A New Hybrid Approach to OSPF Weight Setting Problem

Nikhil Hemant Bhagat
Department of Telecommunications Engineering
University of Colorado at Boulder, Colorado, USA
nikhilhbhagat@gmail.com

***Abstract:-* Routing protocol is considered to be the backbone system (spinal cord) of any network. It helps in directing the data from the source towards the destination, using some unique best path selection algorithm. OSPF is considered to be one of the best routing protocols for intra-domain routing. It uses the shortest path algorithm in determining the best path for its routing process. The routing decision completely depends upon the weights (cost) assigned to each link by the network operator. A famous network device vendor Cisco calculates OSPFs link weight as the inverse of the links bandwidth by default. It doesn't matter how one calculates the weight, unless the network capacity is used to its optimum level. It has been observed that the weight setting algorithm is not yet optimized to consider the projected demands. Thus, the quality of this routing protocol decision is still not reached to the peak as there is no optimal setting to resolve this issue. The paper addresses the weight setting problem and how it affects the routing decision. Furthermore, it studies the present algorithms used to solve this issue and states their disadvantages. A new hybrid approach to currently used genetic algorithm is proposed. Demonstration of this proposed algorithm is presented through diagrams and flowcharts.**

***Keywords* – OSPFWSP; Genetic Algorithm; Local Search; Hybrid Algorithm.**

*****

I. INTRODUCTION

The growth in the internet has been increasing tremendously day by day. Statistics reveal that there has been almost thrice boost in the traffic generation every year from the year 1996 till date [1]. Internet Service Providers (ISPs) try to meet this growing demand consisting of new technologies, with capacity expansion of their existing network links [1]. We also expect this traffic rate to grow more in the near future. Also, there is a growing pressure on ISPs to provide a quality of service (QoS) that considers the parameters such as throughput, loss and delay [2]. Due to high internet usage for a variety of purposes, the growing data traffic finally results in proving the importance of internet traffic engineering which states the effective use of the existing network resources [1].

Open Shortest path First (OSPF) is considered to be the best backbone routing protocol used in internet [3]. It is a link state protocol, which meansthe whole topology information is contained in the link state database of every OSPF running router [4]. OSPF routing decision is based on the shortest path algorithm. Each link in the topology is assigned a positive integer value ranging from 2^0 to 2^{16} i.e. from 1 to 65535 referred to as weight [4,5]. This is fixed parameter assigned by the network operator to a specific link. The traffic will flow naturally through the link with the minimum weight [6]. So, the network operator has to keep all the parameters and set the weights accordingly. These weight setting decisions are made purely on the basis of an assumption that the topology will not have a link failure [6]. This creates a problem when a link fails because the routing protocol has to converge via different available path which may distribute the network traffic unevenly. This might create network congestion on a low capacity link due to the fact that the routing paths are no longer same as the original path. The demand and the optimized weights that were assigned for the original topology will no longer remain good for the new converged topology after the link failure. Due to the absence of that particular link, the potential traffic which was first directed from that link would now flow through other links. As these new converged network links were not optimized for that particular traffic, inefficient mapping of traffic on the available links would be the result.This would cause a potential threat to the network when there is a larger demand on the converged low capacity links. Cisco being one of the larger network device vendors does not consider the traffic demands while calculating the weights [4]. If we continue such type of weight assigning mechanism, we won't ever guaranteethat the network would run effectively.

II. OSPF WEIGHT SETTING PROBLEM

To understand the OSPF weight setting problem let us consider a directed graph or a multi-graph G = (N, A) where [5, 7],
N = Nodes that refer to routers, and
A = Arcs that refer to links.
Also, each arc 'a,' defines a link which has a capacity C(a), where a ϵ A [2].
Let us consider a demand matrix D which is a function of (s, t) where,
s = source of demand, and
t = destination of demand.
Demand matrix D gives information regarding how much traffic we need to send from the source to the destination.
Now, the normal case [2, 6],
D(s, t) = 0, if there is no path from source's' to destination 't.' ……………………………………………………...(1)
D(s, t) ≠ 0, if there are many paths from source 's' to destination 't.'………………………………………….(2)
The network operator now needs to distribute the traffic flow on each link which is a challenge [2]. Thus, the general routing problem states that it is assumed that there are no limitations on how to distribute the traffic on paths from s to t [8]. The above stated demand matrix does not take care of the sudden demand variation. Thus, this matrix can be called as an estimated demand matrix. So, we may consider two demand variations one for the peak hours and another for the light traffic hours traffici.e. a prediction is made. In order to find out the proper utilization on each link we will now declare some functions,

