Load Balancing Using Hybrid ACO – Random Walk Approach

Neha Bhatia, Rohan Kundra, Anurag Chaurasia, and Satish Chandra

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology
Waknaghat, Solan, Himachal Pradesh, India
{neha.juit07,urinrkzone,anurag200788}@gmail.com,
satish.chandra@juit.ac.in

Abstract. Telecommunication and network systems have become more complex in recent years. Routing and optimal path finding are some of the important network problems. Traditional routing methods are not capable to satisfy new routing demands. Swarm intelligence is a relatively new approach to problem solving which provides a basis with which it explores problem solving without providing a global model. A Random Walk approach is similar to a drunkard moving along a sidewalk from one lamp post to another where each step is either backwards or forwards based on some probability. In this paper a hybrid algorithm is proposed that combines Ant Colony Optimization algorithm and Random Walk. The overall time complexity of the proposed model is compared with the existing approaches like distance vector routing and Link State Routing. The new method is found to be better than the existing routing methods in terms of complexity and consistency.

Keywords: Load Balancing, Ant Colony Optimization, Random Walk Routing, Traditional Routing, cover and access time.

1 Introduction

Telecommunication and network systems have become more complex in recent years and the network problems have also been increased. Some of the important network problems are routing and finding the optimal path. Several routing algorithms have been developed to improve routing and cope up with the problems related to networks. Vast complexity of networking problems such as load balancing, routing and congestion requires more efficient methods and techniques to solve these problems [1].

The traditional routing methods are not capable enough to satisfy new routing demands. Increase in number of users and network services force improvement in throughput to satisfy all of the services. This situation has led to development of new routing methods to increase efficiency. Such is the case, for example, of a routing method known as LBR (Load Balancing Routing) based on a load-balancing scheme. This method addresses routing by equally distributing load over all possible paths.

This diminishes the congestion probability in more inexpensive links, improving the overall network performance.

Swarm intelligence is a relatively new approach that solves problem with the help of social behaviour of insects and animals. In particular, ants are the one of the insects which have inspired a number of methods and techniques among which the most successful is the general purpose optimization technique known as ant colony optimization. Despite ACO being a successful algorithm but there are many improvements which can make it more efficient and less complex.

Yet another well known algorithm is Random Walk which is a successful method used for routing that uses random steps to move forward based on transition matrix and Markov chain. Random Walks arise in many models in Mathematics and Physics. It can be used to reach obscure parts of large sets, to generate random elements in large sets, etc. The aim of this work is to use ACO algorithm efficiently with reduced complexity. The proposed algorithm combines two famous algorithms of ACO and Random walk into a hybrid to be used in distributed systems.

2 Load Balancing

Load balancing is a technique to spread work between two or more computers, network links, CPUs, hard drives or other resources, in order to get optimal resource utilization, maximize throughput, and minimize response time. It is useful while dealing with multiple communication links. For example, a company having more than one internet connections always ensures access even if one of the connections fails It means an arrangement is set that second should start its work as first fails .Instead of using only one connection at a time load balancing can help both the connections to work at same time by handling the load and prevent failure due to overload. A program can be designed to select between the connections and can be used. Generally there are multiple routes through the networks in all telecommunication companies. This may result in network congestion leading to the need of load balancing to shift traffic from one path to another and to improve network efficiency.

In such networks, there are number of nodes through which calls are routed**.** Load balancing is distributing the load over all the nodes and minimise lost calls. Shortest routes of calls can be determined in many possible ways.

3 ACO Algorithm

The ant colony optimization (ACO) algorithm is based on the behaviour of ants that control their movement on basis of corresponding pheromone level in their path. These ants deposit pheromone on the ground in order to leave their footsteps as favourable path that should be followed by other ants of the colony. Ants cooperate using an indirect form of communication through pheromone they deposit on the edges of the graph while moving [3]. Many Special Cases of ACO algorithm have been proposed. The three most successful being Ant System, Ant Colony System and

Max-Min Ant System. The scheduled activities and the general algorithm for ACO can be given as follows:

```
1.Initialization:
   Set routing tables and initialize pheromone levels
2.Loop:
   While destination is not reached
      Construct Routing Tables
      Optional Search
      Update Pheromones
   end while
```

3.1 Construct Routing Tables (Pheromone Tables)

In this case Routing tables are replaced by Pheromone tables which tell us the probability of choosing a node in form of pheromone strengths. An n- node network uses n different kinds of pheromones. The entries in the tables are probabilities which influence the ants' decision of choosing the next node on the way to their destination.

