MULTIPLE ANT-BEE COLONY OPTIMIZATION FOR LOAD BALANCING IN PACKET-SWITCHED NETWORKS

Mehdi Kashefikia[1],Nasser Nematbakhsh[2], Reza Askari Moghadam[3]

[1]Computer Department, PayameNoor University (PNU), Tehran, Iran
Kashefi.m@gmail.com
[2]Faculty of Computer Engineering, Islamic Azad University, Najafabad Branch, Iran
Nemat@eng.ui.ac.ir
[3]Computer Department, Payame Noor University(PNU), Tehran, Iran
Askari@pnu.ac.ir

ABSTRACT

One of the important issues in computer networks is "Load Balancing" which leads to efficient use of the network resources. To achieve a balanced network it is necessary to find different routes between the source and destination. In the current paper we propose a new approach to find different routes using swarm intelligence techniques and multi colony algorithms. In the proposed algorithm that is an improved version of MACO algorithm, we use different colonies of ants and bees and appoint these colony members as intelligent agents to monitor the network and update the routing information. The survey includes comparison and critiques of MACO. The simulation results show a tangible improvement in the aforementioned approach.

KEYWORDS

Load balancing, swarm intelligence, ant colony, bee colony, routing

1. INTRODUCTION

"Load balancing" is a methodology to distribute workload across multiple computers or a computer cluster, network links, central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload.

A typical network consists of series of routers, each router independently communicating with others. Any router uses number of packets for navigation purposes and several limited queues to buffer packets for each node. In each network, links have limited bandwidth and limited capacity. To reduce the packet transfer time, packets should be distributed in a way that minimizes the queue length in each router.

Ant colony algorithms which are inspired by these tiny creatures strategy to find new resources of food are one of the recent approaches toward solving some important computer network problems. One of these problems is load balancing. Ant colony algorithms have been able to propose solutions for routing and load balancing in packet-switched networks. [1, 10, 21, 22]

Despite the fact that multiple colony algorithms have a natural potential to be used in load balancing, relatively less attention has been paid to them[2,3]. In these algorithms several colonies of ants work together in such a way that ants in same colony are attracted to each other and simply repulse ants from other colonies. This specification can be used in network routing to find several balanced routes.

DOI : 10.5121/ijcnc.2011.3508

Another swarm intelligence algorithm, Bee Colony Algorithm, which is an optimization algorithm based on the intelligent foraging behaviour of honey bee swarm, is proposed by Karaboga in 2005[4, 5, 6, 24, 25].

In this paper, using swarm intelligence solutions, based on several colonies of ants and combine them with bee colony, a new algorithm for load balancing in computer networks has been introduced. The term “Load Balancing” here means sending packets from several relatively optimal paths in order to avoid congestion in the most optimal path.

2. ROUTING BASED ON ANT COLONY OPTIMIZATION

In computer science and operations research, the ant colony optimization algorithm (ACO) is aprobabilistic technique toward solving computational problems which can be reduced to finding good paths through graphs.

The proposed algorithm in this paper is based on Ant-Net that is invented by Caro and Dorigo [7, 8, 9, 19, 26]. Ant-Net algorithm was used at first for routing in packet-switched networks. Unlike the former methods in routing, such as distance-vector and link-state that focus on the minimum cost route, routing based on Ant-Net is aimed toward optimizing performance in the entire network.

Based on important parameters for efficiency, like throughput and average transmission delay, Caro and Dorigo conducted experiments on typical networks such as NTTnet, Simulated NSFNET and SimpleNet. [7, 8, 9, 19].

3. ANT-BEE COLONY ALGORITHM

Ant-Bee algorithm is in fact an optimization on Ant-Net algorithm and tries to improve its performance [12]. This algorithm, at the beginning uses forward ants to find a suitable solution from one node to another and then these are the bees who update the routing tables based on ants collected data.

3.1. Routing Tables

The main purpose of Ant-Bee algorithm is to improve the convergence time in Ant-Net algorithm using more accurate pheromone laying strategy and for doing so it needs more information to be stored in the routing tables. A typical routing table for Ant-Bee is shown in Table 1. This is an LxN table in which L represents all the outgoing links and N is the number of nodes minus one (the node itself). P_{ij} is the chance of node i to be selected as the next node on the way to j and M_{ij} is a vector like [T_{ij}^1, T_{ij}^2,…., T_{ij}^3] in which T_{ij}^K represents the K^{th} dancer bee’s recorded trip time.

