International Journal of Digital Information and Wireless Communications (IJDIWC) 5(4): 237-250
The Society of Digital Information and Wireless Communications, 2015 (ISSN: 2225-658X)

# Lyapunov Stability Analysis for Consequential Message Path in Graded Cognitive Networks

**Kavitha Sooda**[1,2,†], **T. R. Gopalakrishnan Nair**[3,4]

[1]Advanced Networking Research Group (RIIC), DSI
and
[2]Associate Professor, Department of CSE
Nitte Meenakshi Institute of Technology
Bangalore-560064, India.
E-mail: kavithasooda@gmail.com

[3]Saudi ARAMCO Endowed Chair – Technology, PMU, KSA
and
[4] Advanced Networking Research Group
VP, Research, Research and Industry Incubation Center (RIIC), Dayananda Sagar Institutions
Bangalore-560078, India.
E-mail: trgnair@ieee.org, www.trgnair.org

## ABSTRACT

In the domain of network routing, one of the key measure belongs to the stability analysis of the paths obtained by routing algorithms. Lyapunov analysis has been used widely for understanding the nature of the system output to know the stability intensity. Here in this article, we have compared the performance of two bio-inspired algorithms on a graded surface while determining the path. Also the study of the Lyapunov analysis is carried out on a restrcited search space. Here the Lyapunov function derived are best suited for the path derived on graded surface by Memetic and Artificial Bee Colony algorithms. Results show that the path obtained by the bio-inspired algorithms has the required stability as found by Lyapunov function for the transfer of data.

## KEYWORDS

Artificial Bee Colony, Bio-inspired algorithms, Graded network, Intelligent routing, Lyapunov functions, Memetic, Reduced search space.

## 1 INTRODUCTION

The quality of the selection of path in current network depends on parameters like Bandwidth and Delay for data-forwarding purpose. This would involve hello packets to be transmitted to get the information of the required Quality of Service (QoS) parameters the user is looking for. The key necessary factor in today's Internet is to ensure that the nodes have knowledge of its environment. This has become an essential parameter as inter-networking is evolving to complex structures. This formation of complex structure is due to dynamic nature of the environment. Diagnosing failure of forward channel is difficult in such an environment. Thus, we need to ensure knowledge availability in the environment. This can be made possible if the topology of our interest can learn and reason out how to handle forwarding of packets. The advantage of this knowledge would be in the quality selection of the path. The literature shows that many progressive works has been carried out in this direction. One of the prominent approaches that had proved its proximity in achieving this is the nature-inspired algorithm that has facilitated to learn and adapt based on topological formation and emergent conditions [1]. Intelligent networks [2] require an approach, whereby each node cooperates with the data-distribution process and makes use of an information-rich context. To get intelligent network working, there is a need to rethink on the architecture and protocols of the components in the global communication infrastructure. A prominent research direction orients its approach to mimic nature-like mechanisms to realize smarter communication networks that can make sense of the hidden communication patterns and do self-regulation within the topology for achieving intelligent routing. The routing approaches for intelligent routing are based on learning and reasoning techniques. This will require a database to store the learnt aspect, which can be referred to later in the future when necessary. The learnt aspect that will be utilized in future is based on the reasoning methods that will justify the result obtained.

This article presents the implementation of a novel idea of applying intelligence, which enables the nodes to make decisions and carry out the routing of packets. This intelligence at the node level makes the routing efficient and better [3]. A geographical search approximation is done using a cone, where we apply the routing algorithms on reduced search space and apply reasoning to derive the final path. Results show that the grading approach along with Artificial Bee Colony (ABC) and Memetic approaches has achieved the required routing path with promising speed [4-5]. The result thus obtained is verified by Lyapunov function to verify the paths obtained.

The rest of this article is organized as follows: in Section 2, related literature is described and in Section 3, the reasoning method is discussed in detail. Graded networks and Lyapunov analysis is dealt in Sections 4 and 5. Simulation results and analysis are presented in Section 6. Section 7 concludes.

## 2 RELATED WORK

The ability to be aware of network operations and subsequently adjust the operational parameters according to the needs of the scenario is referred to as cognitive network. It not only behaves based on a reference to an active network, but also includes an adaptation and learning technique [6-7], which makes the process different. Cognition was conceptualized by Mitola and later the idea of a feedback loop was derived from it. With the advancement in the field of cognition, the autonomic approach [8] has become more evident to solve a number of network-related issues. Some of the applications of cognitive approaches have been discussed below.

### 2.1 Autonomic network

An autonomic network is applicable in a situation where path determination is more predictable, and the search space can be handled well with an algorithm. When the environment is large, then cognitive approach will play a role in learning and reasoning. IBM is one of the organizations that proposed the autonomic networking concept. It described agents in three dimensions. The first dimension is the agency where it determines the degree of autonomy and authority assigned to the agent. The second dimension is about intelligence, describing the reasoning and learned behaviour. The third dimension is the mobility where the agents travel through the network. The current routing operates based on the packets received. This type of operation is asynchronous and is loosely coupled with a complex network. This satisfies the lowest degree of autonomy [9-10].

