial ants create feasible solutions by following all relevant possible routes. The probability of an ant choosing node j is:

$$P(c_{h+1} = j \mid X_h) = \begin{cases} \dfrac{F_{ij}(\tau_{ij})}{\sum\limits_{(i,j)\in N_i^k} F_{il}(\tau_{il})}, & \text{if } (i,l) \in N_i^k \\ 0, & \text{otherwise} \end{cases}$$

Where (imp) belong to N_i^k satisfying the constraints matrix C.

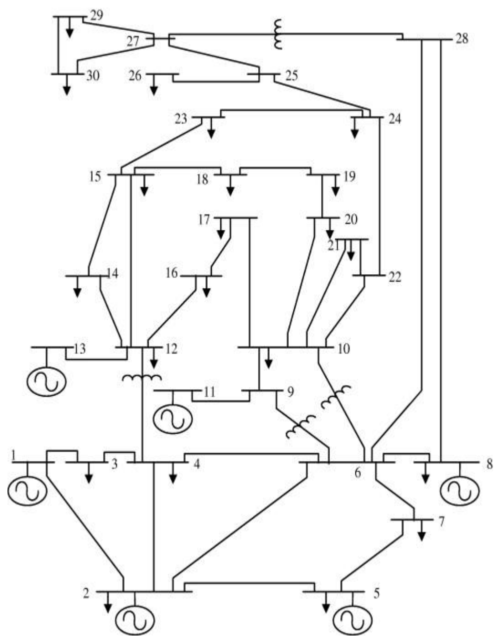
IV. ANT COLONY ALGORITHM FOR CLF

1. Create the search space that represents the discrete settings (states) of the control variables (Stages)
2. Insert the pheromone matrix according to the nodes of the Ant System Graph where n is number of states and m is the number of stages
3. Initialize the pheromone matrix to a common value
4. Allow M wants to run across the various possible combinations of the states and stages
5. Calculate Power Losses
6. Obtain the control variable settings corresponding to minimum power losses.
7. Update the Pheromone matrix first locally then globally corresponding to tap settings.
8. Repeat for a number of Iterations or till the time convergence occurs
9. Run a set of M new ants in each iteration
10. Display the Result
11. Obtain results 10 times and compare results and their variance.

V. METHODOLOGY USED

In this system we have considered all the parameters fixed, except the three transformer tap settings between line 6 & 9, line 6 & 10, line 4 & 12.The objective is to minimize the power losses associated with this system by changing tap settings of these three transformers. We have used Newton Raphson Method to determine the losses associated with different combinations of settings of these three transformers.

Each transformer tap is varied in steps of 16, each step indicating a value of 0.01. The range of turn-ratio varies from 0.9 to 1.05. Hence there are 4096(16*16*16) possible combinations of transformer settings corresponding to which losses can be calculated. Our task is to find out that combination which gives minimum power loss in the minimum possible iterations with the maximum possible accuracy.



After the initial matrices corresponding to bus voltage and power levels have been defined the pheromone probability matrix is initialised in MATLAB. Initially all pheromone matrix elements are given a constant value and a random ant distribution function is run in the first iteration.

```
for t=1:16      % Loop for running the 16 Artificial Ants
ran=rand;    % Using Random function in first Iteration
 p=0;
 n=0;
 for r=1:al  %Taking a Random Probability based Value for Tap A
p=p+pa(r)/sum(pa); %Determining Probability for choosing  path
   if ran<p && n==0
     sa=r;
       n=n+1;
   end
  end
  a(sa);
  ran=rand;
  p=0;
  n=0;
```

As the number of iterations increases the coefficient of pheromone updation is correspondingly changed under the assumption that we are nearing the optimum solution and hence greater weightage is given to the results which are obtained in the later stages of the process.

```
  if q<5            % No. Of Iterations
     cp=2000;      %Coefficient of Pheromone Updation
  else if q<40
     cp=300000;   %Coefficient of Pheromone Updation
  else
     cp=300000;  %Coefficient of Pheromone Updation
  end
 else if f==0
  cp=270000;    %Coefficient of Pheromone Updation
```

After each set of artificial Ants is run and a set of power losses is obtained, the pheromone probability matrix is updated while also taking into account the evaporation matrix.

