Design and Development of Enhanced Optimization Techniques based on Ant Colony Systems

Annu K Simon
PG Student
Department of Computer Science & Engineering
NCERC, Thrissur, Kerala, India

Dr. S Subasree
Head of Dept.
Department of Computer Science & Engineering
NCERC, Thrissur, Kerala, India

Girish R
Associate Professor
Department of Computer Science & Engineering
NCERC, Thrissur, Kerala, India

Abstract

Many cyber defense techniques require time critical decisions and deployments. These decisions cannot be taken by a human alone; rather it requires a mixed initiative of human as well as some other digital agents. This project will make use of biologically inspired swarming agents called digital ants which make use of Ant Colony Optimization algorithm (ACO) for cyber defense. The system is rapidly and automatically adapts to new cyber-attacks while enabling humans to supervise the system at an appropriate level. Ant colony systems can provide approximate solutions for so many dynamic problems, but the performance of ant colony systems is reduced due to its slow convergence rate. This project will incorporate a physarum based matrix optimization strategy in Ant Colony Optimization algorithm, so as to enhance the convergence rate and leads to better performance in cyber defense techniques.
Keywords: Digital Agents, Ant Colony, Ant Colony Optimization (ACO), Physarum based Matrix Optimization

I. Introduction

Today cyber-defense systems involve humans at multiple levels of their infrastructure, but people are often far down in the control structure, so that they have to make too many time-critical decisions. Information flow between humans is slow and sometimes it may be frequently asynchronous. When a crisis occurs, humans may be unable to cooperate because of language, cultural, proprietary, legal, or other obstacles. Such systems cannot adapt to rapid cyber threats. Effective cyber defense entail a framework that concurrently exploit on the adaptability of humans and the speed of machines. Effective cyber defense requires mixed initiative hierarchical framework of humans and some other agents [1]. This paper presents a mixed-initiative hierarchical framework of humans and agents called digital ants for cyber security. Digital ants are working with the Ant Colony Optimization algorithm(ACO).There are different techniques in nature that can be adopted in computer science to solve different dynamic problems. Ant Colony Optimization (ACO) is one of the biologically inspired technique based on the behavior of real ant colonies. Physarum based ant colony system is another technique which make use of Physarum inspired Mathematical Model (PMM) integrated with the ant colony system. These algorithms are considered as the probabilistic technique for finding easiest optimal paths through a problem space. The ant colony optimization algorithms hold the behavior of natural ants. These algorithms make use of digital ants as agents to find an easiest solution for optimization problems by following the amount of phereomone trails produced by the ants to get to their destination in the shortest possible time.

In this paper we review and compare the efficiency of these algorithms based on the cyber defense architecture as well as Travelling Salesman Problem (TSP). These results can be used to find the efficient algorithm for finding anomalies in the systems.

II. Algorithms

A. *Formulation of a TSP*

The Travelling Salesman Problem can be stated as the problem of finding a shortest closed tour that visits each town exactly once [2]. TSP can be represented by a graph (N,E), where N =(1,2….,y) is the set of towns and E ={(i,j) i,j ε N, i ≠j} is the set of edges between towns. d_{ij} is the length of the path between towns i and j. Then, TSP solution algorithms are used to find the shortest closed path x, which is the optimal route. The length of x which is denoted as S_{min}, is shown in equation (1), where x_i represents the i^{th} city in the closed path x, and x_i ε N.

$$S_{min} = \min \left(\sum_{i=1}^{y-1} d_{x_i.x_{i+1}} + d_{x_y.x_1} \right) \quad (1)$$

B. Solving TSP with ACO

Ant System was first introduced and applied to TSP by Marco Dorigo [3]. Initially, each ant is randomly put on a city. During the construction of a solution, ants select the following city to be visited using a probabilistic decision rule. When an ant k situates in city i and constructs the partial solution, the probability for moving to the next city j neighboring on city i is given by equation (2)

$$P_{ij}^{k}(t) = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{u\, \varepsilon J_{k(i)}}[\tau_{iu}]^{\alpha}[\eta_{iu}]^{\beta}} \quad (2)$$