### 2.2 Motorola FOCALE

The autonomic management platform developed by Motorola is referred to as Foundation–Observation–Comparison–Action–Learn–rEason (FOCALE) for core networks [11]. This development was based on the human nervous system that performs unconscious actions. The system actually tries to figure out functions not known to the human. This system contains all the elements of cognitive approaches. It has two control loops, one for maintaining the current state and the other for reconfigurations. The idea behind this development was to consider the business-level requirement in simple language. It is later transferred to the autonomic configured network where it gets converted to a form that would sense and act according to the requirement of the end user.

### 2.3 Software-defined networking

The technique used in software defined networking (SDN) is based on a centralized controller that directly dictates where the packet should move [12-13]. There is support for multi-switch updates, which makes it strong without asking the user to program it with low-level commands. The user need not worry about the step-by-step updates. But programmers who use this must be able to distinguish among the multiple abstract updates that have been useful. This will definitely require a consistency class, mechanism and language to code for proper set-up of SDN.

## 3 IMPORTANCE OF REASONING

A cognitive network is a network with intelligent process that can perceive current network conditions, and then plan, decide and act on those conditions. The network can learn from these adaptations and use them to make future decisions, while taking into account end-to-end goals. The perception, decision, action and learning aspects of this definition are all elements of learning and reasoning. These capabilities are essential to ensure that the decisions made by a cognitive network will improve the network performance as measured against the end-to-end goals. Graded reasoning method is a promising approach for realizing the optimal path determination in the cognitive network. In this context, the estimation of grades at the nodes based on the reasoning technique has been a challenge for a long time.

This work deals with a novel estimation of reduced search space using a graded technique on cognitive network [14-15] using a reasoning mechanism. The transition from packet switching routing towards self-independent routing has been the research focus of the current decade. The realization of the self-independent nature of routing has been evolved through the cognitive approaches. Current routing schemes lack the ability to identify the relationship between the environment awareness and reasoning of the path determined by the technique. Thus, cognition plays a vital role in the process of remembering and formation of the path in Internet routing. The present article proposes and discusses in detail how an optimal path can be determined for graded cognitive network that has information about the environment [16-19].

### 3.1 Requirement for reasoning

Learning and reasoning play an important role in the cognitive network. Reasoning helps in the immediate decision process using the historical knowledge as well as the current state of the network. The primary goal of reasoning is to choose a set of actions. Learning is a long-term process that is based on the accumulation of knowledge on the perceived result of past actions. Cognitive network nodes learn by enriching the Information Base so that efficacy of reasoning improves in the future.

In general, any cognitive network structure will require some assistance to do a certain amount of job for accounting for an optimal solution. This is where agents will assist the network in doing the job. An agent [20-21] in general will work on a particular problem space, where it tries to achieve its goal and remember it for future purposes. To take proper action, it looks into what was learnt and also the current state to take a decision based on belief and goal.

The agent extracts the required parameters from Information Base and operates on the network topology as shown in Figure 1. The input to this network topology is the desired source and destination that comes as an instruction. The network topology carries out the reasoning to obtain the optimal path and to choose an action to be executed in the environment. The reasoning process is driven by the data structures in the Information Base.

In this article, once the awareness of the environment is learnt, and nodes have been graded and two algorithms have been used to determine the final path. First, ABC algorithm is used as developed by Karaboga [22], which was motivated by the intelligent behaviour of the bees.

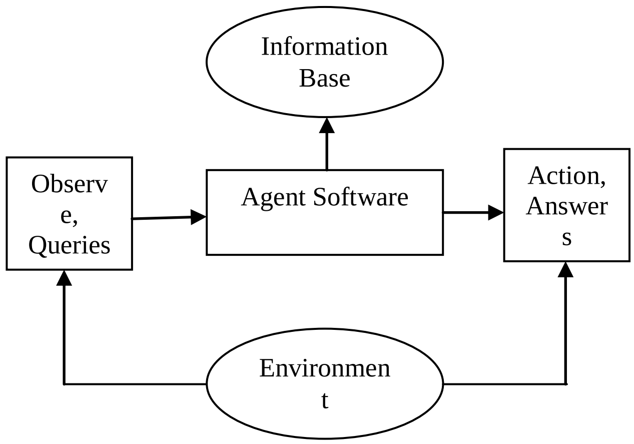


**Figure 1. Standard agent model**

It is an optimization tool, where search is performed in a distributed environment. Second, hybrid evolutionary algorithms that are commonly known as Memetic Algorithms (MAs) or genetic local search [23-25] are used. Such methods have been demonstrated to converge to high-quality solutions more efficiently than their conventional counterparts.

## 4 GRADING THE NETWORK

A grade is like an index, which represents the node QoS parameter-satisfied status. The status helps in estimating the selection of the node quickly once the network is realized. The grade technique has been applied segment wise. Here in each segment there exist thinking processes that run a performance function for determination of homogeneity among the nodes. This thinking process will have the awareness of the complete segment. Once the data have reached their end points in one segment, it will be handed over to the next segment for forwarding the data. Now the thinking process in this new segment will try to compare its evaluation of the nodes with the previous segment to see how the homogeneity property persists among the nodes, across the segments. This evaluation helps in the recognition of familiar pattern that would have been already encountered before. This qualifies for recognition ability from the learnt knowledge of the nodes. The parameters considered are based on bandwidth of the link, buffer resource allocation, delay and packet loss. Based on each of these contributing factors, the grade estimation is calculated based on the lumped parameter equation, which will assign a value to the nodes associated with the metric chosen. The nodes participating in routing are studied under various parameters and evaluated based on Equation 1. The grade value (GV) obtained by this equation for a given node is thus stored in the memory of the node for further reference.