Table 1. A typical Ant-Bee routing table

From \ To	**1**	**2**	**. . .**	**N**
1	P_{11},M_{11}	P_{12},M_{12}	. . .	P_{1N},M_{1N}
⋮	⋮	⋮	⋮	⋮
L	P_{L1},M_{L1}	P_{L2},M_{L2}	. . .	P_{LN},M_{LN}

3.2. Ant-Bee algorithm

In this section we are briefly explaining Ant-Bee algorithm: [12, 20].

- A forward ant starts traveling through the network. Whenever this forward ant reaches a node, if this node is not the destination node, it is directed toward the destination. These forward ants have the same priority as data packets.

- When a forward ant reaches the destination, based on the provided information by her, a backward bee is created then the forward ant is killed and the new born bee continues the journey.

- This backward bee traverses the forward ant's travelled route in reversed direction and on its way updates the pheromone tables and is finally killed at the starting node (the node which had initiated the forward ant). It has to be mentioned that backward bees have more priority than data packets to be able to apply the emergency changes as rapidly as possible.

The distinguishing element between Ant-Bee and Ant-Net algorithms is Ant-Bee's use of backward bees. The three different kinds of bees which are in use in this algorithm are:

1. Dancer bee: whenever this bee reaches a node, after updating the node's pheromone table, sets down its traverse time in related field. The pheromone updating strategy is the same as backward ants in Ant-Net algorithm.

2. Follower bee: These are considered as naive bees that should collect information based on dancer bee's dancing parameters. They use equation (1) for doing so. Assume that the follower bee has come from node j to i on its way to destination d.

$$T_{jd} = \beta T_{jd}^{flw} + (1-\beta) T_{jd}^{dnc} \quad ,0 \le \beta \le 1 \qquad (1$$

In this equation:

- T_{jd}^{flw}: Follower bee's trip time.

- T_{jd}^{dnc} : Selected trip time among dancer bees based on Mij in a way that will be explained shortly.

- β : An impact factor which determines the two first factor's effect. If $\beta = 1$, it means the follower bee doesn't pay any attention to dancer bees and this algorithm will work the same as Ant-Net algorithm. In contrast, if $\beta = 0$ the follower bee is entirely following the dancer bee.

There is a question! When a follower bee refers to M_{ij}, which T_{ij}^{K} should she pick? Well this selection is done randomly but each trip time has a different probability based on which we choose. Having a follower bee starting from node d, travelled from j to i, P_k is determined from equation (2).

$$P_k = \frac{\frac{1}{T_{jd}^{k}}}{\sum_{k'=1}^{M} \frac{1}{T_{jd}^{k'}}} \qquad (2)$$

In this equation:

- P_k:The probability of K^{th}trip time to be selected.
- T_{jd}^{k} : K^{th}trip time in M_{ij} vector

The shorter the trip time is, the higher its probability to be chosen. Shorter bee trip time means the associated forward ant has travelled a shorter way toward destination.

3. Introvert bee: These bees act exactly the same way as backward ants in Ant-Net algorithm and completely ignore other bees' activities.

4. MULTIPLE ANT COLONY ALGORITHM IN LOAD BALANCING

This approach uses several colonies of ants to find the optimal route [2]. Each colony of ants lays its own pheromone which is recognized from others by a different colour. The main reason for using several colonies is for each node to be able to hold several routing tables (pheromone tables). This strategy lets us to choose more than one route in each node. This ability is very important in distributing network traffic between different routes and achieving a balanced network.

Although the ants attract to pheromone laid from ants in same colony, the so-called "Repulsion" strategy prevents ants from different colonies to take the same optimal route. In other word, in making decision for choosing a route not only they take into account the laid pheromone from ants of the same colony, but also they consider other colonies pheromone.

This strategy is introduced by Varela and Sinclair[11]. It is used in Multi-wavelength networks for virtual-wavelength-path routing. In this practice artificial ants are attracted by the pheromone trail of ants from their own colony and also repelled by the pheromone of other colonies.