Where τ_{ij} is the intensity of trails between edge (i,j). The remaining cities to be visited when the ant in the city i is represented as $J_{k(i)}$. α and β are two adjustable positive parameters that control the influence of the pheromone trail and of the heuristic visibility respectively. $0 \leq \alpha$ is a parameter to control the influence of τ_{ij}. η_{ij} is the desirability of state transition i,j (typically $\frac{1}{d_{ij}}$, where d is the distance) and $\beta \geq 1$ is a parameter to control the influence of η_{ij}. After each ant completes its tour, the pheromone amount on each path will be adjusted with equation given in (3).

$$\tau_{ij}(t+1) \leftarrow [\ (1-\rho)\ \tau_{ij}\ (t) + \Delta\tau_{ij}\ (t) \quad (3)$$

In this equation,

$$\Delta\tau_{ij}\ (t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}\ (t) \quad (4)$$

$$\boxed{\Delta\tau_{ij}^{k}\ (t) = \begin{cases} \frac{Q}{L_k}, & \text{if (i,j) } \varepsilon \text{ tour by ant k} \\ 0 & \text{otherwise} \end{cases}} \quad (5)$$

(1- ρ) is the pheromone decay parameter ($0<\rho<1$) where it represents the pheromone content evaporation rate when the ant chooses a city and decide to move to next city. L_k is the length of the tour performed by ant k and m is the number of ants. The algorithm solving TSP with ACO is given below,

1) Algorithm 1 ACO for solving a TSP

Input d_{ij}: The distance between nodes i and j;
y: The number of node;
Output S_{min}: The length of the shortest Hamiltonian circuit;
Begin

1) Initializing parameters α, β , λ, q_0, m and Tsteps
2) Initializing the pheromone trail τ_0
3) Setting the iteration counter N := 0
4) While N < Tsteps Do
5) For k := 1 to m Do
6) Constructing a tour by an ant k
7) Updating the local pheromone matrix
8) End For
9) *best* := the global best ant
10) S_{min}:= the length of the tour generated by the ant *best*
11) N := N + 1
12) End While
13) Outputting the optimal solution S_{min}

End

C. Solving TSP with Physarum based Mathematical Model

In the Physarum based ant colony algorithm Yuxin Liu et.al proposes an optimization strategy for updating the global pheromone matrix in ACO for solving a TSP [4]. As shown in figure 1, we assume that there is a Physarum network with pheromone flows in tubes. In the Physarum network, when the flowing pheromone in a tube is high, then the tube becomes wide. After each iteration of ant colony system, the amount of pheromone in each tube can be calculated. When updating the global pheromone matrix, Physarum based ant colony system considers both the pheromone released by ants and the flowing pheromone in the Physarum network. Since the critical tubes in the Physarum network are also the shortest paths that have a larger amount of flowing pheromone, they have higher probability to be selected for ants travelling[5]. This technique is adopted with ant colony algorithm for finding the shortest path. The meaning of each parameter in Physarum based ant colony system and the calculation process of the amount of flowing pheromone in the Physarum network are defined in the following.

$$\tau_{ij} \leftarrow [\ (1-\rho)\ \tau_{ij} + \frac{p}{S_{global-best}}\ ] + \varepsilon \frac{\rho \times Q_{ij} \times M}{I_0} \quad (6)$$
$$\forall_{\ (i,j)} \in S_{global-best}$$

$$\varepsilon = 1 - \frac{1}{1+\lambda^{\frac{P_{steps}}{2}-(t+1)}} \tag{7}$$

In (6), ε is defined as an impact factor to measure the effect of flowing pheromone in the Physarum network on the final pheromone matrix. As shown in (7), Psteps stands for the total steps of iteration affected by Physrum network, t is the steps of iteration at present, and λ ε (1, 1.2). M represents the number of tubes in the Physarum network, which is equal to the number of roads in a TSP.