3.2 Optional Search

Some actions may be needed before updation of pheromones. Problem specific and/or centralized actions can be implemented by such actions, which cannot be performed by single ants**.** The most prevalent action in the application of optional search to the constructed solutions is the optimization of solutions which can further be used for updating.

3.3 Update Pheromones

This phase of algorithm is implemented to increase or decrease pheromone values based on the impact of the solution whether its good or bad. The process of updating pheromones is quite simple: when an ant arrives at a node, the probability (pheromone) corresponding to that node in the table is increased by following formula:

$$p_{new} = \frac{p_{old}+\Delta p}{1+\Delta p} \tag{1}$$

Here p_{new} is the updated and increased probability and Δp is the increase in probability .For the evaporating ants the other entries in the table of this node are decreased according to:

$$P = \frac{p_{old}}{1+\Delta p} \tag{2}$$

Since the new values sum up to 1, they can again be interpreted as probabilities.

4 Random Walk Algorithm

The random sequence of points which is a result of selecting a neighbour of a node at random , followed by moving to another selected node again at random is a random walk on the graph. It is actually a finite Markov chain [10] that is time reversible.

```
1.Initialization:
    Set parameters, calculate the transition probabilities
    Construct Transition Matrix
2. Loop:
    While termination conditions not met do
        Move to neighbouring nodes based on Matrix
    end while
```

Consider a directed graph G = (V,E), $|V| = n$, $|E| = m$. Each edge (i,j) of the graph has a weight $P(i,j) > 0$. P(i,j) denotes the probability to reach j from i in one step. A Markov chain is a sequence of random variables $X_0, X_1, ..., X_t \in \Omega$ that obey the "Markovian property", that is,

$$\Pr[X_{t+1} = y | X_0 = x_0, X_1 = x_1, ..., X_t = x_t] = \Pr[X_{t+1} = y | X_t = x_t] \tag{3}$$

The transition matrix is used to show the transition probabilities P(i,j) which is computed as follows for a random walk:-

$$P(i,j) = \begin{cases} \frac{1}{deg_{out}(i)} & if\ (i,j) \in E \\ 0 & otherwise \end{cases} \tag{4}$$

Suppose we have the following graph:

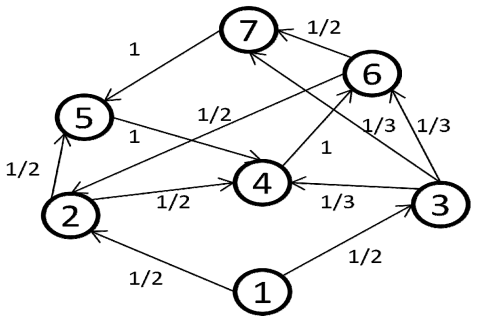


Fig. 1. Sample Graph

Then the transition probabilities and transition matrix is as follows:

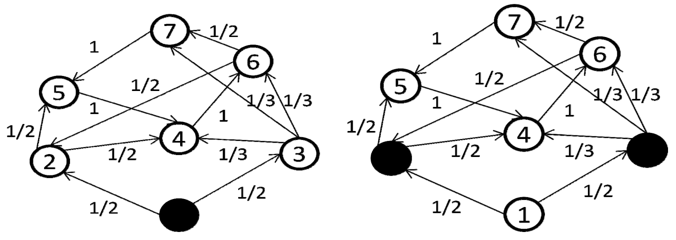
Table 1. Transition matrix

Destination nodes

Source nodes

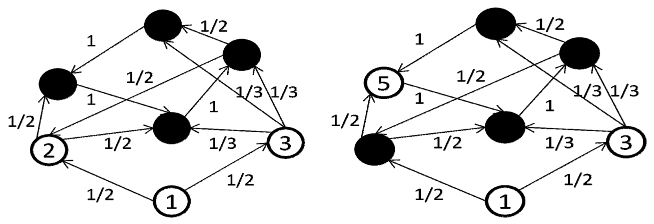
	1	2	3	4	5	6	7
1	0	½	½	0	0	0	0
2	0	0	0	½	½	0	0
3	0	0	0	1	0	1/3	1/3
4	0	0	0	0	0	1	0
5	0	0	0	1	0	0	0
6	0	½	0	0	0	0	½
7	0	0	0	0	1	0	0

Random Walk can be done as follows: (if we need to reach 7)



[1] [1,2/1,3]

Fig. 2a Walk at t=0,1



[124/125/134/136/**137]** [1246/1254/1346/1362/**1367]**

Fig. 2b Walk at t=2,3

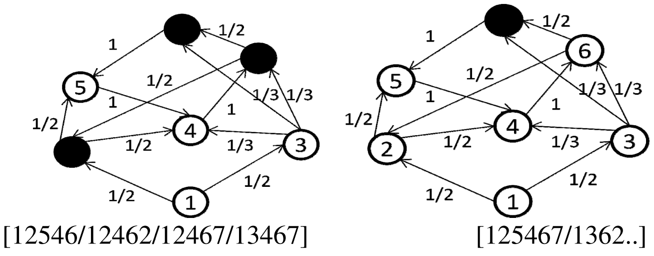


Fig. 2. Random Walk at time instants t=5,6

4.1 Access Time or Hitting Time

It is the expected number of steps before node j is visited,starting from node i.The sum

$$\boldsymbol{\kappa}(\mathbf{i},\mathbf{j}) = \mathbf{H}(\mathbf{i},\mathbf{j}) + \mathbf{H}(\mathbf{j},\mathbf{i}) \tag{5}$$

is called commute time:this is the expected number of steps in a random walk starting at i,before node j i visited and then node i is reached again.There is a way to express access time in terms of commute times.

$$H(i,j) = \frac{1}{2}(\kappa(i,j) + \sum \mathbb{II}(u)[\kappa(u,j) - \kappa(u,i)]) \tag{6}$$

4.2 Cover Time

The expected number of steps from each node to reach every other node is called the cover time.

Cu(G) = E[# steps to reach all nodes in G on walk that starts at u] ; and
C(G) = max Cu(G) :

Cover times for simple graphs:-

- $C(K_n) = \theta(n \log n)$, where Kn is the complete graph on n nodes that included self loops. The bound follows from the coupon collector.

- $C(L_n) = \theta(n^2)$, where Ln is the line graph on n nodes.

- C(n-node lollipop) = $\theta(n^3)$, where an n-node lollipop is a Ln=2 with a Kn=2 at one of the ends.

It turns out that the $\theta(n^3)$ bound is the worst possible for cover time.

5 Proposed Algorithm

The Proposed Algorithm combines ACO and Random Walks in following way:

```
1.Initialization:
    Construct the pheromone tables and transition matrix
    Take a pheromone level as threshold
2.Loop:
    While ant doesn't reach the destination
        If the probability to go to some node is less than threshold then use
        transition matrix probability to move else use pheromone table to move
        Update pheromone tables
    End while
```