5. MULTIPLE ANT-BEE COLONY ALGORITHM

In multiple Ant-Bee colony, using several ant-bee colonies with different types of pheromone, an effort has been made to distribute the network traffic in several optimal local paths by creating an using several pheromone tables in each node and routing the data packets using these tables. We will shortly explain how this strategy results in reducing congestion and collision in a typical route and consequently reducing packet transmission time in the network. Therefore, because of holding various pheromone tables in each node, memory cost increases tangibly.

5.1. Routing table and pheromone laying strategy

General structure of a routing table for Multiple Ant-Bee colony is shown in Table 2. This is an LxN table in which L represents all the outgoing links and N is the number of nodes minus one (the node itself). Each cell contains values M^{k}_{ij} and P^{K}_{ij}. $P^{K}_{ij}$is the chance of node i to be selected as the next node for colony K and M^{k}_{ij} represents some of the recorded trip times of dancer bees form K^{th}colony which have travelled from i to j. M^{k}_{ij} is a vector like the one which is presented in Ant-Bee colony algorithm in section 4.

Table 2. Multiple Ant-Bee Colony routing table for colony K

From \ To	1	2	. . .	N
1	P_{11}^{K}, M_{11}^{K}	P_{12}^{K}, M_{12}^{K}	. . .	P_{1N}^{K}, M_{1N}^{K}
⋮	⋮	⋮	⋮	⋮
L	P_{L1}^{K}, M_{L1}^{K}	P_{L2}^{K}, M_{L2}^{K}	. . .	P_{LN}^{K}, M_{LN}^{K}

In this algorithm we use one pheromone table like Table 2. for each colony of Ant-Bees. When a bee from colony K meets a node, it only updates the associated pheromone table. Although the forward ant is interested in pheromone table of its own colony, because of the "Repulsion" strategy it considers other colonies' pheromone tables as well.

5.2. Selection of next node

As mentioned before, ants are attracted to the trail of their own colony and are repelled by the pheromone of other colonies. We explain these two reactions briefly:

5.2.1. Attraction strategy

We use α_{jd}^{K} parameter to indicate attraction of a forward ant from colony K to node j on its way to destination d. We calculate this parameter from equation (3)

$$\alpha_{jd}^{K} = \frac{P_{jd}^{K}}{\sum_{i \in N_i} P_{id}^{K}} \qquad (3$$

In this equation:

- P_{jd}^{K}: quantity of pheromone K in the edge linked to node j on its way to destination d(jd element in pheromone table of K^{th} colony)
- N_i :A set containing all possible outgoing edges for the forward ant.

5.2.2. Repulsion strategy

In the proposed algorithm we use parameter β_{jd}^{K} to show repulsion of forward ants of colony K to j on its way to destination d. β_{jd}^{K} is calculated from equation (4).

$$\beta_{jd}^{K} = \frac{P'^{K}_{jd}}{\sum_{i \in N_i} P'^{K}_{id}} \qquad (4)$$

In this equation:

- P'^{K}_{jd} : Amount of pheromone from all colonies (except K itself) in edge linked to node j on its way toward destination d (sum of all jd elements in pheromone tables except colony K). This parameter is calculated from $P'^{K}_{jd} = \sum_{H \neq K} P_{jd}^{H}$

- N_i: Sum of all possible edges for the forward ant.

After calculating attraction and repulsion parameters, it's time to determine the chance of each node to be selected as next node. For this purpose we introducethat is γ_{jd}^K computed from equation (5).

$$\gamma_{jd}^K = \frac{\alpha_{jd}^K / \beta_{jd}^K}{\sum_{i \in N_i} \alpha_{id}^K / \beta_{id}^K} \quad \text{Such that} \quad \sum_{i \in N_i} \gamma_{id}^K = 1 \quad (5)$$

Node j has the chance of γ_{jd}^K to be selected as next node by a forward ant of colony K on its way to destination d.

6. SIMULATIONS AND RESULTS

In this section we show simulation results on NSFNET and NTTNet for both MACO and MABC. We use different number of colonies and percentages of bees are 40% introvert bees, 30% dancer bees and 30% follower bees. In Table 3.and Table 4. we show results for MACO on NSFNET and NTTNet networks. Table 5.and Table 6. show the simulation results with the same parameters on same networks using MABC algorithm. As it is clear from these tables, results are approximately equal.