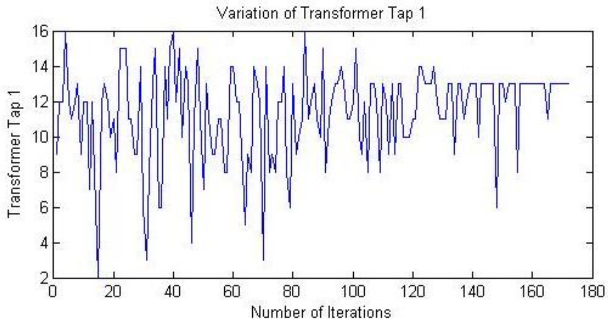
```
pa=.999*pa;       %Evaporation Rate for Pheromone A
pb=.998*pb;       %Evaporation Rate for Pheromone B
pc=.998*pc;       %Evaporation Rate for Pheromone C
cp;
if q~=1           %Global Updation Loop
pa(s(1))=pa(s(1))+.1*cp*f; %Updating Pheromone Level of A
pb(s(2),s(1))=pb(s(2),s(1))+.40*cp*f; %Updating Pheromone Level of B
pc(s(3),s(2))=pc(s(3),s(2))+.35*cp*f; %Updating Phermone Level of C
if f==0 %Updating Pheromone Levels when previous set minimum value again comes as a result

pa(s(1))=pa(s(1))+.1*cp*.0007;
   pb(s(2),s(1))=pb(s(2),s(1))+.25*cp*.0007;
   pc(s(3),s(2))=pc(s(3),s(2))+.28*cp*.0007;
end
```

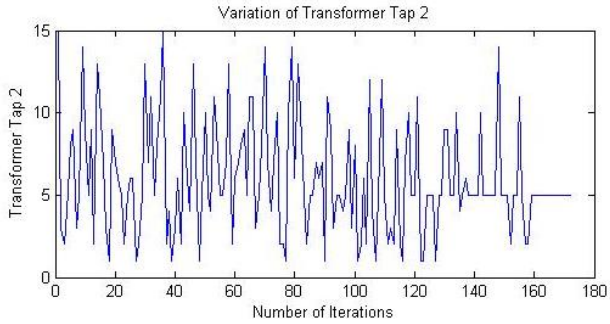
Similar to the constraints which are put on the various parameters of the Load Flow System in case of a fall of pheromone probability levels below a minimum value or a rise above maximum values the matrix components are reset to the standard values.

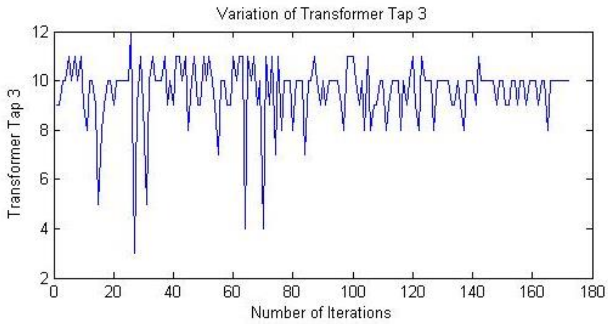
```
for h=1:al
   if pa(h)<=.01   %If Pheromone A falls below a minimum value
      pa(h)=.01;
   end
   if pa(h)>=500  %If Pheromone A exceeds a maximum value
      pa(h)=500;
   end
```