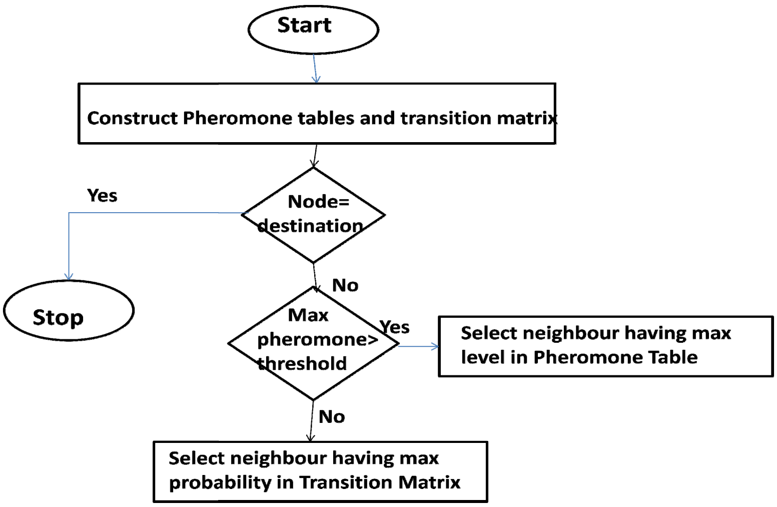

Start
Construct Pheromone tables and transition matrix
Yes
Node= destination
Stop
No
Max pheromone> threshold
Yes
Select neighbour having max level in Pheromone Table
No
Select neighbour having max probability in Transition Matrix

Fig. 3. Flowchart of proposed algorithm

5.1 Proposed Algorithm versus Traditional Routing

The Traditional Routing methods [1] algorithms such as the distance vector routing or RIP and link state routing or OSPF are discussed. Of particular interest are the issues of

1) Dependency on routing table
2) Moving routing table

1. Dependency on routing table

In both RIP and OSPF, a node N_i depends on the routing table created by interaction between all the neighbouring nodes. Further the neighbouring nodes of N_i depend on the routing information of their neighbouring nodes which in turn depend on other neighbouring nodes.

But in our proposed algorithm, the paths are independent of each other and can be chosen without interference and we can move from any source to a destination.

2. Moving routing table

Routing in RIP involves the overhead of moving routing tables of each node N_i to every one of its neighbours. For a large network N, the routing table consisting of costs between different nodes of the graph will be too large. It will increase the overhead to a very high extent due to large size of the routing table to be passed from one node to its neighbours at every step.

In OSPF, routing is achieved by flooding process that is passing a link-state-packet (LSP) to every other node in a network. Although an LSP carrying the cost information is smaller than routing table but the flooding process passes a copy of LSP to all the nodes in the network. Moving through different paths can lead to transmission of multiple identical copies of LSP to the same node.

Routing in Proposed algorithm involves transmission of ants rather than routing tables or by flooding LSPs.

6 Load Balancing in Telecommunication Network

The proposed algorithm can be used in telecommunication network for routing of calls. In addition to calls, the network also supports a population of simple mobile agents with behaviours modelled on the trail laying abilities of ants. The ants move across the network by randomly choosing its neighbours based on the pheromone level and the transition matrix.[13] The distribution of pheromones at each node help decide the path to be followed by any node. Calls between nodes are routed as a function of the pheromone distributions and random walk at each intermediate node.

To determine the route for a call from particular node to a destination, check the largest probability in pheromone table for this destination. If it is greater than threshold value then the neighbour node corresponding to this probability will be the next node on the route to this destination. else the neighbour is selected based on transition matrix. The route is valid if destination is reached and then the call is placed on the network.

In this way, calls and ants dynamically interact with each other. Newly arriving calls influence the load on nodes, which will influence the ants by means of delay mechanism. Ants influence the routes represented by pheromone table and transition matrix which in turn determine the routing of new calls.