Table 3.MACO simulation on NSFNET. Queue length=10

Number Of colonies	Received Packets percentage	Throughput	Average Queue Delay	Average Packet Delay	Average Queue Length	Queue Length Variance	Average Leap Number
1	99.718	10.514	0	0.005	0.258	0.589	2.869
2	99.754	21.015	0	0.005	0.178	0.313	2.828
3	99.784	31.581	0	0.005	0.163	0.234	2.808
4	99.805	42.368	0	0.005	0.173	0.247	2.805
5	99.818	52.765	0	0.005	0.177	0.25	2.792
6	99.825	63.376	0	0.005	0.200	0.311	2.794
7	99.833	73.692	0	0.005	0.214	0.338	2.792
8	99.838	83.924	0	0.005	0.232	0.391	2.785

Table 4.MACO simulation on NTTNet. Queue length=1000

Number Of colonies	Received Packets percentage	Throughput	Average Queue Delay	Average Packet Delay	Average Queue Length	Queue Length Variance	Average Leap Number
1	99.230	043.238	0.022	0.223	31.632	13715.573	9.030
2	99.375	085.876	0.018	0.192	20.897	8986.933	9.099
3	99.467	128.462	0.015	0.165	12.953	4936.962	9.065
4	99.508	170.948	0.013	0.148	10.573	3815.496	9.049
5	99.536	213.583	0.013	0.140	11.992	4607.675	9.074
6	99.710	042.632	0.008	0.098	10.181	3609.257	9.109
7	99.752	042.425	0.006	0.085	08.493	2890.44	8.962
8	93.191	079.078	0.034	0.160	73.626	57265.047	9.266

In Table 4.simulation results of MABC on NTTNet network with queue length 1000 is shown.

The main difference between MACO and MABC algorithms is when a failure happens in the network. We simulate results a failure in both networks in next section.

Table 5.MABC simulation on NSFNET. Queue length=10

Number Of colonies	Received Packets percentage	Throughput	Average Queue Delay	Average Packet Delay	Average Queue Length	Queue Length Variance	Average Leap Number
1	99.631	09.931	0	0.005	0.261	0.633	2.924
2	99.700	10.565	0	0.005	0.188	0.346	2.909
3	99.694	20.256	0	0.005	0.166	0.252	2.888
4	99.721	30.751	0	0.005	0.170	0.238	2.883
5	99.743	41.315	0	0.005	0.179	0.251	2.875
6	99.751	51.759	0	0.005	0.198	0.3	2.870
7	99.803	10.260	0	0.005	0.214	0.344	2.906
8	99.814	20.735	0	0.005	0.234	0.396	2.862

Table 6. Simulation results on NTTNet. Queue length in each node=1000

Number Of colonies	Received Packets percentage	Throughput	Average Queue Delay	Average Packet Delay	Average Queue Length	Queue Length Variance	Average Leap Number
1	99.190	42.565	0.023	0.243	33.542	14309.464	9.371
2	99.339	85.543	0.019	0.203	20.626	8786.751	9.353
3	99.431	127.992	0.016	0.175	13.854	5201.978	9.359
4	99.639	41.911	0.008	0.103	11.502	4452.687	9.235
5	99.648	84.277	0.008	0.097	9.523	3206.432	9.175
6	99.652	127.056	0.008	0.100	10.721	3929.405	9.214
7	99.643	169.884	0.008	0.101	10.749	3928.296	9.254
8	99.176	211.622	0.010	0.107	23.194	14996.891	9.291

6.1. MACO and MABC behaviour on failure

In this section we show the simulation results using both MACO and MABC algorithms on occurrence a common failure in the network. This experiment is conducted on NTTNet network. In this experiment we interrupted node 21 from second 150 to 650 and node 40 from second 300 to 650. The entire simulation time is 1000 seconds. In Figure 1 we show NTTNet schema.