GV = $a_1$ * contribution from B + $a_2$* contribution from De + $a_3$* contribution from RA + $a_4$ * contribution from PL, i.e.,

$$GV = 0.6 * B + 0.2 * De + 0.1 * RA + 0.1 * PL \quad (1)$$

where,

B = bandwidth of a link; De = delay;
RA= resource availability and PL = packet loss.

Higher GV indicates better satisfaction of the parameters considered. Figure 2 depicts a pictorial representation of the GV assigned on the nodes in the search space. A GV of ‘5’ is represented as red colour, GV of ‘4’ is represented as green, GV of ‘3’ is represented as blue, GV of ‘2’ is represented as black and GV of ‘1’ is represented as white colour.

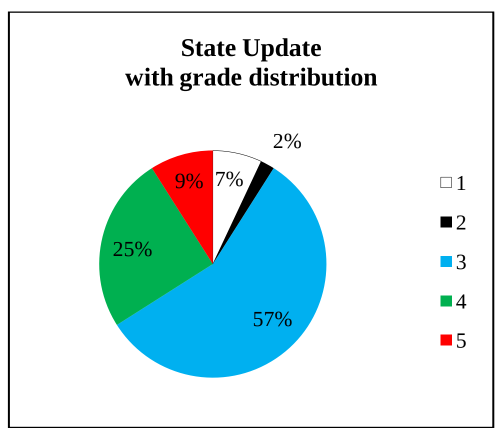


**Figure 2. Nodes grade index distribution in set-up**

The GV for a node is assigned based on the following criteria as depicted in Table 1.

## 5 LYAPUNOV STABILITY

Lyapunov stability analysis plays an important role in the stability analysis of control systems usually described by the state space equations for nonlinear systems. A Lyapunov function is that which allows one to deduce stability. There are two methods for stability analysis used by Lyapunov approach. First method is the Lyapunov Linearization that is applied for local stability in small range motion. This method fails to conclude about marginal stability. The second method overcomes this shortcoming and is used when it does not require the solutions of the differential equations for analysis. This method is called as the direct method.

**Table 1. Grade Value (GV)**

| Levels | GV |
|---|---|
| All conditions satisfied | 5 |
| Only delay exists | 4 |
| Resource alone not satisfied | 3 |
| PL alone not satisfied | 2 |
| No bandwidth available/two or more QoS not satisfied | 1 |
| None of the parameters satisfied | 0 |

Here, in this article, we develop the equations using the direct method. The design of the network topology is not explicitly dependent on time. Therefore, our system is a nonlinear system that is an autonomous system [26-30].

### 5.1 Stability of a system with inputs

Stability theory addresses the stability of solutions of differential equations and of trajectories of dynamical systems. If a small change happens in the hypothesis it leads to small variations in the conclusion of the theorem. We need to specify the metric used for determining the stability. In dynamical systems, an orbit is said to be in Lyapunov stable state if the forward orbit does not deviate more. Under favourable conditions the problem can be formulated using eigenvalues of matrices. A more general method involves Lyapunov functions. Stability of the system can be proven by various criteria. Asymptotic properties for differential equations and dynamical systems play an important role in determining what happens to the system after a long time. Stability in this article indicates the ability of the system to sustain quality message paths over a period of time ensuring lesser hop count on the path finalized for forwarding the data packets.

A system with parameters is written as

$$\dot{x} = f(x, p) \,, \tag{2}$$

where *p* is the parameter that controls the function *f*'s output. One needs to quantify the effect of inputs for considering the stability of the system. We have input-to-state stability mechanism for nonlinear systems for analysis purpose.

A nonlinear system does not satisfy the property of superposition and homogeneity and possess many equilibrium points. Lyapunov analysis is the best when we try to understand the system with respect to equilibrium points. Lyapunov function gives the information about the location of the limit set. It is nothing but the information about the asymptotic behaviour. Here in this article we have described three levels of asymptotic classification, namely low, medium and high. Lyapunov's method helps in determining the stability of the system without solving system equations. The functions in quadratic form give better understanding of Lyapunov equations.

The following are the properties of Lyapunov function, V:

- V is continuous.
- V has a unique minimum with respect to all other points in some neighbourhood of the equilibrium of interest.
- Along any trajectory of the system, the value of V never increases.

Here the stability of the working structure is classified as high, neutral or low (unstable) stability. These values help us in justifying the output as the output depends on the QoS metric values.

The input for the Lyapunov is an ordinary differential equation. Consider DE of the form

$$\frac{du}{dt} = f_L(u) \tag{3}$$

where $f$:R→R is a continuous differentiable function. A point $u^*$ is called a fixed point of the DE if $f(u^*) = 0$. The variation equation of $du/dt = f(u)$ is given by

$$\frac{du}{dt} = \left(\frac{df_L}{du(u(t))}\right)y \tag{4}$$

where it is assumed that $f$ is continuous differentiable.

Here we have assumed that the destination nodes act as the equilibrium points. An equilibrium state $x_e$, of the system is said to be asymptotically stable if it is stable in the sense of Lyapunov and if every solution starting within $S(\delta)$ converges, without leaving $S(\epsilon)$, to $x_e$ as $t$ increases indefinitely.