L(a) = the load on each link 'a' with capacity C(a), or we can also define as L(a) is the sum of all the demands of the amount of flow on the link 'a.' Then the overall utilization will be L(a)/C(a) [2]. To define the cost function our objective is to keep load less than the actual capacity i.e. L(a) should never exceed C(a).
Then the cost function can be written as [1, 2, 6],

$$\Phi = \sum_{a \in A} \Phi_a(\ell(a))$$

where for all $a \in A$, $\Phi_a(0) = 0$ and

$$\phi'_a(x) = \begin{cases} 1 & \text{for } 0 \leq x/c(a) < 1/3 \\ 3 & \text{for } 1/3 \leq x/c(a) < 2/3 \\ 10 & \text{for } 2/3 \leq x/c(a) < 9/10 \\ 70 & \text{for } 9/10 \leq x/c(a) < 1 \\ 500 & \text{for } 1 \leq x/c(a) < 11/10 \\ 5000 & \text{for } 11/10 \leq x/c(a) < \infty \end{cases}$$

A small modification is made on the last part i.e. on the last line of the above formula [5].

$$\Phi(x) = n \times 500 \quad \text{for } 11/10 < x/c(a) < \infty$$

n = number of links.

This change enables us to figure out if the congestion is occurred or not. The congestion can occur only if [5],

$$\Phi > T(n \times 500)$$

n = number of congestion links. Thus the updated cost function will give more information regarding congestion.

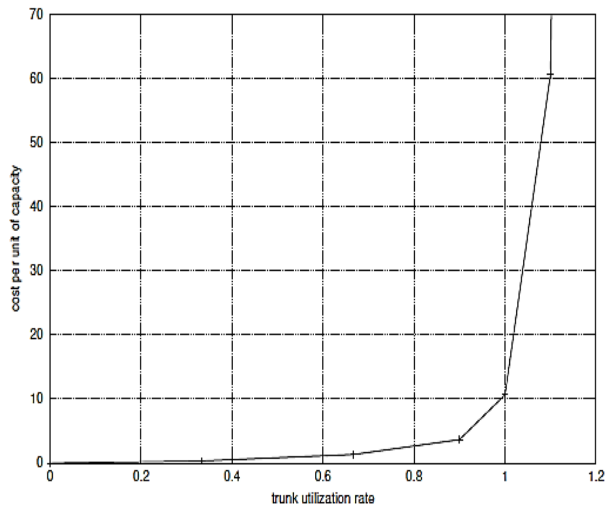

Fig. 1 trunk utilization rate v/s cost per unit of capacity [1]

The above graph is plotted as trunk utilization rate v/s cost per unit of capacity. The above graph illustrates an exponentially growing curve. It states that as the cost function increases with respect to the growing demand, it is less expensive to direct traffic on the light traffic link

over the loaded link [9]. The cost function increases in accordance with the effective traffic increase thus leading to an optimization of utilization rate equal to 1. The congestion occurs when the load exceeds the utilization rate i.e. utilization rate > 1.
To formulate the general routing problem, let's define a term $f_a^{(s,t)}$ which means that for each pair (s, t), $f_a^{(s,t)}$ refers to the amount of traffic flown from s to t over a link 'a' [2, 10]. Also, we will use ϕ_a which refers to linear cost function on arc 'a.' Thus, the general routing problem can be formulated as [1, 5, 11],

$$\min \Phi = \sum_{a \in A} \Phi_a$$

Subject to,

$$\sum_{x:(x,y)\in A} f_{(x,y)}^{(s,t)} - \sum_{z:(y,z)\in A} f_{(y,z)}^{(s,t)} = \begin{cases} -D(s,t) & \text{if } y=s, \\ D(s,t) & \text{if } y=t, \\ 0 & \text{otherwise,} \end{cases} \qquad y,s,t \in N, \quad (1)$$

