Solving Unit Commitment Problem using Ant Colony Optimization (ACO) Technique

R. B. Baldaniya[1] H. D. Mehta[2]
[1]P.G. Student [2]Associate Professor
[1, 2]Electrical Engineering Department
[1, 2] L. D. College of Engineering, Ahmedabad, Gujarat

Abstract---In this paper, Unit Commitment (UC) problem for generation scheduling using Ant Colony Optimization (ACO) technique has been carried out. UC is the process of determining which generators should be operated each day to meet the daily demand of the system at minimum cost.ACO technique is inspired by nature of search for food by real ants from its source to the destination by building the unique trail formation. This approach has been applied to a system comprised of 3 generating units. A MATLAB code has been generated for solving ant colony based unit commitment problem. Comparative analysis of results using dynamic programming and Ant Colony Optimization technique are presented which clearly indicate that execution time required by ACO technique is much less as compared to dynamic programming

Keywords: Unit Commitment, Ant Colony Optimization, Dynamic programming.

I. Introduction

Nowadays electric power plays a very important role in human life. As population increases demand for power also increases. The operation and control of power system is very essential to meet the load demand. Various problems involved are Unit Commitment, Economic Dispatch and Load Forecasting. The unit commitment problem is a hard combinatorial mixed integer optimization problem to determine the optimum schedule of generating units while satisfying a set of system and unit constraints. Finding a good solution to the unit commitment problem in a reasonable time is very critical since it could mean significant annual financial savings in power generating costs. In essence, global optimum solution of such a combinatorial optimization problem can be obtained by examining all combinations.

Several solution techniques have been applied to this problem. Mainly they can be categorised into three - deterministic, Meta heuristic, and hybrid approaches. Deterministic approaches include priority list (PL), dynamic programming (DP), Lagrangian Relaxation (LR), integer mixed-integer programming, and the branch-and bound methods**.** The priority list is the simplest and fastest but achieves poor final solution. Dynamic programming techniques, essentially based on priority lists are flexible but the computation time suffers from the “curse of dimensionality”. Thesemethods have only been applied to small UC problems. Meta-heuristic approach involves genetic algorithm (GA), seeded memetic algorithm, evolutionary programming, simulated annealing (SA), and tabu search (TS). These methods can accommodate more complicated constraints and are claimed to have better quality of solution.

Because of the advancement of computational ability, there has been growing attention in algorithms inspired by the observation of natural phenomena to help solving complex combinatorial problems in the last decades. This paper uses the Ant Colony Search Algorithm, which was inspired by the observation of the behaviour of the real ant colonies, is applied to solve the UC problem.

The ant colony search algorithm was inspired by the behaviour of real ants, which are almost blind, but capable of finding the shortest path to the food form home. They can also find a new shortest path in case the old one is no longer available due to any type of obstacle. The medium, which they used to communicate information among individuals regarding paths, and used to decide where to go, consists of pheromone trails. A moving ant lays some amount of pheromone, on the ground, thus marking the path by a trail of this substance. Other following ants can identify it and decide with high probability to follow it, thus reinforcing the trail with its own pheromone. The collective behaviour that emerges is a form of *autocatalytic* (positive feedback) behaviour where the more the ants following a trail, the more attractive that trail becomes for being followed. The process is thus characterized by a positive feedback loop, where the probability with which an ant chooses a path increases with the number of ants that chose the same path in the preceding steps. They have been applied to many complicated function optimization problems such as the traveling salesman problemand the sequential ordering problem.

The ACSA used in this paper uses artificial ants (or *agents),* and lives in an environment where time is discrete. The state transition, global and local updating rules are also introduced to ensure the optimality of the solution.

II. Background Of Aco

Ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. This algorithm is a member of ant colony algorithms family, in swarm intelligence methods, and it constitutes some meta heuristic optimizations. Initially proposed by Marco Dorigo in 1992 the first algorithm was aiming to search for an optimal path in a graph, based on the behaviour of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behaviour of ants. The original idea comes from observing real ants, which are able to find the shortest path between a food source and the nest.

A. The process can be described as under:

1. The first ant finds the food source, via a random way, then returns to the nest, leaving behind a trail pheromone.
2. Ants follow all possible ways, but the strengthening of the runway makes it more attractive as the shortest route.
3. Ants take the shortest route, as long portions of other ways lose their trail pheromones rapidly.

1) ACO

An ant colony optimization (ACO) is a population based heuristic algorithm on agents that reproduce the natural behaviour of ants developing mechanisms of cooperation and learning which enables the searching of the positive feedback as a search mechanism.

