International Journal of Neural Networks and Advanced Applications
DOI: 10.46300/91016.2022.9.1 Volume 9, 2022

Neural Tau (Neuτ): Hinge-pin of Neural Augmented Ant Colony Optimization (NaACO) Technique

[1]Muhammad Umer, [2]Riaz Ahmad, [3]Amir Farhan
[1]School of Manufacturing and Mechanical Engineering (SMME), National University of Science and Technology (NUST), Islamabad, Pakistan
[2]School of Manufacturing and Mechanical Engineering (SMME), National University of Science and Technology (NUST), Islamabad, Pakistan
[3]Muhammad Ali Jinnah University (MAJU), Islamabad, Pakistan

Abstract—In this paper we introduce and elaborate on the concept and application of Neural Tau (neuτ) which has become the hinge pin for the formulation of Neural Augmented Ant Colony Optimization (NaACO) in our previous research and publications. The concept of neuτ has been discussed in detail and it is compared with the concept of traditional or conventional pheromone upgradation function. The application of this entity through NaACO in various worker assignment and scheduling problems along with the areas of future research has been depicted.

Keywords—Worker Assignment; Neural Augmented Ant Colony Optimization (NaACO); Artificial Neural Networks (ANN), Neural Tau (neuτ).

I. LITERATURE REVIEW

THE concept of Pheromone up-gradation in Ant ColonyOptimization (ACO) is to enable this meta- heuristic to forget the already learnt solution and enable ACO for global optima. The function of evaporation or reinforcement is thus developed and is elaborated in detail by Marco Dorigo and Thomas Stutzle [1] in their pioneering research on ant colonies and its mathematical formulation. The pheromone evaporation in real ants is not pivotal in finding the shortest path or trail for them once they are in search of food. In contrary to this once we talk about artificial ants, it is far more significant as the artificial ants have to deal with far more complex paths and problem sets as compared with the real ants. The conventional pheromone update (τ) is given as:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} \qquad (1)$$

Where ρ is a parameter ranging from 0 to 1. As per Dorigo and Stutzle if a single ant is taken and the "*Pheromone Depositing*" is done only on a forward or backward path, the result is that the ant colony is unable to choose the shortest branch. This impediment has to be catered for once dealing with a problem set in which inadvertently we have to use one ant to converge to the food source such as in the most common scheduling problems (problem discussed in preceding sections). As such the nature and function of the up-gradation mechanism changes and the formulation of τ has to adjust. As such τ is no longer required to upgrade but the corrective nature of τ can be used to decide upon the "*Combinational Correctness"*of the whole solution set.

This capability is enforced in the conventional τ through the introduction of Artificial Neural Networks (ANNs) in the up-gradation or the corrective loop while implementing ACO. Hence we introduce **neuτ**which has become the hinge pin for our implementation of NaACO [2].

NP-complete optimization problems were the hardest or more exhaustive [3]. Branch and Bound algorithms were first used to tackle these problems. The use of operations research techniques such as dynamic programming or integer linear programming were also used to solve the basic scheduling problems. Mathirajan et al. [4] have designed heuristic methods to tackle optimization related scheduling problems.

"Intelligent scheduling" from "systems approach" can be generated in a more better and compact way through the artificial intelligence approach to scheduling problems. The basic notion of the systems approach is "Divide and Conquer", as it "conquers" the problem through amalgamation of achieved results for parts by "dividing" the problem into more realistic and manageable domains. The whole idea of implementing the AI approach to any scheduling problem is to identify the agents that are "adaptive" yet "intelligent.

The development of ACO can be associated with the theory of stigmergy by Grasse in 1959 [5]. The optimization problems were tackled by ACO through the movement of artificial ants. The most successful application of ACO was on the travelling salesman problem (TSP). In 1992, Dorigo [6] presented the first algorithm. Various other variants of ACO were introduced by Dorigo and Thomson [1]such as Ant System (AS), MIN-MAX ant systems ATNS-OAP etc. ACO has proven to be more efficient than its counterparts as shown by single machine tardiness problem [7].Likewise, flow shop optimization was also done through ACO[8]. The resource constraint problem was done by Merkle et al. [9]. A non-delay scheduling sequence was tackled by local search byBlum and Sampels[10].

