Open Shortest Path First Weight Setting (OSPFWS) solving algorithms comparison and new method for updating weights

Ahmed Mahmoud Abo Ghazala
National Bank of Egypt
1187 Kornish el Nile, Cairo, EGYPT
Communication DEP

Ayman EL-SAYED (IEEE Member)
Computer Science and Engineering Dept.
Faculty of electronic Engineering,
Menouf 32952, EGYPT

Abstract

Open shortest path first is the most commonly used intera-domain routing protocol. It selects the paths which traffic is routed within autonomous systems. OSPF calculates routes as follow. Each link is assigned weights by operator. Each node within the autonomous system computes shortest paths and creates destination tables to route data to next node along the path to its destination. Shortest paths are selected according to path cost. Path cost is determined by sum of its weight links. Then link weights determine the shortest paths, which in turn determine the routing of network traffic flow. OSPF weights setting problem is to find a set of OSPF weights that optimizes network performance. OSPF weights setting problem is an NP-hard problem.

In the last couple of years, various algorithms for OSPF weights setting problem have been proposed. In this paper, we present a survey of OSPF weights setting algorithms and compare their performance; also new method for updates weights is introduced.

1. Introduction

Traffic engineering aims to make more efficient use of network resources. Although it does not increase the bandwidth of the network, it enables us to transmit more demands. With the growth of the internet, traffic is approximately doubling each year [1]. These developments highlight the important of traffic engineering. OSPF runs at most IP large network. OSPF directs traffic based on dimensionless static link weights.

OSPF requires nodes to exchange routing information with all other nodes in the AS. Each node has to have complete network topology knowledge for computation of shortest paths.

OSPF mandates that each node computes a graph of shortest paths with it is self as a root [2]. This graph gives the least weight routes to all destinations nodes. In the case of multiple shortest paths, OSPF uses load balancing and splits the traffic flow over several shortest paths which leaving from each router [3].

Weights are assigned by the network operator. It is ranging from 1 to 65535. The lower the weight, the greater the chance that traffic will be routed on that link. Cisco, a leader router vendor, by default assigns OSPF weights inversely with link capacity, and then the shortest paths will be the links of the higher bandwidths.

Because CISCO does not take into account traffic demands when calculates weights. This type of assigning weights does not guarantee that the network will run efficiently. Then OSPFWS problem is determining weights to be assigned to links so as to optimize a cost function typically associated with network utilization measure.

We classified the solutions into four types according to used algorithm for solving. The first type is local search algorithm, the second type is genetic algorithm, the third type is simulated annealing algorithm and, the fourth type is hybrid genetic algorithm (genetic algorithm with local search algorithm).

This paper is organized as following. Section 2 is formulation of the problem. Section 3 describes local search method. Section 4 describes genetic algorithm method. Section 5 describes simulated annealing method. Section 6 describes Hybrid genetic algorithm method. Section 7 is comparison between all methods. Section 8 is the proposed solution, at the end, the conclusion exists.

2. Problem Formulation

The general routing problem is defined as follows. If the network is represent in a directed graph, G (N, A) whose nodes and arcs represent routers and the links between them. Each arc (a) has a capacity C (a) which is the amount of traffic flow that the link can carry. We have a demand matrix D that for each pair S and T of nodes tells us how much traffic send from S to T where S as the source and the T as the destination of the demand. Many of the entries of demand matrix D may be zero, and in particular, D (S, T) = 0 if there is no traffic flow from S to T in G (N, A).

The general routing problem is defined like that" there are no limitations to how the flow demand can be distributed between the paths from S to T".