$$\ell(a) = \sum_{(s,t)\in N\times N} f_a^{(s,t)} \qquad a \in A, \quad (2)$$

$$\Phi_a \geq \ell(a) \qquad a \in A, \quad (3)$$

$$\Phi_a \geq 3\ell(a) - \tfrac{2}{3}c(a) \qquad a \in A, \quad (4)$$

$$\Phi_a \geq 10\ell(a) - \tfrac{16}{3}c(a) \qquad a \in A, \quad (5)$$

$$\Phi_a \geq 70\ell(a) - \tfrac{178}{3}c(a) \qquad a \in A, \quad (6)$$

$$\Phi_a \geq 500\ell(a) - \tfrac{1468}{3}c(a) \qquad a \in A, \quad (7)$$

$$\Phi_a \geq 5000\ell(a) - \tfrac{19468}{3}c(a) \qquad a \in A, \quad (8)$$

$$f_a^{(s,t)} \geq 0 \qquad a \in A;\ s,t \in N. \quad (9)$$

All the above constraints can be described as follows [2]:
Constraint (1) ensures the routing of the desired data traffic from source to destination known as the flow conservation constraint [2].
Constraints (2) describes the actual load on every arc 'a,' and [2],
Constraints (3) - (8) refers to the cost on every arc according to the linear cost function ϕ . A complete linear program formulation of the stated routing problem is seen in the above program [2, 11].

III. EXISTING ALGORITHMS USED TO SOLVE OSPFWSP

Three of the main algorithms used to solve OSPFWSP are genetic algorithm, local search and simulated annealing. The known algorithm which was used a lot was the genetic algorithm [1]. Genetic Algorithm is empirically proven best robust search technique which is derived from the evolution theory. It is based on the Darwin's theory of survival [1]. So, what Genetic Algorithm does is it by default transforms a fair population in one generation associated with a fitness value into a new generation of population [13]. As, per the name suggest 'Genetic,' it uses a kind of a mutation technique (selection and reproduction process), thus creating anew generation approximation. This algorithmis a two-step process which is as follows [1]:

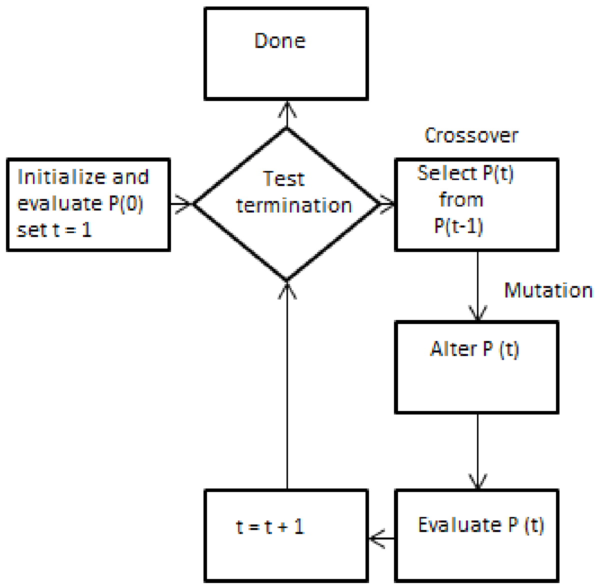

Fig.2 Genetic Algorithm Approach [12]

Step 1: Initialize a random population of P(0) where P(t) is the population at time t [1].
Step 2: Until the termination criteria are fulfilled iterate the below steps on the current generation [1].
(i) Use fitness function to assign a fitness value to each individual in the current generation.
(ii) Select parent for the mating process.
(iii) Using the mutation and crossover technique in order to form children from the above selected parents.
(iv) Finally, the best individual should be identified for the iteration of this genetic algorithm.

Now, let's see how this algorithm was actually used to solve the OSPFWSP.