Neural networks have two functions which can be used very potently to formulate a τ which is capable of learning the pattern and then participating in formulation of a Fitness Threshold for the analysis of the whole scheduling problem. The neural networks are used to memorize the information related to the problem or to remember the inputs to the problem [11]. Secondly, ANNs are also used for the optimization in a given set of constraints and objective functions.[12], [13]. The capability of self-organization, convergence through unsupervised learning, generalization of a constraint problem, fault-tolerance are further advantages not typically found in "classical" approaches. The ANN thus constructed learns the patterns based on the inputs of the scheduling problem, and the outputs. The combination of ANN with ACO for the solution of scheduling problems which are constrained by the fact that the artificial ants can

only traverse in the forward direction in the first iteration, and thus this represents a hybrid approach to optimization problems. ANNs can be utilized to handle the variable of "resource" in the scheduling, in order to come up with a form of intelligent learning system or framework. Moreover, as already discussed this paradigm applies to the combinational analysis approach in scheduling problems and redefining the role of τ in the ACO approach.

II. PROBLEM FORMULATION

We have to schedule *n* jobs to m parallel machines subjects to the following conditions:

a. Each machine is capable of handling one job at a time.

b. There are three times A, B and E for each machine.

c. No job hopping is allowed.

d. Each job has to pass through a sequence of machines.

e. We have to allocate n jobs by the application of ACO and use neuτ to check for the combinational aspects of the problem through ANN.

f. The job can go to any machine first.

g. The setup times of machines are not catered for, so we are not focusing on minimizing the setup slacks.

h. Multitasking of machines is not allowed.

i. We are not accounting the transportation time taken between the machines.

j. For a start we are not focusing on adding resources in the form of machines or adding jobs to the initial problem.

k. The jobs allocation involves the calculation of the total time spent by each job on each machine.

l. At the start all machines and jobs are available.

m. The time for each machine is then calculated as;

$$P_i = A + B/E \ldots\ldots \quad (2)$$

Where the processing time of job *i* is given by *P* and A, B and E are respective times allocated to each machine. A set of 100 problems(figure #1) are selected each with three machines and twelve jobs. The following table represents a typical problem set:

The pseudo code for this problem is as under:

Part-I (Scheduling of Jobs)

Set n as the set of scheduled jobs for m machines. Initialize allocation of jobs through triggering heuristic.

While first loop is running apply heuristic information;

Every job has to be allocated to a machine.

Local search is to be applied.

Repeat for (n=1, 2, 3… 12);

All schedulable jobs are to be identified and partial solution is to be constructed.

Assign jobs to machines (m=1,2,3) do

end for

Part-II (Combinational Fitness)-neuτ

Train ANN for machine inputs (m=1, 2, 3) with τ values.

Display: machine times and number of workers proposed;

If YES end;

If NO do loop.

	Machine 1			Machine 2			Machine 3		
Job	A_i	B_i	E_i	A_i	B_i	E_i	A_i	B_i	E_i
1	7	556	8	8	227	1	4	328	3
2	5	784	4	7	36	5	2	330	9
3	5	195	3	3	236	9	6	570	1
4	2	427	9	9	305	4	6	260	4
5	3	85	6	8	240	7	2	506	4
6	7	799	6	0	758	5	2	166	5
7	0	540	4	9	783	5	8	148	2
8	7	12	1	3	321	5	8	466	5
9	8	460	6	5	222	4	5	64	3
10	7	80	7	7	128	4	9	366	6
11	0	82	9	0	130	3	9	724	5
12	4	639	8	5	517	1	2	209	2

Figure # 1

A. Neural Augmented ACO Formulation andResults:

Step 1

The probability by which the ants choose their path towards the food source is given by:

$$P_{i,j,k} = [\tau_{i,j}]^{\propto}[_{i,j}]^{\beta} / \sum[\tau_{i,k}]^{\alpha}[_{i,k}]^{\beta} \quad (3)$$

In ACOη is the reciprocal of the heuristic function or the greedy algorithm to trigger the initial assignment. In case of job assignment to machines the heuristic is the machine time of each machine, such that the ant (job) first goes to the machine with the shortest processing time. As a result a sequence for each job is generated for the machines (figure #2).