Figure 1. Failure in nodes 21 and 40 in NTTNet

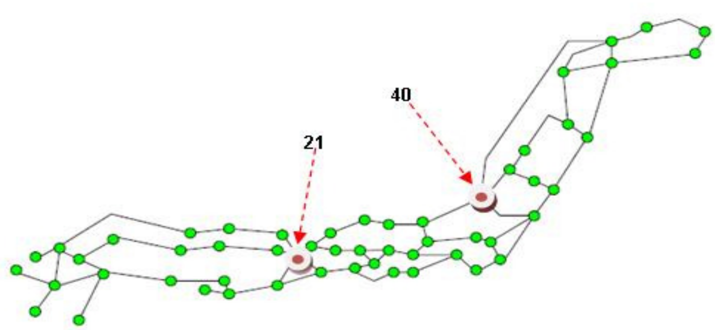


Figure 2. Transmissiondelay on NTTNet, using MABC and number of colonies from one to seven

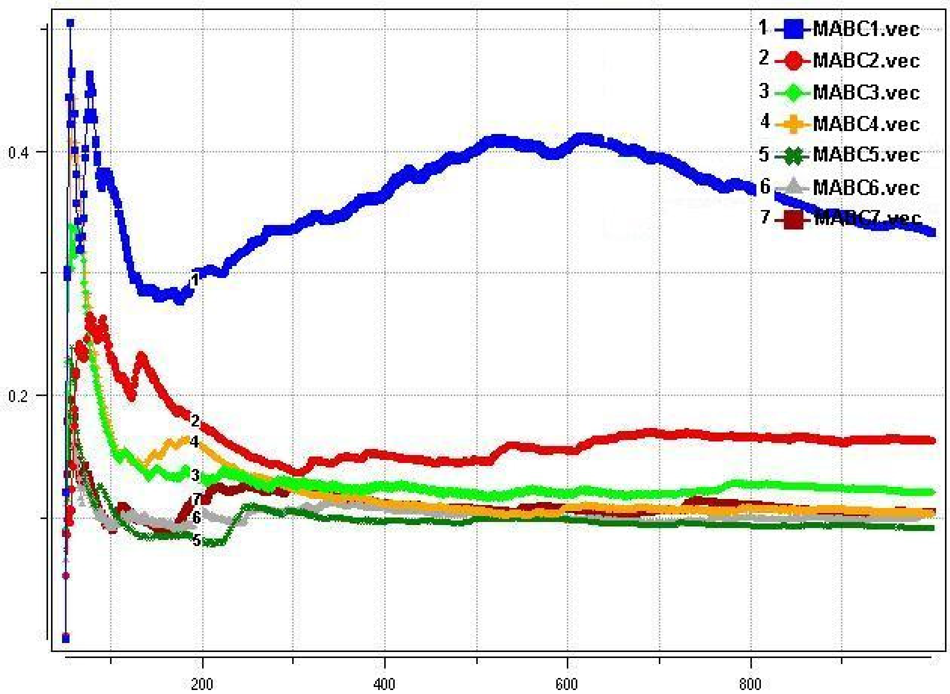
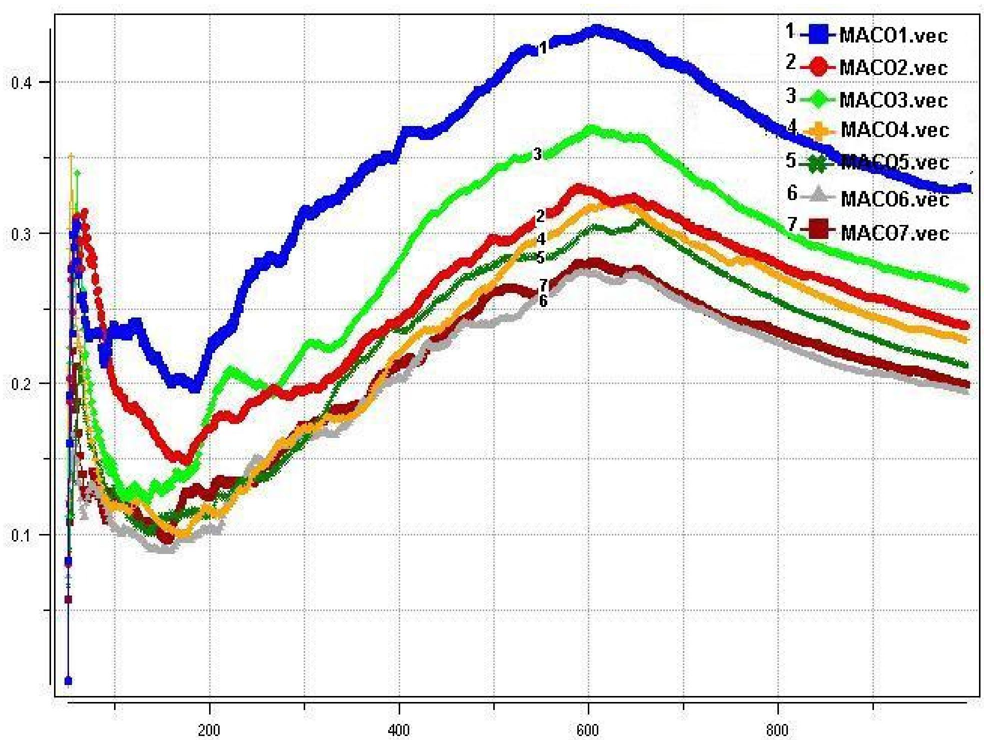