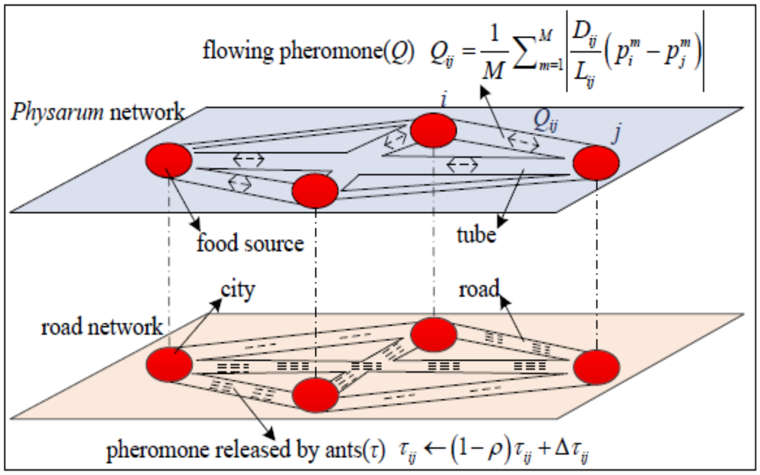

Fig. 1: Working mechanism of Physarum based ant colony system

I_0 represents the fixed flux flowing in the Physarum network. When two nodes i and j connected by the m^{th} tube are selected as inlet and outlet nodes, respectively, the pressure on each node p_i^m can be calculated according to the Kirchhoff Law based on (8)

$$\sum_i \frac{D_{ij}}{L_{ij}}(p_i^m - p_j^m) = \begin{cases} -I_0 & \text{for } i=a \\ I_0 & \text{for } j=b \\ \text{Otherwise } 0 & \end{cases} \tag{8}$$

Where L_{ij} is the length of a tube (i,j), and D_{ij} is defined as a measure of the conductivity, which is related to the thickness of the tube. The conductivity of each tube will be initialized before computation. The above process is repeated until all pairs of nodes in each tube are selected as inlet/outlet nodes once. Then, the flux Q_{ij} through the tube (i, j) is calculated based on the equation (9). Algorithm 2 represents the steps of the Physarum based ant colony system for solving TSP.

$$Q_{ij} = \frac{1}{M}\sum_{m=1}^{M} \left| \frac{D_{ij}}{L_{ij}}(p_i^m - p_j^m) \right. \tag{9}$$

1) Algorithm 2: Physrum based ant colony system for solving a TSP

Input d_{ij}: The distance between nodes i and j;
y: The number of node;
Output S_{min}: The length of the shortest Hamiltonian circuit;
Begin
1) Initializing parameters α, β , λ,q_0, ρ, I_0 , m, Psteps and Tsteps
2) Initializing the pheromone trail τ_0 and the conductivity of each tube D_{ij}
3) Setting the iteration counter N := 0
4) While N < Tsteps Do
5) For k := 1 to m Do
6) Constructing a tour by an ant k
7) Updating the local pheromone matrix
8) End For
9) *best* := the global best ant
10) S_{min} := the length of the tour generated by the ant *best*
11) Calculating the flowing pheromone in the Physarum network
12) $\tau_{ij} \leftarrow [\,(1-\rho)\,\tau_{ij} + \frac{p}{S_{global-best}}\,] + \varepsilon\frac{\rho \times Q_{ij} \times M}{I_0}$
13) N := N + 1
14) End While
15) Outputting the optimal solution S_{min}
End

III. SYSTEM ARCHITECTURE

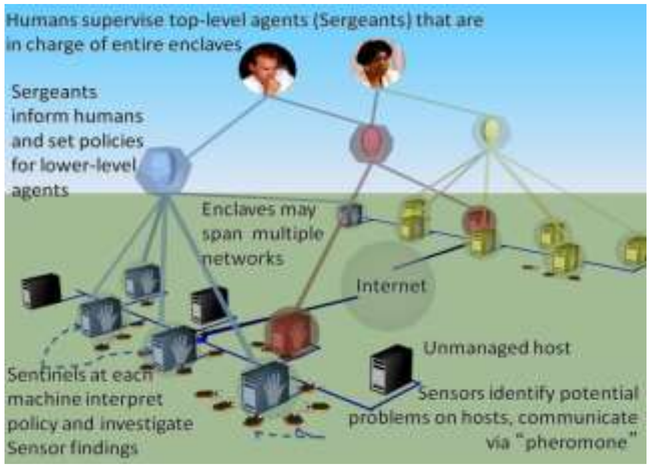

Fig. 2: System architecture of ant based cyber defense

Exemplar based Inpainting technique is used for inpainting of text regions, which takes structure synthesis and texture synthesis together. The inpainting is done in such a manner, that it fills the damaged region or holes in an image, with In the cyber defense architecture humans and various types of software agents share the responsibilities for securing a cyber infrastructure [6]. Figure 2 shows how one human can supervise a multi enclave system which includes some enclave-level agents or middle level agents called sergeants, a host-level agent at each machine or group of similar machines called as sentinels, and a group of simple mobile agents called as sensors.