### 5.2 Lyapunov equation for the system

Let $X^*= 0$ be an equilibrium point of V($X$). Let V:D→R be a continuously differentiable function such that:

(i) V(0) = 0

(ii) V($X$) > 0, in D–{0} means positive definite

(iii) $\dot{V}(X) < 0, in\ D - \{0\}$ means negative definite

Then $X = 0$ is asymptotically stable.

Consider the quadratic form of Lyapunov equation which is given as follows:

QF : Rn→R

$$QF(x) = x^T P x \tag{5}$$

where P is a symmetric matrix where $P = P^T = [P_{ij}]$.

It may be shown that a quadratic function QF is positive definite if all the eigenvalues of P are positive.

The stability model for the proposed work, which indicates the global asymptotical stability, is as follows:

$$\dot{x}_1 = ax_1 \tag{6}$$

$$\dot{x}_2 = -bx_2 \tag{7}$$

Here $x_1$ represents the bandwidth, which is a positive effect as it increases and $x_2$ represents the delay which is a negative effect as bandwidth decreases.

$$\frac{\partial V}{\partial X} = g(X) = \begin{bmatrix} K_1 & K \\ K & K_2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{8}$$

$$\frac{\partial g_1}{\partial x_2} = \frac{\partial g_2}{\partial x_1} = K \tag{9}$$

Consider $K = 0$. Then,

$$g(X) = \frac{\partial V}{\partial X} = \begin{bmatrix} g_{1(X)} \\ g_{2(X)} \end{bmatrix} = \begin{bmatrix} K_1 x_1 \\ K_2 x_2 \end{bmatrix} \tag{10}$$

Here $K_1 = a$ and $K_2 = b$.

$$V(X) = \int_0^{x_1} g_1\,(\overline{x}_1,0)d\overline{x_1} + \int_0^{x_2} g_2\,(x_1,\overline{x_2})d\overline{x_2} \tag{11}$$

$$= \frac{1}{2}(K_1 x_1^2 + K_2 x_2^2) \tag{12}$$

If $K_1 > 0$ and $K_2 < 0$, then V($X$) is Lyapunov function. Therefore,

$$\dot{V}(X) = g^T(X) f(X)$$

$$= [K_1x_1 \;\; K_2x_2].\begin{bmatrix} ax_1 \\ -bx_2 \end{bmatrix} \tag{13}$$

$$= a\,K_1 x_1^2 - bK_2 x_2^2 \tag{14}$$

If $K_1 < 0$ and $K_2 > 0$, then $\dot{V}(X)$ is negative definite.

Here we superimpose the $f$(destination node) = 0 as it is the point we are trying to reach from source. The following are the steps involved in obtaining the optimal path and determine its stability:

Step 1: sub_path = source node

Step 2: $f(x) = V(x_1,x_2) = Ax_1^2 + Bx_2^2$, (15)

where $A$ and $B$ are the GV obtained for each node. For a node to be selected it should have satisfied the grade property parameters. Here we differentiate twice to get the GV. A is associated with Type-I check and B is associated with Type-II check.

Step 3: sub_path = sub_path + $f(x)$. Here + represents the concatenation of two nodes that have been selected after considering the GV.

Step 4: now choose the node $x$ obtained from $f(x)$ and go to Step 2, if $x$ is not the destination node, else go to Step 5.

Step 5: final path obtained, therefore we end.

We have considered $k_1 = 0.6$ and $k_2 = 0.2$. We have done Lyapunov calculation for only bandwidth and delay parameters, and therefore the GV has been reassigned as shown in Table 2.

**Table 2. Reassigned GV for Lyapunov calculation**

| B | De | SL |
|---|---|---|
| H | H | 2 |
| H | L | 3 |
| L | H | 0 |
| L | L | 1 |

We substitute in Equations 6 and 7, for $x_1$ and $x_2$ depending on the criteria met by the nodes from Table 2. Consider the simulation run for 15 nodes as in Table 3. We have Lyapunov value defined for the same as in Table 4.

**Table 3. Sample from simulation**

| Node number | RA | Neighbour node | B | De | PL |
|---|---|---|---|---|---|
| 0 | 35 | 1 | 38 | 4 | 0 |
| | | 6 | 21 | 2 | 1 |
| | | 8 | 21 | 6 | 0 |
| | | 10 | 38 | 6 | 0 |
| 1 | 38 | 0 | 38 | 4 | 0 |
| | | 4 | 3 | 5 | 0 |
| 2 | 22 | 9 | 12 | 6 | 1 |
| | | 10 | 11 | 4 | 1 |
| 3 | 22 | 10 | 11 | 4 | 1 |
| 4 | 36 | 1 | 3 | 5 | 0 |
| | | 5 | 21 | 2 | 1 |
| | | 7 | 18 | 2 | 0 |
| | | 14 | 7 | 6 | 1 |
| 5 | 35 | 4 | 21 | 2 | 1 |
| 6 | 35 | 0 | 21 | 2 | 1 |
| 7 | 39 | 4 | 18 | 2 | 0 |
| | | 11 | 16 | 4 | 0 |
| 8 | 39 | 0 | 21 | 6 | 0 |
| | | 10 | 16 | 4 | 0 |
| | | 13 | 33 | 3 | 0 |
| 9 | 24 | 2 | 12 | 6 | 1 |
| | | 12 | 33 | 3 | 0 |
| 10 | 35 | 0 | 38 | 6 | 0 |
| | | 2 | 11 | 4 | 1 |
| | | 3 | 11 | 4 | 1 |
| | | 8 | 16 | 4 | 0 |
| 11 | 35 | 7 | 16 | 4 | 0 |
| 12 | 36 | 9 | 33 | 3 | 0 |
| 13 | 36 | 8 | 33 | 3 | 0 |
| 14 | 19 | 4 | 7 | 6 | 1 |