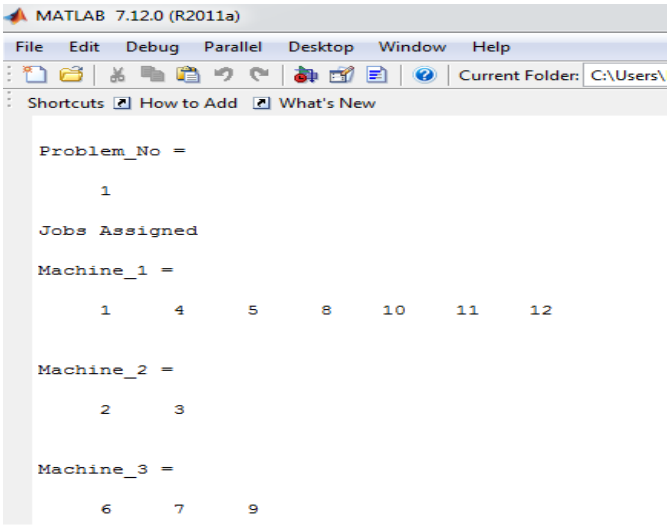

Figure #2

Step 2

A Feed Forward Neural Network is used to train the ANN in which the inputs of all 100 problems are linked with the outputs to come up with a trained ANN.Another feature is the inclusion of a YES/NO output parameter which is easy to comprehend and understand in pure industrial environments. The feed forward ANN comprises of The ANN is logic based (Yes/No) artificial neural network model, which clusters the combination of the three-machine times, initial job assigned and combines these entities to give and generate holistic optimum combination (neuτ). If the overall combination is correct the answer is "***Yes***", which means that the overall compatibility of all the variables is "recognized" and the value can be "***combinatorially used***".

Figure # 3 shows the training of ANN on the given set of 100 problems. The network is thus trained to give the NaACO approach.

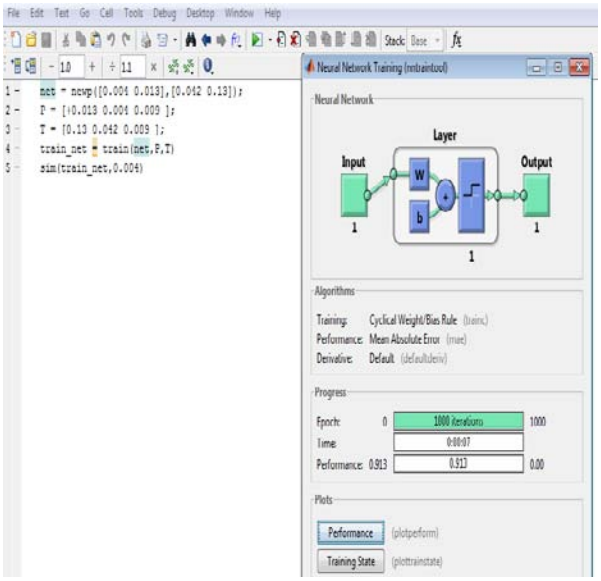

Figure # 3

This combinational optimization approach is then tested on various problem sets to check for the fitness of the combination with each other. The parameters involved include the times A, B and E on each machine, the no of jobs forecasted for that machine and the level of correctness recommended or being tested for (figure #4).