For representation we will use the weights $w = w_1, w_2......w_{|A|}$, where$w_i \epsilon$ $[1, 65535]$for all the arcs $i = 1,....|A|$. All points here refer to the practical solutions but instead of using the maximum available upper weight limit 65535 we will use the upper limit assigned by the user i.e. the network operator [1]. As per the algorithm we now initialize a random population of individuals choosing from the well-defined search space [1, 65535]. All these solutions are represented by integer vectors. In addition to these vectors, we introduce two more vectors *UnitOSPF* and *InvCapOSPF*and add them to the above integer vectors.*UnitOSPF* represents unit vector whereas *InvCapOSPF* refer to the weight vector which is inversely proportional to the arc capacity [1]. WE now go to the step (ii) which defines the evaluation phase. This phase is more complicated as it requires the OSPF routing process to determine the load on every arcfrom the given weights. We assign a fitness value to each arc using the fitness function i.e. we assign a cost to every arc using a cost function ϕ [1]. This means that the fitness value is the same as the cost value. Now, the user defined weight setting will completely give us the shortest path to determine the OSPF routing by considering the lowest cost to be better. We will now follow the Fortz and Thorup algorithm's procedure to evaluate the above factors [14]. The arc utilization derived from the total load on that arc will finally give us the cost of that arc. Fitness value is thus defined as the total cost for all the available arcs. In order to compute the cost function we first need to compute the arc load L(a) from the given weight setting which turns to be complicated. This algorithm gives a five step approach to calculate the arc's load. To begin with, we consider one destination t at a time, for all the demand pairs $d_{st} \epsilon$ D. This will give the arc load for that particular destination denoted asl_a^t, t ϵ N' where N' is the set of the destination nodes [1, 14].

Step 1: Using the Djikstra's shortest path algorithm, we compute the shortest distance $d_u^t$to the destination t from a node u. In order to compute the distance to t, we need to apply this algorithm by reversing all the paths on the graph.

Step 2: Computing all the set of arcs A^t to the destination node t which lie in the shortest path as [1],

$$A^t = \{(u,v) \epsilon \ A : d_u^t - d_v^t = w_{(u,v)}\}$$

Step 3: For each node u [1],

$$\delta_u^t = |\{v \in N : (u,v) \in A^t\}|.$$

Where, δ_u^t denotes its out degree in G^t.

If $\delta_u^t > 1$, it means at node u the traffic flow is split [1].

Step 4: As these are the partial loads, we can compute them as [1],

(i) We visit node v, in the decreasing order of distance to t and v ϵ N.

(ii) Also, when we visit node v, then for all the nodes (v, w) ϵ A^t, we set [1]

$$l_{(v,w)}^t = \frac{1}{\delta_v^t}(d_{vt} + \sum_{(u,v)\in A^t} l_{(u,v)}^t).$$

Step 5: Finally, we sum up all the loads to get the actual load on the arc 'a [1].'

$$l_a = \sum_{t\in \bar{N}} l_a^t.$$

In order to compare the costs for different size of networks and topologies, we will apply the normalized scaling factor applied by Fortz and Thorup[1, 2]. In order to get the actual scaled cost for the complete arc we need to divide the basic cost function with the incapacitated measure of the cost function [1].

$$\Phi^* = \Phi / \Phi_{uncap}.$$

where,

$$\Phi_{uncap} = \sum_{(s,t)\in N\times N} d_{st} \cdot h_{st},$$

where, $h_{st}$is the distance measured from the nodes s to t with respect to hop count (unit weights) [1]. In case if $\phi^* = 1$, it implies that the data traffic is routed to the destination using the unit weight shortest path where all the loads in the path stay below 1/3rd of the links capacity. After we derive the cost function and are able to separate the individuals with respect to their costs (fitness values), we partition the individuals in three groups upper class, middle class and the lower class [1]. In parent section process, class A is transferred unchanged to the next generation. Class C is replaced by the solutions generated randomly. Next generation class B is obtained by using the crossover principle between one elite parent $P_1$from upper class A and another non-elite parent P_2 from either class B or C [12, 15].

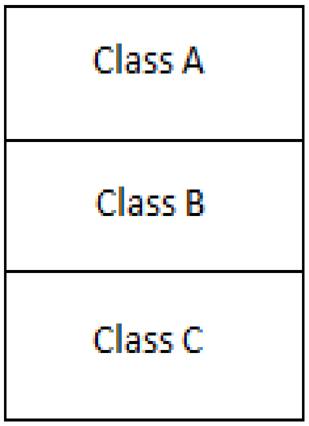

Fig.3 Individual partition into three categories [12].