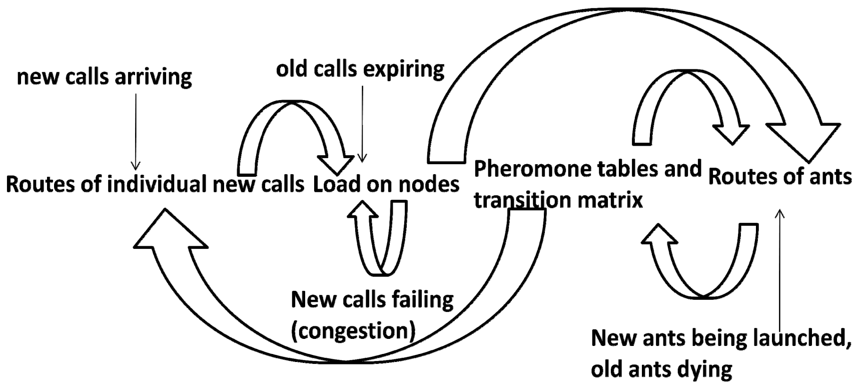

new calls arriving
old calls expiring
Routes of individual new calls
Load on nodes
Pheromone tables and transition matrix
Routes of ants
New calls failing (congestion)
New ants being launched, old ants dying

Fig. 4. Routing paths of the calls

7 Performance

The Proposed algorithm is better than traditional routing methods and has less complexity. It is described as follows using the concept of convergence time which is proved for the ACO before.[17] .The convergence time is average iteration time to reach an optimal solution. The convergence time of the proposed algorithm is less than only applying ACO. Discussing about the convergence time of ACO alone as follows using the proved results:

S(S,Ω,f) is an optimization problem where S is the discrete solution space, Ω is the constraint set, $f\colon S \rightarrow R^{+}$ is the objective function,and R^{+} is the positive real domain. The ACO algorithm finds the optimal solution s* such that f(s*)<= f(s) for $\forall s \in S$.*T* is the pheromone matrix, of which the element τ (*i, j*) is the pheromone trail value of edge (*i, j*).E_{γ} is a measurement of the time complexity of ACO algorithms.

$q(i,t,s^{*}) = P(s_i = c_i^{*}|T, x_{i-1})$ is the crucial probability that an artificial ant selects c_i^{*} as the *i*th component, where s_{i-1}, c_i^{*} ∈ S and $s*$ is the optimal solution (*i* =1,2,3 . . . , *n*).Given *K* ants in an ACO algorithm, the probability that $s*$ is attained at the *t*th iteration time can be calculated according to the following:

$$\begin{aligned} p^{t} &= P\{\xi(t) \in Y^{*}|\xi(t-1) \notin Y^{*}\} \\ &= 1-(1-P(t,s^{*}))^{K} \\ &= (1-(1-\textstyle\prod_{i=1}^{n} q(i,t,s^{*}))^{K} \end{aligned} \tag{7}$$

where,

$Y*$ is the optimal state set if there is at least a solution $s*$
ξ(t) is the state of the stochastic process for the ACO algorithm per iteration.

The Time complexity can be calculated as [17]

$$E_{\gamma} \leq \big(1-\lambda(0)\big)[1-(1-p_{low}^{n})^{K}]^{-1} \tag{8}$$

is held if $q(i,t,s^*) \geq p_{low} > 0$ for $t = 1, 2, \ldots$ and $i = (1, \ldots, n,)$where s^* is the only optimal solution.

The Pheromone value is an important aspect of ACO algorithm so if we want to use it to calculate probability we can do as follows [17]:

$$P(s_{i+1} = c_j | T, x_i) = \begin{Bmatrix} \frac{\tau(x_i, x_j, t)}{\sum_{y \epsilon C \wedge (x_i, y) \in J(x_i)} \tau(i,y,t)}, if < x_i, x_j > \in J(x_i) \\ 0 \qquad , otherwise \end{Bmatrix} \tag{9}$$

Where

J(x_i) is the set of feasible neighbour for $s_{i.}$

C(i,t) is the pheromone rate of the path $<a_{i-1}^*, a_i^* > \in s^*$

The Updation Of pheromones runs *K.n* times per iteration if there are *K* artificial ants.

So we know how probability is dependent upon the pheromone status and in turn the convergence time. Generally the ACO algorithms took exponential time $\approx n^n$ when analyzed. The Probability is better if pheromone level is high.Taking the threshold prevents this exponential time to choose the node. Instead random walk can reduce the time to nlog(n),n^2 or at max n^3 and then further optimality depends upon the ACO algorithm to determine the optimal solution.