Figure 3. Transmission delay on NTTNet, using MACO and number of colonies from one to seven



As it's shown in Figure 2, on failing node 21 from second 150, using just one colony (equivalent to Ant-Bee algorithm), packet transmission delay starts to rise and continues rising until second 600. In this stage, transmission delay which has risen from 0.3 on second 150 to 0.4 on second 600 starts to decrease but is never stabilized until the end of the simulation (second 1000). Repeating the algorithm using two colonies, results in stabling the network very soon. Using three or six colonies have even better results and don't cause any delay at all. Using four colonies causes a slight delay at the beginning of the interrupt but stabilizes sooner than the three-colony algorithm. Using five or seven colonies has the same behaviour as four-colony algorithm but shows smaller transmission delay.

Figure 3.shows the same experiment using MACO algorithm. As it's clear from this Figure, MACO shows the same behaviour for different number of colonies. When a failure occurs, the network loses its stability and transmission delay rises dramatically.

Comparing Figure 2. and Figure 3., we can obviously conclude that for number of colonies greater than one, MABC has significantly better effect when a failure happens in the network and is more suitable for unstable networks.

7. CONCLUSION

In this paper we introduced a new approach called MABC. The main goal of this algorithm is load balancing and stabilizing the network. This approach uses several colonies of ants or ant-bees to find various locally optimum routs to be used along with the most optimum rout to avoid congestion in the most optimal rout. Using this approach instead of having all the traffic between two given nodes in just one rout, we devide the traffic between several routs. Comparing simulation results of MABC and MACO algorithms, we see that in normal situation the two algorithms have relatively same behaviour but in unreliable networks MABC has better effect and comes with more fault-tolerance.

REFERENCES

[1]K. MongSim, W. Hong Sun, (2003) "Ant Colony Optimization for Routing and Load-Balancing: Survey and New Directions" *In IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS*, VOL. 33, NO.5.

[2] K. M. Sim and W. H. Sun,(2002) "Multiple ant-colony optimization for network routing," *in Proc. 1st Int. Symp.Cyberworld, Tokyo, Japan,* pp. 277–281.

[3] Al-Dahoud Ali and Mohamed A. Belal, (2007) "Multiple Ant Colonies Optimization for Load Balancing in DistributedSystems*", ICTA'07*, Hammamet, Tunisia.

[4]D. Karaboga, (2005)"An Idea Based On Honey Bee Swarm for Numerical Optimization*", Technical Report-TR06,Erciyes University, Engineering Faculty, Computer Engineering Department*.

[5]L.P.Wong, M.Y. Hean Low, C.S. Chong, (2008)"Bee Colony Optimization with Local Search for Traveling Salesman Problem" *in Proceeding of the IEEE International Conference on Industrial Informatics(INDIN)*.

[6]A.Baygosuglu,L.Ozbakir,P. Tapkan, (2007)"Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem" *in Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*. pp. 532.

[7]G. D. Caro and M. Dorigo, (1998) "AntNet: Distributed stigmergetic control for communications networks," *J. Artif. Intell. Res.*, vol. 9, pp. 317–365.