In order to evaluate the effectiveness of the different solutions to OSPFWS problem a linear multi-commodity flow routing problem (general routing problem) is solved with piecewise linear cost function, we can formulate the general routing problem as follow

Let L (a) is the sum over all demands of the amount of flow for that demand which is sent over link a. then utilization of link a is L(a)/C(a)

The objective function here is L(a) not excess C(a) for avoid congestion then the cost functions Φ is sum of all costs of links at G(N,A)

The cost function can be formulated as in [4]

$$\Phi = \sum_{a \in A} \Phi_a\,(L(a)) \quad \text{FOR all } a \in A\ ,\ \Phi_a\,(0) = 0$$

Fig1 illustrate graph between the cost function and the demand load it approximates an exponentially growing curve the figure depicts that it will be more expensive to direct traffic over loaded link as the cost function increases with increasing the traffic load and the cost increases dramatically as the utilization arrive up to 1. After that congestion can be occurs if utilization >1

Then optimal routing problem can be formulated as

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

subject to

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

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

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

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

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

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

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

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

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

Where $\boldsymbol{f_a^{(s,t)}}$ is the traffic demand that flow from node *s* to node t and pass on link *a*. The above formula is used in most of solutions to OSPFWS problem that can be used for only comparison. In the next section we begin to discuss the different algorithms used to solve OSPFWS problem.

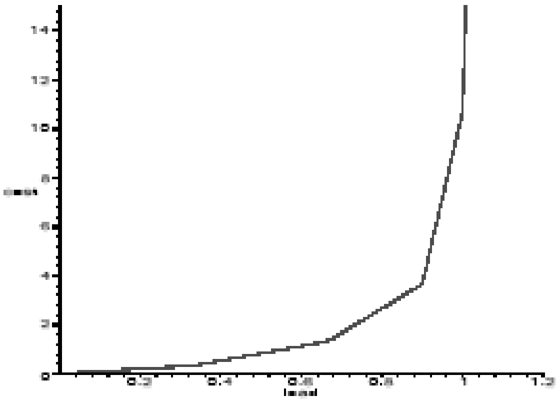

Figure 1, Evolution of cost function with link load

3. Local Search Method

Local search is a metaheuristic for solving computationally hard optimization problems.

Local Search also referred to as Neighborhood Search or Hill Climbing which is the basis of many heuristic methods for combinatorial optimization problems. In isolation, it is a simple iterative method for finding good approximate solutions. The idea is trial and error.

Local search can be used on problems that can be formulated as finding a solution maximizing or minimizing a criterion among a number of candidate solutions. Local search algorithms move from solution to solution in the space of candidate solutions (the search space) until a solution deemed optimal is found or a time bound is elapsed [5]

A local search algorithm starts from a candidate solution and then iteratively moves to a neighbor solution. This is only possible if a neighborhood relation is defined on the search space.

Typically, every candidate solution has more than one neighbor solution; the choice of which one to move to is taken using only information about the solutions in the neighborhood of the current one, hence the name local search.

When selecting the neighbor solution is done by taking the one locally maximizing the criterion, the metaheuristic takes the name hill climbing. Termination of local search can be based on a time bound. Another common choice is to terminate when the best solution found by the algorithm has not been improved in a given number of steps. Local search algorithms are typically incomplete algorithms, as the search may stop even if the best solution found by the algorithm is not optimal. This can happen even if termination is due to the impossibility of improving the solution, as the optimal solution can lie far from the neighborhood of the solutions crossed by the algorithms

For the purposes of explaining local search, we will consider the following definition of a combinatorial optimization problem.

A combinatorial optimization problem is defined by a pair (X, F), where X is the set of all feasible solutions (i.e. solutions which satisfy the problem constraints) and F is the objective function that maps each element x in X to a real number. The goal is to find the solution x in X that minimizes the objective function F. The problem is stated as: *min F(x)*

Local search can be used in OSPFWS problem. We can take Benard and Fortez [6] as an example for using local search to solve OSPFWS problem they developed efficient algorithm using local search heuristic. the aim is determining a weight vector values that achieve the object function.

The differences between local search heuristics arise essentially from how to select neighborhood, the way it is explored, and the choice of the next solution from the neighborhood. Descent methods consider the entire neighborhood, select an improving neighbor and stop when a local minimum is found.