The components in the system are given below,

A. *Supervisor Module*

Humans function as Supervisors. They provide guidance to multiple agents in the system as well as they monitor and take feedback from the lower level agents. They are known as the ultimate authority to the system. They take action only when the lower level agents encounter a problem that requires human involvement. However, direct human control of the system is discouraged, because such involvement would adversely affect its natural adaptive abilities.

B. *Sergeant Module*

Enclave-level agents or middle level agents are called Sergeants, are responsible for the security state of an entire enclave. Sergeants will take guidance from the supervisor for solving security issues. Using the guidance from the supervisor sergeants will create and enforce executable policies for the entire enclave. Sergeant will receive the values from the lower level agents such as sentinel. Sergeant will run the ACO or Physarum based ant colony algorithm with the received values from the sentinel. If any variation occurred from the threshold value then the system will alert a message. The message is the output from the Naive Bayes Classifier. If any large variation occurs and that variation is unknown to the sergeant machine then it will be forwarded to the supervisor, the ultimate authority.

C. *Sentinel Module*

Host-level agents, called Sentinels, protect and configure a single host or a collection of similarly configured hosts. The sensor values from the database will be evaluated by the sentinel. Sentinel will run the ACO or Physarum based ant colony algorithm with the listed value from the database. If any variation occurred from the threshold value, then the system will alert a message. The message is the output from the Naive Bayes Classifier. If any large variation occurs, then it will report to the sergeant system.

D. *Sensor Module*

Swarming mobile agents, called Sensors, roam from machine to machine within their enclave searching for the parameters given in their identity and reporting to the appropriate sentinel. The actual Sensor code resides on the Sentinel itself. Ants are simply messages that carry an ant's identity. Here we created three ants called cpu ant, memory ant, disk ant. Ants identity can be anything other than these cpu, disk and memory. These ants will be continuously checking the cpu, memory, disk variations in the system. These values will be written to a database.

IV. ANALYSIS

This section presents different kinds of experiments used for comparing the performance of Ant Colony Optimization (ACO) algorithm and Physarum based ant colony system based on TSP and cyber defense architecture.

The graph in figure 3 shows the comparison results of execution time v/s iteration count of TSP. From the graph we can observe that the Physarum based ant colony system has lower execution time than ACO. It is identified from the graph that the execution times are directly proportional to the variable and increase linearly along with the variable.

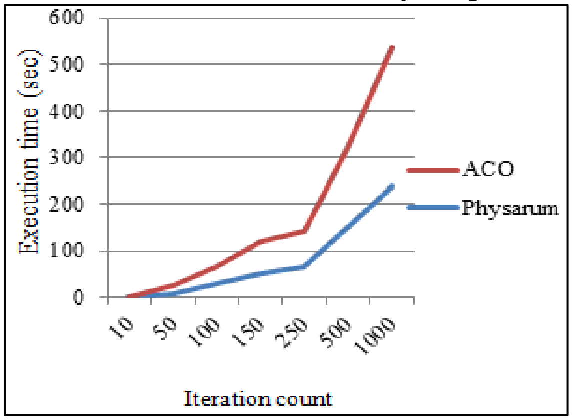

Fig. 3: Execution time (seconds) v/s iteration count analysis graph in TSP (Ant count is 10)

The following figures contains the cpu usage (fig: 4.a, 4.b) of cyber defense architecture with respect to Physarum and ACO algorithms. From these graphs we can get a clear idea that physarum algorithm is faster than ACO algorithm based on execution time. Figures 5.a, 5.b and 6.a, 6.b are the graphs of memory usage and disk usage respectively of the cyber defense architecture. These graphs also pointing that Physarum algorithm works better than ACO algorithm for the cyber defense.