[8]G. D. Caro and M. Dorigo, (1997) “AntNet: A Mobile Agents Approach to Adaptive Routing,” *Univ. Libre de Bruxelles, Brussels, Belgium*, Tech. Rep. IRIDIA/97-12.

[9]G. D. Caro and M. Dorigo, (1998) “Mobile agents for adaptive routing*,” in Proc. 31st Hawaii Int. Conf. Systems Sciences, Kohala Coast*, HI,pp. 74–83.

[10] UrmilaBhanja, SudiptaMahapatra, Rajarshi Roy, (2010)“A novel solution to the dynamic routing and wavelength assignment problem in transparent optical networks”, *International journal of Computer Networks & Communications (IJCNC)*, Vol2., No.2.

[11]N. Varela and M. C. Sinclair, (1999)“Ant colony optimization for virtual-wavelength-path routing and wavelength allocation,” *in Proc. Congress Evolutionary Computation, Washington, DC*, pp. 1809–1816.

[12]Sh. Rahmatizadeh, H. Shah-Hosseini and H. Torkaman, (2009)“The Ant-Bee Routing Algorithm: A New Agent Based Nature-Inspired Routing Algorithm*”, Journal of Applied Sciences*, vol 9, issue 5, pp.983-987.

[13] R. Schoonderwoerd, O. Holland, J. Bruten and L.Rothkrantz, (1996)”Ant-based load balancing in Telecommunications Networks”, *Adaptive Behavior,* vol.5,no.2.

[14]Schoonderwoerd, R., Holland, O., Bruten, J., (1997)“Ant-like agents for load balancing in telecommunications networks”, *Proceedings of the First International Conference on Autonomous Agents, ACM Press*, p. 209-216.

[15]T. Stuzle and H. H. Hoos, (2000) “MAX-MIN ant system,” *Future Gener.Comput. Syst. J.*, vol. 16, no. 8, pp. 889–914.

[16]T.D. Seeley, (1995) “The Wisdom of the Hive”, *Harvard University Press, Cambridge*, MA.

[17]B. Baran and R. Sosa, (2000) “A new approach for AntNet routing,” *presented at the Proc. 9th Int. Conf. Computer Communications Networks, Las Vegas, NV.*

[18]T. White, (2000) “SynthECA: A Society of Synthetic Chemical Agents*,” Ph.D. dissertation, Carleton University, Northfield, MN.*

[19] M. Dorigo, G. Di Caro, (1999) “The Ant Colony Optimization MetaHeuristic”, *in Corne D., Dorigo M. and Glover F., New Ideas in Optimization, McGraw-Hill*, ISBN:0077095065.

[20] D. Karaboga, B. Basturk, (2007)“Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems”, *LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing*, Vol: 4529/2007,pp: 789-798, Springer- Verlag, IFSA.

[21] N. Monmarché, F. Guinand& P. Siarry (eds), (2010)"Artificial Ants", *Hardback* 576 pp. ISBN 9781848211940.

[22]A. M. Abdel-Moniem, M. H. Mohamed, and A. Hedar, (2010)“An ant colony optimization algorithm for the mobile ad hoc network routing problem based on AODV protocol”, *in IEEE 10th International Conference on Intelligent Systems Design and Applications* (ISDA 2010), 1332–1337.

[23] ShashankShanbhag, Nico Schwan, IvicaRimac and MatteoVarvello, (2011) “SoCCeR: Services over Content-Centric Routing”, *ICN'2011, Toronto, Canada.*

[24] D. Karaboga, C. Ozturk, (2011) "A novel clustering approach: Artificial Bee Colony (ABC) algorithm", *Applied Soft Computing*, Vol. 11 (1), pp. 652-657.

[25] Omkar, S.N., Senthilnath, J., Khandelwal, R., NarayanaNaik, G., Gopalakrishnan, S., (2011) "Artificial Bee Colony (ABC) for multi-objective design optimization of composite structures", *Applied Soft Computing Journal 11 (1)*, pp. 489-499.

[26] Hsien-Kai Hsin, En-JuiChang, Chih-Hao Chao, An-Yeu Wu, (2010)"Regional ACO-based routing for load-balancing in NoC systems", *NaBIC* pp.370-376.