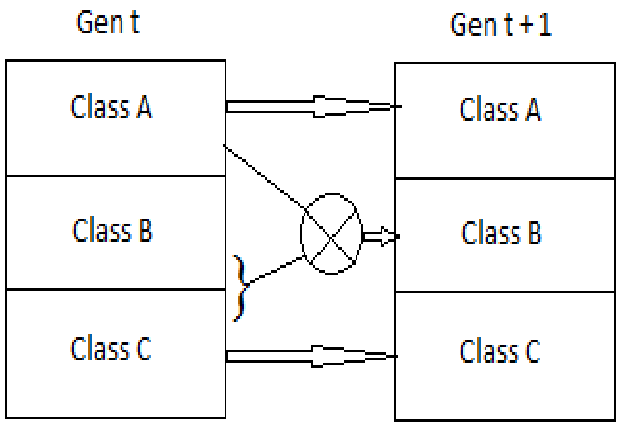

Fig.4 Parent selection [12]

The two vital principles followed by this parent selection process are [5, 12]:

(i) Reproduction is more likely to happen in better individuals.

(ii) Better individuals have the potential to breed more than once so reselection is allowed.

Bean in 1994 proposed crossover technique in which two parents P_1 and P_2 are combined which is called as random keys [16]. In order to find whether the gene is inherited from P_1 or P_2, generate a random variable 'r' which lies between 0 and 1 and also generate a cut off variable K which would lie between 0.5 to 1 [17].After breeding a child is formed 'c.'

```
for all genes $i = 1, \ldots, |A|$ do
    if $r[i] < K$
        $c[i] = p_1[i]$
    else
        $c[i] = p_2[i]$
end.
```

It has been proved experimentally that the best convergence results are obtained when cut off variable K = 0.7 But the main problem noticed is that it gives a premature convergence to local optima.

IV. DISADVANTAGES OF CURRENT EXISTING ALGORITHMS

Until a time bound is found or a solution deemed optimal is found, local search move from one solution to the other solution in the search space [18,19]. The problem with the local search is that, it gets stuck in local minima. Genetic Algorithm also gets stuck in the same local minima [5, 18]. The local minima problem can be shown by a graphical representation as below.

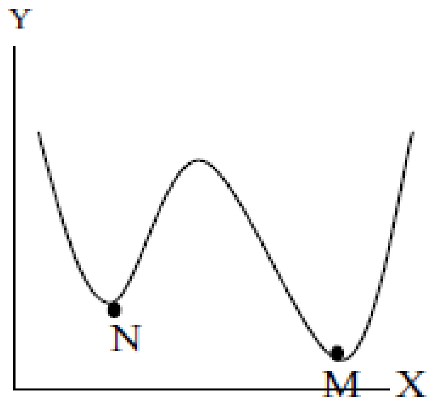

Fig.5 Local minima problem [5].

In the above figure, we can clearly see that though there is an optimal solution at point 'M,' due to local minima problem we are able to see that the optimal solution is at N which is actually not the optimum solution.

V. A NEW HYPRID APPROACH TO OSPFWSP

In order to make the genetic algorithm more efficient, a new approach is used in which a new step is included. The efficient uselocal search algorithm with the genetic algorithm is found to be remarkable. Local search is used in order to help in improvement of individual population selection [12].

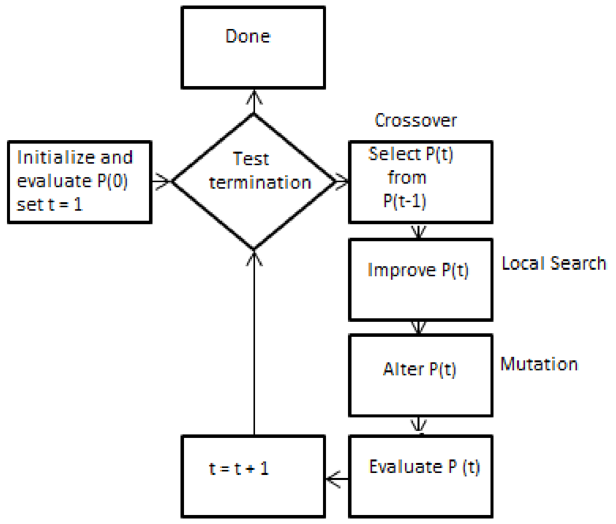

Fig.6A hybrid approach to OSPFWSP [12]

Here, after the crossover a fast local search is used in order to obtain a better individual in the next generation of class B [12].