They starts with randomly generated weights values and defined it is moves up on one of two operations applied to ***w***

1) Single weight change. They change one weight and start in process
2) Evenly balancing flows. They set the weight in order to split the flow of data as evenly as possible between different arcs.

Local minimum is drawback of local search algorithm where the neighborhood exploration does not lead to a solution better than current one although it is often far from the optimal solution of the problem, they solve Local minimum problem by diversification they used hashing table This table is generally reset at the end of each iteration, If the neighborhood exploration does not lead to a solution better than the current one, they do not reset the table. If this happens for several iterations, more and more collisions will occur and more potentially good solutions will be excluded, forcing the algorithm to escape from the region currently explored. For these collisions to appear at a reasonable rate, the size of the secondary hashing table must be.

This approach for diversification is useful to avoid regions with a lot of local minima with the same cost, but is not sufficient to completely escape from one region and go to a possibly more attractive one. Therefore, each time the best solution found is not improved for 300 iterations; they randomly perturb the current solution in order to explore a new region from the search space. The perturbation consists of adding a randomly selected perturbation, uniformly chosen between -2 and +2, to 10 % of the weights small compared to the primary one. In their experiments, its size is 20 times the number of arcs in the network.

Another problem due to local minimum is cycling, Tabu search algorithms is used (Glover [7]), for example, make use of a tabu list that records some attributes of solutions encountered during the recent iterations and forbids any solution having the same attributes. They prove that by using local search for the same network and capacities, it could support a 50%–110% increase in the demands.

4. Genetic Algorithm

A genetic algorithm (GA) is a search technique used in computing to find exact or approximate solutions to optimization and search problems.

It derived from the principles of natural selection and evolutionary theory [8, 9].

To use genetic algorithm we have to define two things a genetic representation of the solution domain and a fitness function to evaluate the solution domain

Representation of the solution domain in OSPFWS problem is a value in the search space [1, 65535] all values in the search space represent feasible solutions. In our problem Representation of solutions is in form of weights of links arrays

And the fitness function determine the fitness weight values that achieve the object function

The first step to tailor genetic algorithm for OSPFWS problem is Representation

Second step initial population in this step we try to construct vectors of solutions for beginning the optimization process we can select random numbers from search space take as solution vector and also we can put all link weight values to 1 and take as another vector solution we can also take values [1/bandwidth] like CISCO recommendation this vectors used to begin optimization

Third step is evaluation function mean test the solution vectors and show how it fitness thought fitness function it is a necessary part in genetic algorithm that each solution apply in a simulated or real network and generate traffic after that measure the load of links this enable us to get the cost function which equal sum of all link utilization the cost function value is the fitness value

Fourth step is population partition. Which mean arrange the solution vector (individuals) according to their fitness values to three classes

Class A called elite individuals it content the best solutions and class B called middle class it content solution between class A and class C and class C called lower class and content the worst solution

Fifth step is parent selection this step to select which solution (parents) be selected to create the next improved solution (generation) selection can be done like that:

1- random select from class A

2- select other parent from class B and C
3- leave class A without updates but try to modifies class C by random generated solution

Sixth step crossover is used to vary the primary solution (parent) to next solution (child) in other meaning take some gens (weights values) from the first parent and others from the second parent to create the new chromosome (new vector for solution) for the children

Seventh step is mutation it means change some values of bad gens (weight) which reason of bad result.

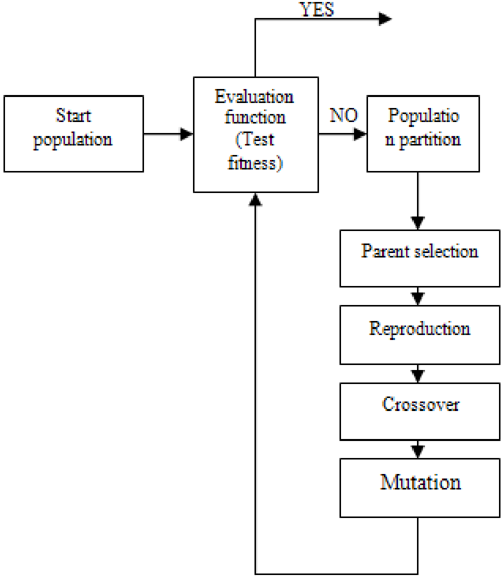

Figure 2. Steps of Genetic algorithm

M. Ericsson and RESENDE and PARDALOS use genetic algorithm like above and have good results they implement genetic algorithm for solve OSPFWS they find that it was able to increase network traffic by 70% until the network becomes congested [10].Figure 2; show the all steps of genetic algorithm.