**Table 4. Lyapunov calculation for the sample of 15 nodes**

| Path | GV | V*x* | $\dot{V}(X)$ |
|---|---|---|---|
| 0-8-10 | 4-4 | 0.5(0.6*2*2-0.2*2*2)+ 0.5(0.6*2*2-0.2*2*2)=1.6 | (-0.6*0.6*2*2-0.2*0.2*2*2)+ (-0.6*0.6*2*2-0.2*0.2*2*2)=–3.2 |
| 0-10 | 4 | 0.5(0.6*2*2-0.2*2*2)=0.8 | (-0.6*0.6*2*2-0.2*0.2*2*2)=–1.6 |

In V*x* calculation, we consider *a* as positive and *b* as negative and hence we have obtained the value. In $\dot{V}(X)$ calculation, *a* and $k_1$ are same and *b* and $k_2$ are of same value. Here $k_1$ is $< 0$ and hence has a negative value associated with it.

## 6 SIMULATION RESULTS

The simulation studies were carried out extensively using a Java-built simulation bed. In the simulation system developed for this total model of cognitive network, in any run, the user enters the number of nodes to be built for a random run. Depending on the number entered by the user, the nodes are spread across the region of view and links get generated and learning is performed as shown in Figure 3. The required QoS parameter is also assigned randomly.

We enter the source and destination node for determining the path. The learn button helps in remembering the path already executed on the run. This button also carries out grading of the node, which signifies the quality of the node. This information is useful for future, i.e. if such a pair of source and destination node is again looked for later, then the path is not computed by the

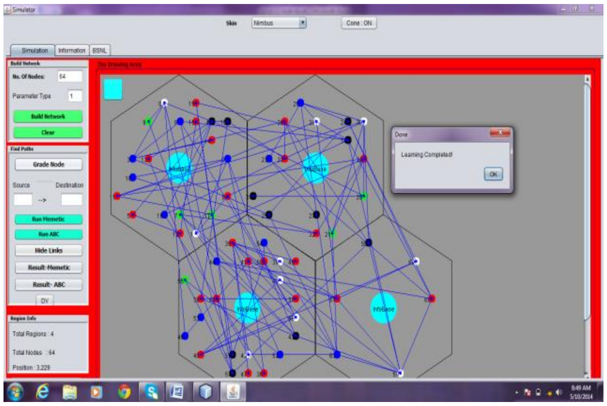

**Figure 3. Distribution of nodes and links along with learning**

algorithm, instead the path is determined from the Information Base. We choose the desired algorithm for viewing the output that uses the help of logical reasoning for the graded node.

The state updates were calculated depending on the programmable frequency factor like once in 30 min or once in 60 min. The whole network is analysed in abstract mode, hence there is no need for a real-time coupling of the system. Here the QoS parameters are learnt and updated based on the consumption of the resources during the simulation run. The initial state update is shown in Figure 1, and the final state update is shown in Figure 4.

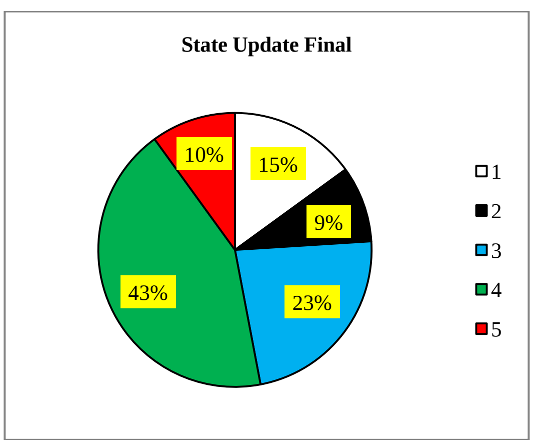


**Figure 4. Final state update**

Here the data is assigned based on random function. Table 5 gives a detailed description of the experimental runs carried for node numbers varying from 32 to 1024, which are all to the power of 2.

**Table 5. Fitness value**

| | **Memetic** | | **ABC** | |
|---|---|---|---|---|
| No. of nodes | Fitness value | Bandwidth | Fitness value | Bandwidth |
| 32 | 0.9 | 15 | 0.9 | 15 |
| 64 | 0.74 | 102 | 0.74 | 102 |
| 128 | 0.70 | 234 | 0.70 | 234 |
| 256 | 0.76 | 125 | 0.81 | 110 |
| 512 | 0.55 | 376 | 0.65 | 220 |
| 1024 | 0.54 | 717 | 0.72 | 534 |

Table 5 shows how the two algorithms perform on the graded network. The total throughput chosen by Memetic is equal or more than all six cases as compared to ABC.