2) Biological ant colony system

In a series of experiments on a colony of ants with a choice between two unequal length paths leading to a source of food, biologists have observed that ants tended to use the shortest route. A model explaining this behaviour is as follows:

1. An ant runs at random direction around the colony;
2. If it discovers a food source, it returns directly to the nest, leaving in its path a trail of pheromone;
3. These pheromones are attractive, nearby ants will be inclined to follow the track;
4. By returning to the colony, these ants will strengthen the route;
5. If there are multiple routes to reach the same food source then, the shorter one will be travelled by more ants than the long route;
6. The short route will be increasingly enhanced, and therefore become more attractive;
7. The long route will eventually disappear because of pheromone evaporation;
8. Eventually, all the ants have determined and therefore "chosen" the shortest route.

Ants use the environment as a medium of communication. They exchange information indirectly by depositing pheromones.

3) Artificial ant colony system

An Artificial Ant Colony System (AACS) is a population based heuristic algorithm on agents that simulate the natural behaviour of ants developing mechanisms of cooperation and learning which enables the exploration of the positive feedback between agents as a search mechanism. In AACS, the use of: (*i*) a colony of cooperating individuals, (*ii*) an artificial pheromone trail for local communication, (*iii*) a sequence of local moves to find shortest paths, and (*iv*) a stochastic decision policy using local information are the same as real ACS.

B. But artificial ants have also some characteristics which do not find their counterpart in real ants. They are

1. Artificial ants live in a discrete world and their moves consist of transitions from discrete states to discrete states.
2. Artificial ants have an internal state. This private state contains the memory of the ant's past actions.
3. Artificial ants deposit an amount of pheromone, which is a function of the quality of the solution found.
4. Artificial ants timing in pheromone laying is problem dependent and often does not reflect real ant's behaviour. For example, in many cases artificial ants update pheromone trails only after having generated a solution.

C. Essentially, an ACS algorithm performs a loop applying two basic procedures:

1. A procedure specifying how ants construct or modify a solution for the problem in hand;
2. A procedure for updating the pheromone trail.

The modification or construction of a solution is performed in a probabilistic way. The pheromone trails are updated considering the evaporation rate and the quality of the current solution.

III. PROBLEM FORMULATION

The objective of unit commitment problem is to minimize the total production cost and also computation time.

Production cost = minimum (Fuel cost + Start-up cost + shut-down cost)

A. Fuel cost

For a N number of committed units at any time, the minimum total fuel cost is,

$$\text{Min F } = \sum_{i=1}^{N} FC_i(P_i) \tag{1}$$

$$FC_i(P_i) = a_i p_i^2 + b_i p_i + c_i \ \ \$/\text{hr} \tag{2}$$

where,

a_i, b_i *and* c_i arethe cost coefficients.

p_i is the power generation of unit *i* (MW)

The total fuel cost is subjected to constraints

1) System power balance constraint

$$\sum_{i=1}^{N} P_i = Load\ (H) \tag{3}$$

Where $Load\ (H)$ is the system load at hour H.

2) Generation limit constraint

$$P_{i\,min}(t) < P_i(t) < P_{i\,max}(t) \tag{4}$$

Where $P_{i\,min}(t)$ and $P_{i\,max}(t)$ are minimum and maximum generation limit of generating unit *i.*

3) Spinning reserve constraint

Spinning reserve is the total amount of generation available from all units synchronized (i.e. spinning) on the system, minus the present load and losses being incurred. Spinning reserve must be carried out in such a way that the loss of one or more units does not cause too far a drop in system frequency.

$$\sum_{i=1}^{N} P_{i\,max}\,(t) \geq D(t) + R(t) \tag{5}$$

Where *D(t)* and *R(t)* are the total load and losses of system at time *t.*

4) Minimum up time constraint

Once the unit is running, it cannot be turned off immediately.

$$TiON \geq \text{MUi} \tag{6}$$

5) Minimum down time constraint

Once the unit is decommitted, there is a minimum time before it can be recommitted.

$$TiOFF \geq \text{MDi} \tag{7}$$

IV. IMPLEMENTATION OF THE ACO MODEL

Once the search space is identified, the ants are allowed to pass continuously through the ant search space. Each ant starts its journey from the initial condition termed as the starting node and reaches the end stage and finally vanishes.

So it's a continuous flow of ants. Once an ant reaches the end stage, a tour is completed and it calculates the overall generation cost path. This process is continued until the ants find an optimal solution.