The following figures contains the cpu usage (fig: 4.a, 4.b) of cyber defense architecture with respect to Physarum and ACO algorithms. From these graphs we can get a clear idea that physarum algorithm is faster than ACO algorithm based on execution time. Figures 5.a, 5.b and 6.a, 6.b are the graphs of memory usage and disk usage respectively of the cyber defense architecture. These graphs also pointing that Physarum algorithm works better than ACO algorithm for the cyber defense.

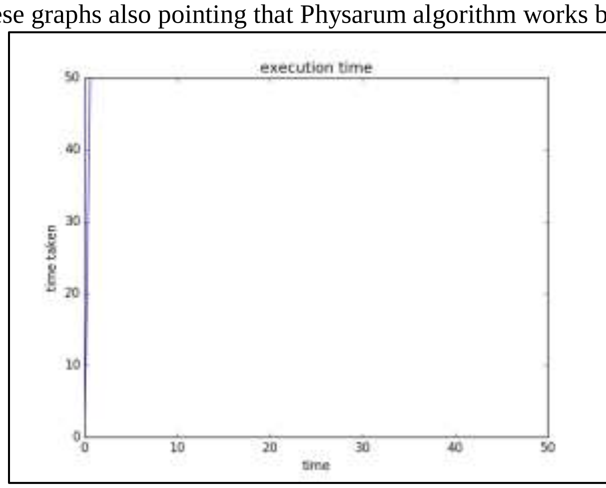

Fig. 4(a): Cpu usage execution time (seconds) using ACO Algorithm

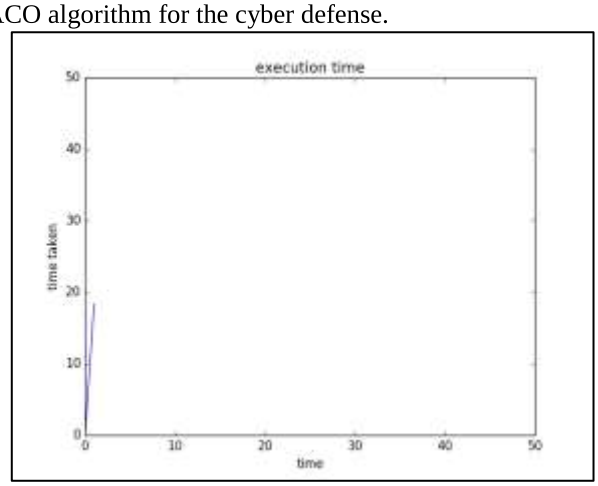

Fig. 4(b): Cpu usage using execution time (seconds)Physarum Algorithm

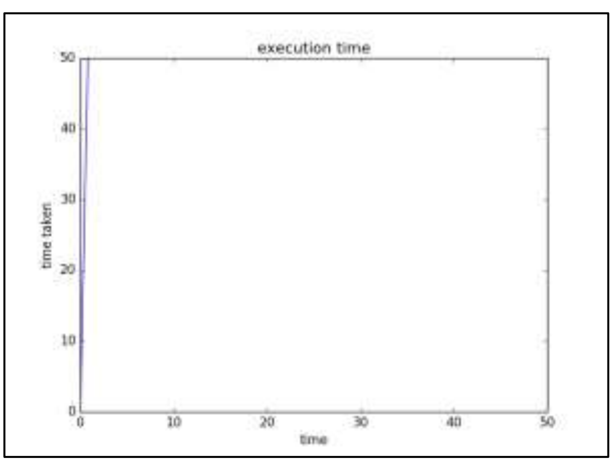

Fig. 5(a): Memory usage execution time (seconds) using ACO Algorithm

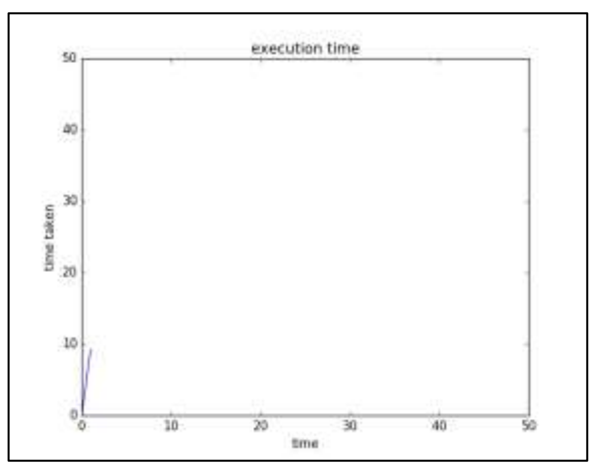

Fig. 5(b): Memory usage execution time (seconds) using Physarum Algorithm

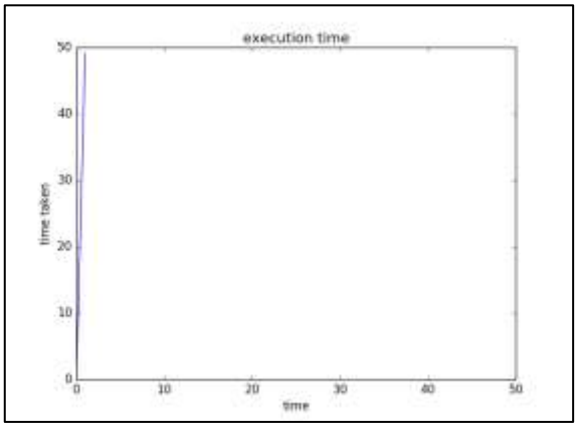

Fig. 6(a): Disk usage execution time (seconds) using ACO Algorithm

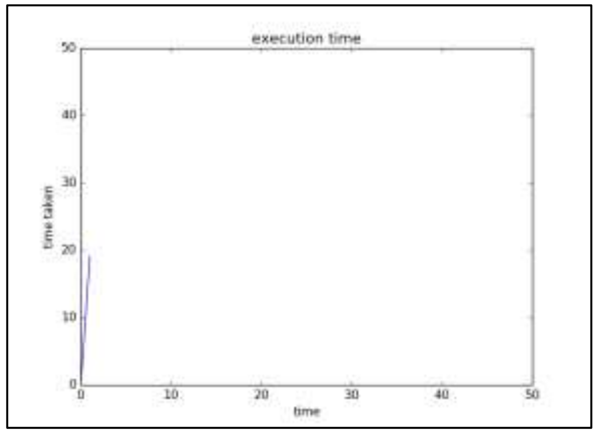

Fig. 6(b): Disk usage execution time (seconds) using Physarum Algorithm

V. Conclusion

Ant Colony Optimization (ACO) and Physarum based Ant Colony System are biologically inspired techniques used to solve dynamic problems. In this paper we presented a review and comparison of these two meta heuristic algorithms based on a cyber defense architecture