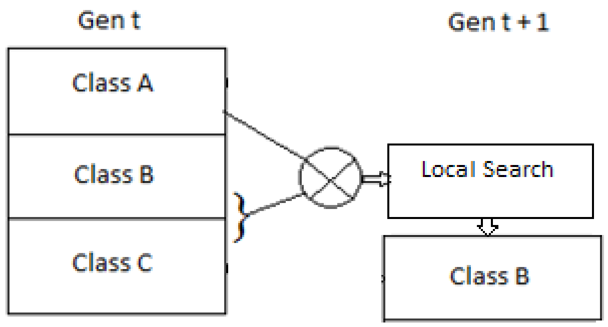

Fig.7 Crossover + Local search [12]

The fast local search will work as follows:
Step 1: Consider a topology from to source to destination containing five arcs A', where a ϵ A having large ϕ_a values [12].
Step 2: This step includes scanning of all the arcs from the highest cost to the lowest cost ϕ_a[12].

(i) Slowly start increasing the arc weight, one unit at a time, which is in range of $[w_a, w_a[(w_{max} - w_a)/4]]$

(ii) Look at the end cost, if the cost ϕ is reduced, then restart the local search.

The next phase is the final phase of determining the dynamic shortest paths. When the arc weight starts to increase, in local search then [1, 12],

(i) The shortest path tree may completely change which is a rare case [12].

(ii) The shortest path tree may remain completely unchanged which happens in the case when the arc is not present in the shortest path tree [12], or

(iii) The shortest path tree may change partially i.e. few trees change or a small portion of a particular tree changes [12].

For (iii), it is not proper to recompute the trees from the beginning. This algorithm serves to be a more efficient algorithm than the simple genetic algorithm.

VI. CONCLUSION

The paper studies the OSPF weight setting problem and how it affects the routing decision. The analysis highlights insightful features of the current algorithms available to solve OSPFWSP. Furthermore, a careful examination of the currently used genetic algorithm is done. A new hybrid approach is proposed which includes a mixture of both the algorithms Genetic Algorithm and Local Search. Finally, explanation of how this hybrid approach can serve as the best algorithm is presented.

VII. ANALYTICAL ASSESMENT

I have been working on this project, and have found certain loopholes which are present in the current algorithms. A new approach to this OSPFWSP is proposed but have not yet implemented practically. This approach has already been started by a researcher but has not yet come to a perfect conclusion. If implemented successfully, this algorithm will prove to be the best algorithm for this OSPFWSP.

VIII. FUTURE WORK

I am still working on the project and the progress is found to be remarkable. Practical implementation of this algorithm plus new changes in order to increase the potentiality of this algorithm is something which has to be done. I haven't thought of using simulated annealing in this approach. So, a careful study of simulated annealing might be helpful in this new algorithm.

REFERENCES

[1] M. Ericsson, M.G.C. Resende, P.M. Pardalos.“A genetic algorithm for the weight settingproblem in OSPF routing.” P.1-35. Oct 9, 2001.

[2]B. Fortz, M. Thorup, "Intemet traffic engineering by optimizing OSPF weights" in Proc. INFOCOM, pp. 519-528, Mar. 2000.

[3] F.Bizri. B Sanso, “Corouting: an IP Hybrid Routing Approach**”,** Networking and Services, 2008. ICNS 2008. Fourth International Conference On page(s): 46-52, 16-21 March 2008

[4] Y.Zuo, J.Pitts**,”**Impact of Link Weight Ranges on OSPF Weight Solutions”, Communications and Networking in China, 2007. CHINACOM '07. Second International Conference, Shanghai, 22-24 Aug. 2007, On page(s): 72-76

[5] A.A. Ghazala, A. El-Sayed, M. Mosa. “A new approach for Open Shortest Path Weight Setting Problem (OSPFWSP)” p.188-193.Doi: 10.1109/ICCIT.2008.44

[6] M. Squalli, S. Sait, S. Asadullah. “OSPF weight setting optimization for single link failures.”*International Journal of Computer Networks & Communications (IJCNC)* Vol.3, No.1, January 2011.