The ant colony search mechanism can be divided into a) initialization, b) transition rule, and c) pheromone trail update rule.

A. Initialization

During initialization the parameters such as the requisite number of ants, the relative importance of the pheromone trail, relative importance of the visibility, initial available pheromone trail, aconstant related to the quantity of the trail laid by ants, evaporation factor, tuning factor etc… have to be fixed and taken care.

B. Transition rule

The transition probability for the k^{th} ant from one state i to next state j for an ACO model is given by

$$P_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_k[\tau_{ik}(t)]^{\alpha}[\eta_{ik}]^{\beta}} & if\ j \epsilon allowed_k \\ 0 & otherwise \end{cases} \quad (8)$$

Where,

τ_{ij} is the trail intensity on edge (i, j),

$\eta_{ij} = (1/C_{ij})$ called heuristic function, C_{ij} is the production cost occurred for that particular stage,

α is the relative importance of the trail, $\alpha \geq 0$

β is the relative importance of the visibility, $\beta \geq 0$

'$allowed_k$' is the available states k^{th} ant can choose from i^{th} state to j^{th} state.

C. Pheromone trail update rule.

1) Local Updating Rule

During the establishment of its tour, an agent changes the amount of pheromone level on the visited path by applying the local updating rule,

$$\tau_{ij} = (1 - \varrho)\tau_{ij} + \tau_0 \quad (9)$$

Where,

ϱ is the heuristically defined parameter,

τ_0 is the initial pheromone level.

The local updating rule is intended to shuffle the search process. Therefore the desirability of paths can be dynamically changed. Every time an agent uses a path it becomes slightly less desirable, since it losses some of its pheromone. This allows agents to make a better use of pheromone information.

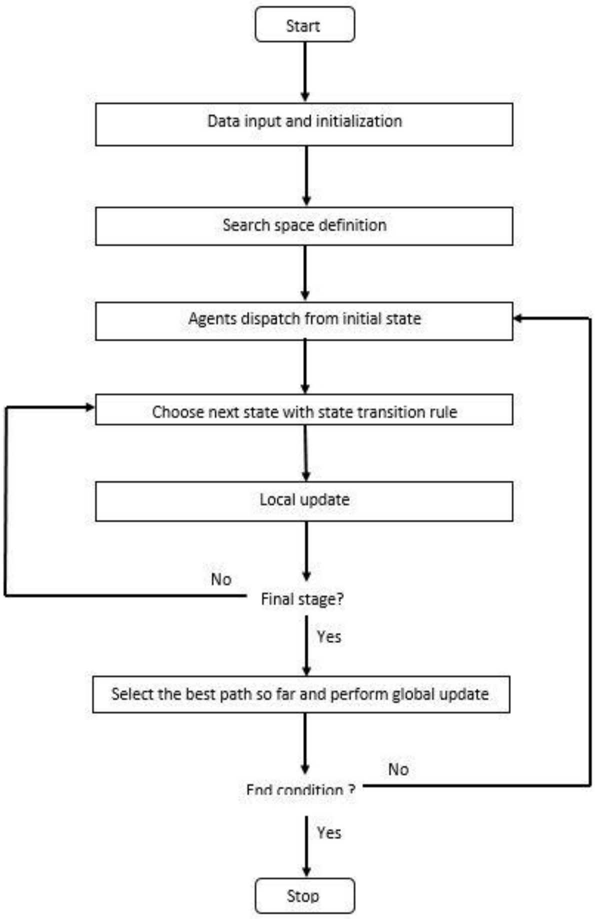
2) Global Updating Rule

When all the agents have built their individual solution the global pheromone-updating rule is applied only to paths that belong to the best agent tour. In other words, the pheromone-updating rules are designed so that they tend to give more pheromone to paths that are visited by more agents. The pheromone level is updated by applying the global updating rule

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij} \quad (10)$$

Where,

ρ is the pheromone decay factor ($0 < \rho < 1$).



V. CASE STUDY

The unit commitment problem has been solved by ACO technique on a 3-unit system considering a time horizon of 24 hours. System data is given in table I, control parameters used are shown in table II and results are shown in the table III. In table I, columns 4-6 are the coefficients of units' cost function $FC_i(P_i) = a_i p_i^2 + b_i p_i + c_i$.