and TSP. With the unique feature of critical tubes in the Physarum network, shows better result against ACO. The review suggests that if we incorporate physarum based mathematical model with ACO in cyber defense architecture then it will produce better results.

References

[1] J.N. Haack et al., "Ant-Based Cyber Security," 8th International Conference on Information Technology: New Generations (TNG11), pp. 918–926, April 2011
[2] S.A. Sheibat Alhamdy, A.N. Noudehi, and M. Majdara, "Solving traveling salesman problem (TSP) using ants colony (ACO) algorithm and comparing with tabu search, simulated annealing and genetic algorithm," J. Appl. Sci. Res., vol. 8, no. 1, pp.434-440, Jan. 2012.
[3] M. Dorigo and G.D. Caro, "Ant colony optimization: A new metaheuristic, "Proc. Congr. Evolutionary Computation (CEC'99), vol. 2, pp. 1470-1477, Jul. 1999.
[4] Z Yuxin Liu,et.al., "Solving NP-Hard Problems with Physarum-Based Ant Colony System", IEEE/ACM Transactions on Computational Biology and Bioinformatics, no. 1, pp.1,PrePrints, doi:10.1109/TCBB.2015.246234
[5] Z.L. Zhang, C. Gao, Y.X. Liu, and T. Qian,"A universal optimization strategy for ant colony optimization algorithms based on the Physarum-inspired mathematical model, "Bioinspir. Biomim., vol. 9, p. 036006, Mar. 2014.
[6] Fink, G.A. et al., "Defense on the move: Ant based cyber defense", IEEE Transactions on Security & Privacy, Volume:12, Issue: 2, pp 36 – 43 Mar.-Apr.2014