[7] M Zagozdzon, M Dzida, M Pioro,” Traffic Flow Optimization in Networks with Combined OSPF/MPLS Routing **“,**Advanced Computing and Communications, 2007. ADCOM 2007. International Conference 18-21 Dec., On page(s): 131-137, Guwahati, Assam

[8] D. O. Awduche, J. Malcolm, J. Agogbua, M. O’Dell, and J. McManus, “Requirements for traffic engineering over MPLS,” Network Working Group, Request for Comments, http://search.ietf.org/rfc/rfc2702.txt, September 1999.

[9] Bernard Fortz&MikkelThorup (2003) “Robust optimization of OSPF/IS-IS weights”, *Proceedings of the International Network Optimization Conference*, 225-230.

[10] Mohammed H. Sqalli, Sadiq M. Sait& Mohammed AijazMohiuddin (2006) “An Enhanced Estimator to Multi-objective OSPF Weight Setting Problem” *Network Operations and Management Symposium, NOMS.*

[11] Ahmed Abo Ghazala, Ayman EL-SAYED, and Mervat MOUSA, "A Survey for Open Shortest Path First Weight Setting (OSPFWS) Problem", The 2nd International Conference on Information Security and Assurance (ISA2008), Busan, Korea, 24-26 April 2008

[12] M. Resende. “A memetic algorithm for the weightsetting problem in OSPF routing.”*INFORMS Conf. on Telecommunications Boca Raton, Florida*, March 13, 2002.

[13] D.E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning.Addison-Wesley, 1989.

[14] B. Fortz and M. Thorup.Increasing internet capacity using local search.Technical report,AT&T Labs Research, 2000. Preliminary short version of this paper published as \InternetTra_c Engineering by Optimizing OSPF weights," in Proc. IEEE INFOCOM 2000 { The Conference on Computer Communications.

[15] R.B. Hollstein. Arti_cial genetic adaptatiion in computer control systems.PhD thesis, Universityof Michigan, 1971.

[16] J.C. Bean. Genetic algorithms and random keys for sequencing and optimization.ORSA J.on Computing, 6:154{160, 1994.

[17] J.R. Koza, F.H. Bennett III, D. Andre, and M.A. Keane.Genetic Programming III, DarwinianInvention and Problem Solving.Morgan Kaufmann Publishers, 1999.

[18] B. Fortz and M. Thorup, "Increasing Intemet capacityusing local search", Computational Optimization and Applications, 29(1): 13-48,2004.

[19] Bernard Fortz&MikkelThorup (2000) “Increasing Internet Capacity Using Local Search”, *Technical Report IS-MG.*

BIOGRAPHY

Nikhil HemantBhagatis currently pursuing his Masters of Science in Telecommunications from University of Colorado at Boulder, USA. He completed his B.E in Electronics and TelecommEngineering from Mumbai University, INDIA in 2012. He has also completed CompTIA'sNetwork + technician and PC technician certifications from NIIT in the year 2010. He became Microsoft Certified in Dec. 2010 and was regarded as Microsoft Certified Application Specialist 2007. He underwent practical experience in Mobile Communications from Mahanagar Telephone Nigam Ltd. (Govt. of India), Mumbai in June-July 2011. He went deep into Computer Networks by acquiring Cisco certifications like Cisco Certified Network Associate (CCNA Routing & Switching) and Cisco Certified Network Professional (CCNP Routing & Switching) in the year 2011. He further assimilated couple of certifications like Windows XP Professional and IT Technology Professional 2010 from Ranksheet.com and Juniper Network Certified Internet Associate in 2013.. He is currently targeting Cisco Certified Internet Expert (CCIE Routing and Switching) certification. His areas of interests are Computer Networks, Network optimization and Data analysis.