8 Summary

In this paper a new method based on hybrid of ACO and Random Walk was proposed. ACO is definitely better than the traditional methods used for routing and load balancing. The proposed algorithm is designed in a way to improve its performance further. The integration of Random Walk increases the probability of always finding the shortest path. Evaporation of Pheromone will decrease the level that can take ants to a wrong neighbor. That might result in choosing the wrong path as the probability of finding optimal solution depends on the pheromone level. At this instant ,when pheromone level is less than the threshold , random walk is one of the optimal algorithms that can help decide the next neighbor and increase consistency. Regarding the complexity integration of random walk , we have seen it reduces the complexity as discussed in performance section earlier. So the proposed method is better than using traditional approaches.

References

[1] Sim, K.M., Sun, W.H.: Ant colony optimization for routing and load-balancing: survey and new directions. IEEE Transactions on Systems Man, and Cybernetics, Part A 33(5), 560–572 (2003)

[2] Dorigo, M., Gambardella, L.M.: Ant Colony System: A cooperative learning approach to the traveling salesman problem. Journal of IEEE Transactions on Evolutionary Computation (1997)

[3] Randall, M., Lewis, A.: A Parallel Implementation of Ant Colony Optimization. Journal of Parallel and Distributed Computing 62(9), 1421–1432 (2002)

[4] Dorigo, M., Birattari, M., Stutzle, T.: Ant Colony Optimization. IEEE Computational Intelligence Magazine (2009)

[5] Mukherjee, D., Sriyank: Ant Colony Optimization Technique Applied in netwrk Routing Problem. International Journal of Computer Application 15, Article 13 (2010)

[6] Hung, K.-S., Su, S.-F., Lee, Z.-J.: Improving Ant Colony Optimization Algorithms for Solving Traveling Salesman Problems. Journal of Advanced Computational Intelligence and Intelligent informatics 11(4) (2007)

[7] Liu, H., Li, P., Wen, Y.: Parallel Ant colony Optimization Algorithm. In: Proceedings of the 6th World Congress on Intelligent Control and Automation, China (2006)

[8] Zhao, D., Luo, L., Zhang, K.: An Improved Ant Colony Optimization for communication network routing problem. In: Bio-Inspired Computing, BIC-TA (2009)

[9] Burioni, R., Cassi, D.: Random walks on graphs: ideas, techniques and results. Journal of Physics A: Mathematical and General 38(8) (2005)

[10] Lawrence, R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE 77 (1989)

[11] Aldous, D.: An introduction to covering problems for random walks on graphs. Journal of Theorotical Probability 2(1), 87–89 (1989)

[12] Hildebrand, M.V.: A survey of results on random random walks on finite groups. Probability Surveys 2, 33–63 (2005)

[13] Tian, H., Shen, H., Matsuzawa, T.: Random Walk Routing for Wireless Sensor networks. In: Proceedings of the Sixth International Conference on Parallel Computing, Applications and Technologies (2005)

[14] Kim, S.-S., Smith, A.E., Hong, S.-J.: Dynamic Load Balancing Using Ant Colony Approach in Micro-cellular Mobile Communication Systems. In: Advances in Metaheuristics for Hard Optimization, pp. 137–152 (2008)

[15] Schoonderwoerd, R., Holland, O., Bruten, J., Rothkrantz, L.: Ant-Based Load Balancing in Telecommunications Networks. Adaptive Behaviour 5(2), 169–207 (1996)

[16] Blum, C.: Ant Colony Optimization:Introduction and Hybridizations. In: 7th International conference on Hybrid Intelligent Systems, HIS 2007, pp. 24–29 (2007)

[17] Gutjahr, W.J.: First steps to the runtime complexity analysis of ant colony optimization. Computers and Operations Research 35(9), 2711–2727 (2008)

[18] Huang, H., Wu, C.-G., Hao, Z.-F.: A Pheromone-Rate-Based Analysis on the convergence Time of ACO Algorithm. IEEE Transactions on Systems,Man and Cybernetics-Part B:Cybernetics 39(4) (2009)