Using the cone approach we reduce the search space. Each region has one agent that collects information about the nodes belonging to the region and also assigns the GV to each node. We observed that the simulation set-up time using cone is higher than without using the cone. This is because learning aspect takes place within the cone. The learnt knowledge is made available in the Information Base which is nothing but the agent. The stability of the output path obtained is justified by the GV as well as the Lyapunov value as shown in Table 6.

**Table 6. Lyapunov value and grade equation value**

| $\boldsymbol{V(X)}$ | $\boldsymbol{\dot{V}(X)}$ | **Average grade** | **GV** |
|---|---|---|---|
| 1.6 | –3.2 | 4 | 0.8 |
| 5.2 | –10.4 | 4 | 0.85 |
| 8.6 | –17.2 | 3 | 0.828 |
| 6.2 | –12.4 | 4 | 0.9 |
| 19.4 | –38.8 | 4 | 0.938 |
| 26.2 | –52.4 | 4 | 0.89 |

It was observed that longer paths had higher $V(X)$ than the shorter paths. The GV helps in classifying the stability of the path. The graphs in Figures 5 and 6 show the hop count obtained when MA and ABC was carried out with cone and without cone. With respect to hop count, the cone actually helped in obtaining shorter path with required quality of the complete path.

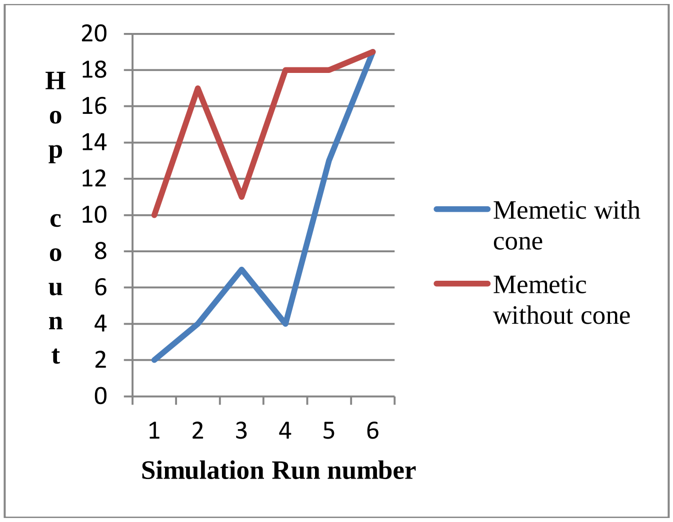


**Figure 5. MA – hop count comparison**

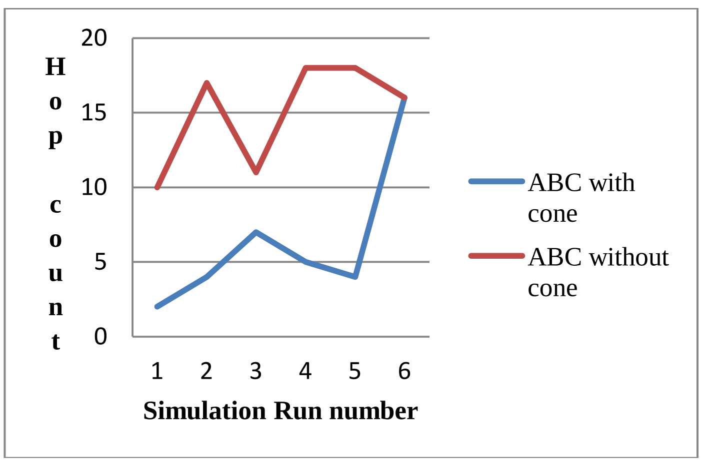


**Figure 6. ABC algorithm – hop count comparison**

A complete result of different run is provided in the Appendix as shown in Table A.1 and A.2. The initial assignment range of values assigned to the nodes which have been taken from the BSNL provider is described in Table A.3. For all runs the packet size is considered as 1Mb. The run for 32 nodes is captured and is as shown in Figure A.1. The values assigned for the QoS parameter in the simulation is provided in Table A.4. the Result obtained for the 32 node run is as given in Table A.5 and A.6.

## 7 CONCLUSION

In this article, we have presented the successful implementation of optimal path determination by accounting the quality levels of the nodes and the way in which they are getting graded over the network, bringing in self-awareness concepts to network. This technique of obtaining an estimation of the quality of the nodes and its nature of distribution is realized through parametric grading. It presented the various modes of grading with state update process for maintaining apt cognition. Further on this set-up having cognitive ability, we reduce the search space by creating a geographical focus cone directed to the destination. The optimal path was obtained based on the nature-inspired algorithms. Further evaluation whether to calculate the path or not, is determined by the Information Base that remembers the path obtained earlier.

The results show that ABC algorithm outperformed the Memetic technique in terms of throughput and fitness for optimal path determination. Also it has shown the advantage of reduction in geographical search space to realize lesser hop count. The suggested algorithms are shown to be successful by the simulation. The results are justified by Lyapunov values which gives the proof that the obtained paths are of required quality.