Generator number	P_i^{min} (MW)	P_i^{max} (MW)	a_i ($/$MW^2$hr)	b_i ($/MWhr)	c_i ($/hr)
G_1	10	160	0.005	2.450	105.00
G_2	20	80	0.005	3.510	44.1
G_3	20	50	0.005	3.890	40.6

Table. 1: Three Units System Data

Parameter	**value**
Total No. of layers	3
No. of nodes in each layer (x)	Layer 1 – 152 Layer 2 – 62 Layer 3 – 32
No. of ants (N)	300
Degree of importance of the pheromones (α)	1
evaporation rate (or the pheromone decay factor) (ϱ)	0.5
parameter used to control the scale of the global updating of the pheromone (ρ)	0.5

Table. 2 : Control Parameters Used in ACO Process

Hour	Load (MW)	Total cost ($)	Execution time (sec)	
			For ACO	For DP
1	20	115.3	0.01898	0.05549
2	35	170.01	0.07637	0.09936
3	40	188.5	0.11722	0.13494
4	65	274.81	0.10932	0.41145
5	75	302.21	0.08394	0.57988
6	90	345.74	0.09801	0.94259
7	115	419.81	0.02371	1.57531
8	140	497.00	0.02105	2.91463
9	170	640.45	0.02309	4.87537
10	185	694.41	0.02178	6.19898
11	220	824.7	0.08248	9.40716
12	160	561.0	0.02290	4.19538
13	130	465.75	0.01999	2.42884
14	100	375.0	0.01788	1.25774
15	85	331.31	0.01622	0.73753
16	230	863.05	0.03023	10.58124
17	255	1001.41	0.02580	13.90605
18	260	1021.3	0.03740	12.39045
19	290	1143.25	0.51596	20.42929
20	220	824.7	0.02440	9.18367
21	180	676.3	0.02162	5.65087
22	120	435.0	0.01724	1.74082
23	70	288.75	0.02085	0.48923
24	65	274.81	0.02689	0.40254

Table. 3: Comparison of Results of UC Solution

VI. CONCLUSION

It can be seen from the table III that the execution time for dynamic programming method increases as the number of units and load increase. However in ACO technique the execution time is nearly constant and is not dependent on the magnitude of load. This advantage indicates that this method gives satisfactory results when applied to a large scale UC problem.

REFERENCES

[1] A. J. Wood And B. F. Wollenberg, "Power Generation Operation And Control," 2"D Ed., John Wiley And Sons, New York, 1996.

[2] D. P. Kothari And I. J. Nagrath, "Modern Power System Analysis", 4th Edition, Tata Mcgraw Hill, New Delhi , 2011, Pp. 262-268, 687-688.

[3] N.P. Padhy, "Artificial Intelligence And Intelligent Systems", 1st Edition, Oxford University Press, New Delhi, 2005, Pp. 536-544.

[4] Singiresu S. Rao, "Engineering Optimization, Theory and Practice", 4^{th} Edition, John Willey & Sons, Inc., Hoboken, New Jersey, 2009, Pp-714-722.

[5] N. S. Sisworahardjo And A. A. El-Keib, "Unit Commitment Using The Ant Colony Search Algorithm," Proceedings Of The Ieee/Lescope2002, Halifax, Nova Scotia, June 2002.

[6] S. P. Simon, N. P. Padhy, R. S. Anand, "An Ant Colony System Approach For Unit Commitment Problem", Journal Of Electrical Power And Energy Systems, Vol. 28, Pp. 315-323, 2006.

[7] Z. Ouyang, S.M. Shahidehpour, "An Intelligent Dynamic Programming For Unit Commitment Application", IEEE Trans. Power Systems, Vol. 6, No. 3, Pp. 1203-1209, 1999.

[8] M. Y. El-Sharkh, N. S. Sisworahardjo, A. Rahman And M. S. Alam, "An Improved Ant Colony Search Algorithm Forunit Commitment Application", Ieee 2006.

[9] Songsa kchusanapiputt, Dulytatnualhong Sujate Jantarangand Sukum vitphoom vuthisarn, "Unit Commitment By Selective Self-Adaptive Aco With Relatively Pheromone Updating Approach", The8th International Power Engineering Conference (Ipec 2007).

[10] Yuyang Lai, Yan Bai And Ying Fan, "Optimization Of Unit Commitment Based On Improved Max-Min Ant System Algorithm", Ieee 2012.

[11] Se-Hwan Jang, Jae Hyungroh, Wook Kim, Tenzi Sherpa, Jin-Ho Kim And Jong-Bae Park, "A Novel Binary Ant Colony Optimization: Application To The Unit Commitment Problem Of Power Systems", Journal Of Electrical Engineering & Technology Vol. 6, No. 2, Pp.174~181, 2011 Doi: 10.5370/Jeet.2011.6.2. 174