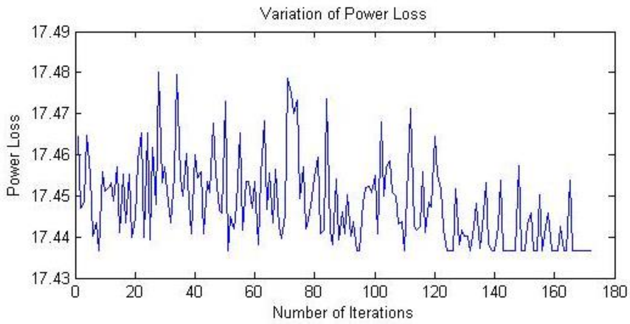
VI. RESULTS

The Minimum Power Loss obtained for the IEEE 30 Bus System was 17.4367 pu and it corresponded to a tap transformer setting of 1.03 pu on Transformer A, 0.95 pu on Transformer B and 1.00 pu on Transformer C. The process took an average of 174 iterations when run 10 times as compared to 4096 iterations which it would have taken had we applied conventional problem solving algorithms.









VII. CONCLUSION

The procedure implemented gives the minimum power losses combination within an error of 0.05% Power Loss. However we have only considered 3 Tap transformers with 16 discrete levels each. If the number of tap transformers is increased along with more variable elements like Variable Capacitors, variable Generator Buses etc, a greater degree of efficiency is possible. The ACO algorithm enables us to find the optimum solution to the CLF problem in just 174 iterations whereas by conventional methods the Newton Raphson Load Flow analysis would have to be run for the stipulated 4096 times.However a major flaw with the ACO Algorithm for CLF .

The main problem is the pheromone updation function. The function constants are highly dependent on the system to which it is applied and hence direct application to any system is not entirely possible as it leads to an increase in the number of iterations and reduction in accuracy.

REFERENCES

[1] Ant Colony System-Based Algorithm for Constrained Load Flow Problem John G. Vlachogiannis, Nikos D. Hatziargyriou, Senior Member, IEEE, and Kwang Y. Lee, Fellow, IEEE
[2] Ant Colony Optimization Algorithm Nada M. A. Al Salami
[3] M. Dorigo, luca M. G., " Ant Colony system: A Cooperative learning approach to the Travelling Salesman Problem", IEEE transaction on evolutionary computation, Vol. 1, No. 1, 1997.
[4] M. Dorigo and G. Di Caro, "The Ant Colony Optimization metaheuristic", in New Ideas in Optimization, D. Corne et al., Eds., McGraw Hill, London, UK, pp. 11-32, 1999.
[5] Solving Travelling Salesman Problem using Ant Colony Ivan Brezina Jr. Zuzana Čičková
[6] I.-K. Yu, C. S. Chou, and Y. H. Song, "Application of the ant colony search algorithm to short-term generation scheduling problem of thermal units," in Proc. Int. Conf. Power Syst. Technol., vol. 1, 1998, pp. 552–556.
[7] S.-J. Huang, "Enhancement of hydroelectric generation scheduling using ant colony system based optimization approaches," IEEE Trans. Energy Conver., vol. 16, no. 3, pp. 296–301, Sep. 2001.
[8] J.-H. Teng and Y.-H. Liu, "A novel ACS-based optimum switch relocation method," IEEE Trans. Power Syst., vol. 18, no. 1, pp. 113–120, Feb. 2003.
[9] Y.-J. Jeon, J.-C. Kim, S.-Y. Yun, and K. Y. Lee, "Application of ant colony algorithm for network reconfiguration in distribution systems," in Proc. IFAC Symp. Power Plants Power Syst. Control, Seoul, Korea, Jun. 9–12, 2003, pp. 266–271.
[10] W.Kellermann, H. M. Zein El-Din, C. E. Graham, and G. A. Maria, "Optimization of fixed tap transformer settings in bulk electricity systems," IEEE Trans. Power Syst., vol. 9, no. 3, pp. 1126–1132, Aug. 1991.
[11] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, Politecnico de Milano, Milan, Italy, 1992.
[12] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," IEEE Trans. Syst., Man, Cybern. B, vol. 26, no. 1, pp. 29–41, Feb. 1996.
[13] Phermone Models in Ant Colony Optimisation(ACO) E Foundas, A Vlochas Department of Informatics University of Piraeus Greece