5. Simulated Annealing

Simulated Annealing is a generic probabilistic meta-algorithm for the global optimization problem, namely locating a good approximation to the global optimum of a given function in a large search space. It originated as a generalization of a Monte Carlo method for examining the equations of state and frozen states of n-body systems .The first simulated annealing algorithm was proposed by Metropolis et al. in 1953 [12]. It was motivated by simulating the physical process of annealing solids. The process can be described as follows. Firstly, a solid is heated from a high temperature and then cooled slowly so that the system at any time is approximately in thermodynamic equilibrium. At equilibrium, there may be many configurations with each one corresponding to a specific energy level. The chance of accepting a change from the current configuration to a new configuration is related to the difference in energy between the two states [13]. S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi were the First to introduce simulated annealing to optimization problems in 1983 [14],Since then, simulated annealing has been widely used in combinatorial optimization problems and has achieved, The power of simulated annealing is that, besides accepting solutions with improved values, it also, to a limited extent, accept solutions with deteriorated values that prevent simulated annealing from the problem of local minima it has also great feature that regardless what values has been chosen at initiation configuration it produces very good solutions.

To use simulated annealing in OSPFWS problem Like genetic algorithm we begin to select weight vector values randomly to start optimization process. Simulated annealing start to update any weight in weight vector values to move to another solution at search space and after that evaluate the solution is done that like in genetic algorithm step three but it difference from genetic algorithm in it can accept defect solution than the current solution by some probability

Simulated annealing is depends on four parameter the initial temperature (T0), alpha (α), beta (β) and M. In simulated annealing each time the Metropolis loop is called, T is reduced to α M and M is increased to β M. Initial temperature values are set such that all the initial moves are accepted. This process is repeated for all the test cases with different scaled demand values.

M. Sqalli, S. Sait, and M. Mohiuddin found that α = 0.965, β = 1.01 and M = 10 are suitable parameters for OSPFWS problem. And simulated annealing is performing better than genetic algorithm [15].

6. Hybrid Genetic Algorithm

L.S. Buriol, M.G.C. Resende , C.C. Ribeiro And M. Thorup [16] developed Hybrid Genetic Algorithm for OSPFWS problem, It is like genetic Algorithm method but the difference is after running genetic Algorithm as described before, a local improvement procedure is applied to each offspring solution obtained by crossover from genetic Algorithm. The local improvement procedure analyzes solutions in the neighborhood of a current solution vector in the search for a solution having a smaller routing cost. If such a solution exists, then it replaces the current solution. Otherwise, the current solution is returned as a local minimum.

The local improvement procedure is incorporated in the genetic algorithm, described in Section 4, to enhance its ability to find better-quality solutions with less computational effort. Local improvement is applied to each solution generated by the crossover operator. Besides being computationally demanding, the use of large neighborhoods in a hybrid genetic algorithm can lead to loss of population diversity, and consequently premature convergence to low-quality local minima.

The local improvement procedure examines the effect of increasing the weights of a subset of the arcs. These candidate arcs are selected among those with the highest routing costs and whose weight is smaller than the maximum weight value. To reduce the routing cost of a candidate arc, the procedure attempts to increase its weight to induce a reduction on its load. If this leads to a reduction in the overall routing cost, the change is accepted and the procedure is restarted. The procedure stops at a local minimum when no improvement results from changing the weights of the candidate arcs.