## 8 REFERENCES

1. El-Abd, M.: Performance assessment of foraging algorithms vs. evolutionary algorithms. Information Sciences 182, 243–263 (2012).
2. III, M., Joseph.: Cognitive Radio – An Integrated Agent Architecture for Software Defined Radio (Ph.D. Dissertation), Kista, Sweden: Royal Institute of Technology (2000).
3. Hou, R., Lui, K., Leung, K., and Baker, F.: Performance analysis of quantization-based approximation algorithms for precomputing the supported QoS, Journal of Network and Computer Applications, Vol. 40, April , pp. 244–254 (2014).
4. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm, Applied Soft Computing, vol. 8, pp. 687–697 (2008).

5. Tereshko, V.: Reaction–diffusion model of a honeybee colony's foraging behaviour, in: M. Schoenauer (ed.), Parallel Problem Solving from Nature VI, Lecture Notes in Computer Science, vol. 1917, Springer–Verlag, Berlin, pp. 807–816 (2000).
6. Thomas, R., W., DaSilva, L., A., Allen, B., Mackenzie.: Cognitive Networks. :Proceedings of the First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 352–360 (2005).
7. Fortuna, C., Mohorcic, M.: Trends in the development of communication networks: Cognitive networks. Journal of Computer Networks 53, 1354–1376 (2009).
8. Nair, T., R., G., Sooda, K.: An intelligent routing approach using genetic algorithms for quality graded network, International Journal of Intelligent Systems Technologies and Applications, vol. 11, 196–211., (2012).
9. Kephart, J., O., Chess, D., M.: The Vision of Autonomic Computing, IEEE Computer Society, 36(1), 41–50 (2003).
10. Raymer, D., Meer, S., Strassner, J.:From Autonomic Computing to Autonomic Networking: an Architectural Perspective, 5th IEEE Workshop on Engineering of Autonomic and Autonomous Systems, 174–183 (2008).
11. Konstantinou, A. V.: Towards Autonomic Networks, Ph.D. Thesis, Columbia University (2003).
12. Sooda, K., Nair, T., R., G.:. Optimal path selection in graded network using Artificial Bee Colony algorithm with agent enabled Information. 12th IEEE International Conference on Hybrid Intelligent Systems, Pune, 4–7 December (2012).
13. Reitblatt, M., *et al*.: Consistent Updates for Software-Defined Networks: Change You Can Believe In!, Hotnets '11, ACM, 2011.
14. Nair, T., R., G., Sooda, K.: Application of genetic algorithm on quality graded networks for intelligent routing. World Congress on Information and Communication Technologies, 11– 14 December 2011, Mumbai, India, pp. 558–563 (2011).
15. Lee, C., Wada, H., Suzuki, J.:Exploring biologically-inspired evolvable network applications with the BEYOND architecture: IEEE conference on Bio-Inspired Models of Network, Information and Computing Systems, 1-8 (2006).
16. Magnani, L.: Abduction, Reason, and Science. Processes of Discovery and Explanation. Kluwer Academic/Plenum Publishers, New York (2001).
17. Mohamed, D., T., Kotb, A., M., Ahmed, S., H.: A Medium Access Protocol for Cognitive Radio Networks Based on Packet's Collision and Channels' Usage, International Journal of Digital Information and Wireless Communications (IJDIWC) 4(3): 314-332 (2014).
18. Ong, Y., S.: Artificial intelligence technologies in complex engineering design, Ph.D. Thesis, School of Engineering Science, University of Southampton, UK (2002).
19. Choudhary, D., and Robinson, A.L.: A New Approach to Wireless Channel Modeling using Finite Mixture Models: International Journal of Digital Information and Wireless Communications (IJDIWC) 4(2): 169-183 (2014).
20. Nakano, T.: Biologically inspired network systems: A review and future prospects, IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, vol. 41, 630–643 (2011).
21. Hu, T., Lin, Z.: Properties of the composite quadratic Lyapunov functions, IEEE Transactions on Automatic Control, 49, 1162–1167 (2004).
22. Chen, X, Ong, Y., Lim, M., Tan, K.: A multi-facet survey on memetic computation. IEEE Transactions on Evolutionary Computation, 15(5), 591–607 (2011).
23. Sanusi, H., A., Zubair, A., Oladele, R., O.: Comparative assessment of genetic and memetic algorithms, Journal of Emerging Trends in Computing and Information Sciences, vol. 2, pp.498-508 (2011).
24. Krasnogor, N.: Studies on the Theory and Design Space of Memetic Algorithms, Ph.D. Thesis, Faculty of Computing, Mathematics and Engineering, University of the West of England, Bristol, UK (2002).
25. Karaboga, D., Ozturk, C.: A novel clustering approach: Artificial Bee Colony (ABC) algorithm. Applied Soft Computing vol 11, 652–657 (2011).
26. Naghshvar, M., Zhuang, H., Javidi, T.: A general class of throughput optimal routing policies in multi-hop wireless networks, IEEE Transactions on Information Theory, vol. 58, 2175–2193 (2012).
27. Marquez, H., J.: *Nonlinear Control Systems*, Wiley-Interscience, USA (2003).
28. Luenberger, D., G., Ye, Y.: *Linear and Nonlinear Programming*, 3rd edn, Springer, (2008).
29. Parks, P., C.: A. M. Lyapunov's stability theory—100 years on, IMA Journal of Mathematical Control & Information, vol. 9, 275–273 (1992).
30. Matousek, *R.,* IAENG, I.: Svarc, P., Pivoňka, P., Osmera, M., Seda.: Simple methods for stability analysis of nonlinear control systems, Proceedings of the World Congress on Engineering and Computer Science Vol II, WCECS, 20–22 October 2009, San Francisco, USA (2009).