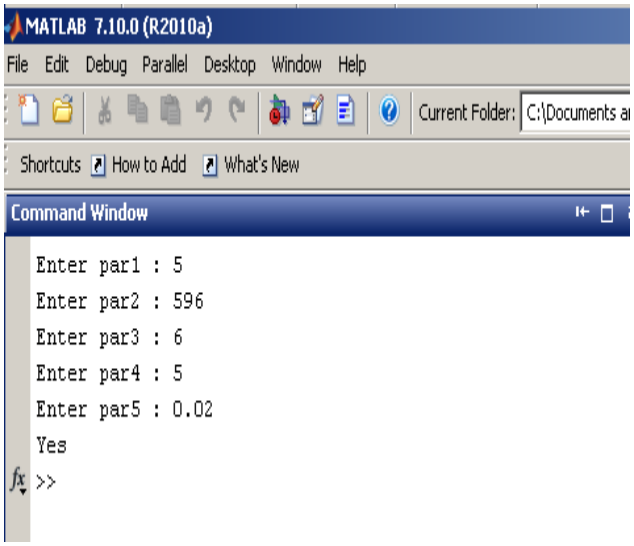

Figure #4

The overall corrective parameters of the problem set shall vary according to the complexity of the problem and according to the variables involved in the problem. ANN has to be trained to ascertain the tolerance levels or the upper and lower bounds of the problem set. In this respect this approach gives a customized methodology to a particular scheduling problem set.

III. AREAS OF FUTURE RESEARCH

This hybridized approach of NaACO has been applied and tested on a simple 3 work stations, 12 jobs environment. It is recommended to test this approach on more complex problems to check the robustness of the model and to recommend changes where necessary. Moreover, in the composition of this model the sensitivity factors of τ and η are taken as constant(i.e. α,β =1). In this respect a sensitivity analysis or post optimal analysis is the area of immediate future research for NaACO technique.

IV. CONCLUSION

The essence of this paper is to develop a neural based self-learning and evaluating mechanism within the basic framework of ACO. In this regards a simple scheduling problem is picked and the approach is tested to evaluate the results. The intelligent pheromone up-gradation mechanism is recommended through the inclusion of ANN in the pheromone update mechanism.

REFERENCES

[1] M. Dorigoand S.Thomas, "Ant Colony Optimization." London: The MIT press, 2004.

[2] M. Umer, Dr. R. Ahmad, Dr. I. Chaudhry, "Unsupervised Artificial Neural Networks (ANNs) For Intelligent Pheromone up Gradation. FurtherEvolution of Neural Augmented Ant Colony Optimization (NaACO)," *Life Science* 10 ed. vol. 3, 2013, pp. 318-327.

[3] M. R. Garey and D. S. Johson, *Computer and Intractability: A Guide to the theory of NP-Completeness*, 1979.

[4]M. Mathirajan,V. Chandru, A. I. Sivakumar, "Heuristic algorithms for scheduling heat-treatment furnaces of steel casting industries." *Sādhanā*32ed. vol. 5, 2007, pp. 479-500

[5] P. P. Grasse, "La reconstruction du nid et les coordinationsinterindividuelleslles chez. bellicositermesnatalensis et cubitermessp 6," 1959, pp. 41-81.

[6] M. Dorigo, "Optimization, Learning and Natural Algorithms." PhD thesis, Politecnico di Milano, Dipartimento di Eletronica, Milan.1992.

[7] M. L den Besten, T.Stutzle, and M. Dorigo, "Ant colony optimization for the total weighted tardiness problem,"*in Proc. of PPSN-VI, Sixth International Conference on Parallel Problem Solving from Nature*, Berlin: Springer-Verlag. 2000, pp. 611-620.

[8] C. Rajendra, and H. Ziegler, "Ant-colony algorithms for permutation flowshop scheduling to mini-mize makespan/total flow time of jobs." *European journal of Operational Reseacrh*, 2003.

[9] D. Merkle,M. Midderdorf, and H. Schmeck, "Ant colony optimization for resource-constrained project scheduling."*IEEE Transactions on Evolutionary Computation*6 ed. vol. 4, 2002, pp.333-346.

[10] C. Blum and M. Sampels, "Ant colony optimization for FOP shop scheduling: A case study on different pheromone representations,"*in Proc. of the 2002 Congress on Evolutionary Computation (CEC-02)*, Piscataway, NJ: IEEE Press, 2000a, pp-1558-1563.

[11] D. E. Rumelhart, and J. L. McClelland, *Parallel Distributed Processing*. Cambridge, MA: MIT Press.1986.

[12] J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems." *Biological Cybernetics*52 ed., 1985.

[13] M. O. Poliac, E. B. Lee, J. R. Slagle, and M. R. Wick, "A Crew Scheduling Problem."*in Proc.of IEEE International Conference on Neural Networks*, 1987, pp. 779-786.