# APPENDIX

**Table A1. QoS distribution**

| No. of nodes | Total links | QoS | | | | Inside cone | | |
|---|---|---|---|---|---|---|---|---|
| | | B | De | RA | PL | Node | Links | Total paths |
| 32 | 109 | 105 | 90 | 26 | 103 | 4 | 11 | 2 |
| 64 | 232 | 220 | 130 | 47 | 230 | 29 | 91 | 2 |
| 128 | 400 | 386 | 224 | 103 | 396 | 35 | 81 | 1 |
| 256 | 857 | 823 | 533 | 210 | 853 | 63 | 212 | 32 |
| 512 | 1648 | 1596 | 1096 | 425 | 1644 | 49 | 178 | 3 |
| 1024 | 3391 | 3269 | 2061 | 803 | 3387 | 364 | 1077 | 40 |

**Table A2. Path and Grade Value (GV)**

| Path | GV |
|---|---|
| 2-1-30 | 4-4 |
| 0-22-28-59-56 | 4-4-3-5 |
| 14-7-22-92-93-98-103-117 | 5-4-3-4-4-4-3 |
| 42-148-180-174-250 | 5-4-3-5 |
| 59-52-57-88-40-158-154-156-166-274-275-306-303-308 | 5-4-5-5-5-5-4-4-5-5-5-5-4 |
| 64-56-801-776-624-607-590-593-558-554-559-599-626-656-645-639-636-802-812-773 | 4-5-5-4-4-5-5-3-3-5-5-4-4-4-4-4-5-5-5 |

**Table A3. Simulation set-up**

| Sl. No | Metric | Range | Threshold |
|---|---|---|---|
| 1. | Bandwidth | 2– 64 Mbps | 4 |
| 2. | Delay | 2–7 ms | 4 |
| 3. | Packet loss | 0–4 | 2 |
| 4. | Buffer space | 15–40, each of 1 Mb length | 20 |

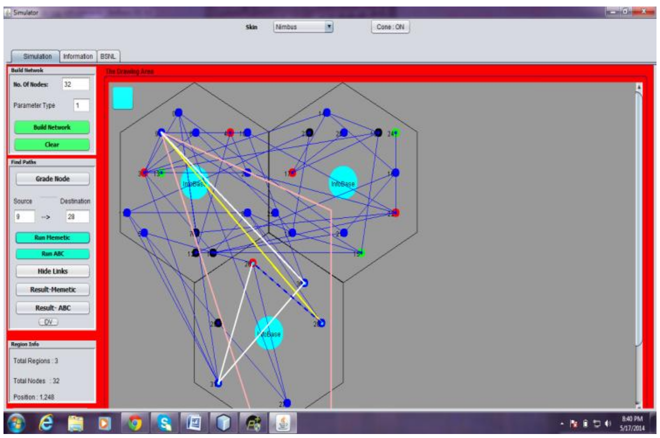

**Figure A1. Run of 32 nodes**

**Table A4. QoS value assigned for 32 nodes**

| Node number | RA | Neighbour node | B | De | PL |
|---|---|---|---|---|---|
| 0 | 21 | 3 | 57 | 6 | 0 |
| | | 7 | 45 | 5 | 1 |
| | | 21 | 46 | 5 | 1 |
| | | 27 | 18 | 6 | 1 |
| | | 31 | 6 | 4 | 1 |
| 1 | 19 | 3 | 45 | 6 | 1 |
| | | 5 | 61 | 6 | 1 |
| | | 8 | 62 | 6 | 0 |
| | | 21 | 46 | 5 | 1 |
| | | 22 | 12 | 5 | 1 |
| | | 26 | 14 | 5 | 1 |
| 2 | 36 | 3 | 45 | 6 | 1 |
| | | 4 | 29 | 2 | 0 |
| | | 13 | 4 | 4 | 0 |
| | | 22 | 10 | 4 | 0 |
| 3 | 34 | 0 | 57 | 6 | 0 |
| | | 1 | 45 | 6 | 1 |
| | | 2 | 45 | 6 | 1 |
| | | 3 | 62 | 5 | 1 |
| | | 6 | 33 | 4 | 1 |
| | | 9 | 36 | 4 | 1 |
| | | 14 | 51 | 6 | 1 |
| | | 18 | 48 | 4 | 0 |
| | | 31 | 53 | 2 | 0 |
| 4 | 33 | 2 | 29 | 2 | 0 |
| | | 12 | 28 | 4 | 0 |
| 5 | 28 | 1 | 61 | 6 | 0 |
| | | 31 | 53 | 2 | 0 |
| 6 | 36 | 3 | 33 | 4 | 1 |
| | | 25 | 40 | 4 | 1 |
| 7 | 19 | 0 | 45 | 5 | 1 |
| | | 22 | 10 | 4 | 0 |
| 8 | 38 | 1 | 62 | 6 | 0 |
| | | 11 | 12 | 5 | 1 |
| | | 19 | 58 | 2 | 0 |
| | | 28 | 14 | 6 | 1 |
| | | 30 | 2 | 2 | 1 |