Advantage of hybrid genetic algorithm is it finds better solutions in substantially less computation times and more robust, in the sense that it rarely gets stuck in suboptimal local minima, while the genetic algorithm often does so.

7. Discussion and Comparison

Local search**:** Local search is good optimization algorithm used in OSPFWS problem but it has a problem like local minima. Local minima is a problem which search algorithm stuck in a non optimal solutions figure 3, illustrate point N represent the local minima while there is an optimal solution than point N.

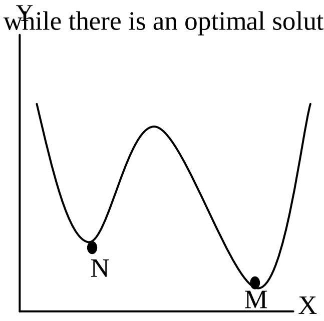

Figure 3. local minima problem

It is point M but this problem can be solved. In Local search we can perform diversification by randomly selecting a new neighborhood. It can be avoided getting trapped in local minima valleys. Another problem is cycling which re-computation of same routings under different weight settings. It can be solved by using of hash tables.

Genetic algorithm: Ericsson, Resende, and Pardalos compare solving OSPFWS problem by Genetic algorithm with the local search procedure of Fortz and Thorup (F&T), they state that the two heuristics find equally good solutions on the AT&T Worldnet backbone network instance, the 2-level hierarchical network instances, and the Waxman graph instances[10]. The running times of Genetic algorithm to solve OSPFWS problem varied from about 10 minutes to about 40 minutes on the IBM machine, while Fortz and Thorup mention CPU times only when they say that 5000 iterations of their algorithm took about one hour [11].

Simulated Annealing: The most merits in using simulated annealing in OSPFWS problem is it can accept worst solution than current solution this give this algorithm capability of avoiding local minima while local search algorithm trapped in it . Sqalli, Sait, and Mohiuddin [14] stat that simulated annealing is better than Genetic algorithm in many cases like maximum utilization and extra load

Hybrid Genetic Algorithm: Hybrid Genetic Algorithm found solutions at least as good as genetic algorithm on all networks and for all demand levels.It is not only found better-quality solutions, but did so in less CPU, It also finds better solutions in substantially less computation times. The experimental results also showed that it is also more robust, in the sense that it rarely gets stuck in suboptimal local minima, while the genetic algorithm often does so. When we compare Hybrid Genetic Algorithm with a local search heuristic, it is more robust than local search algorithms. Hybrid Genetic Algorithm and local search are competitive in terms of solution quality and time. Hybrid Genetic Algorithm is better than local search for some classes of test problems, while local search is better for others. Moreover, the implementation of local search is based on limited-size hashing tables that limit the number of iterations it can perform and, consequently, the solution quality that can be obtained for larger problems [15].

8. Proposed solution

Our opinion about OSPFWS problem that the quality of solution as it depend on the algorithm that used to solve the problem also it depends on the object function that used in test the quality of solution. We suppose new object function our new object function is to maximize non used bandwidth we can formulate as If C is capacity, l is load, and x is non used bandwidth then object function can be formulated like that $F=Max\sum Ca-Ia$, Where (a) from 1 to n and n=number of links. The old object function depends on the value of utilization of the link; it prefers to send data along less utilized links But in many cases it is wrong for example if we have two path the capacity of first path is 2 mb/s and the capacity of the second path is 128kb/s the utilization of the first path is 0.5% and utilization of the second path is 0% in that case the object function tends to send data to the second path regardless of the available bandwidth. in that case the first path have non used bandwidth equal to 2-0.5*2=1mb/s and the second path have nonused bandwidth equal to 128 kb/s then the router will send data on 128 kb/s and leave 1 mb/s and that is wrong but in our object function we prefer max c-l to send data according to our object function will prefer to send data to the first path. This method is named Empty Bandwidth (EM) method .

	Local search	Genetic	Hybrid Genetic	Simulated annealing
Results	50%–110% increase in the demands	where compared to the InvCapOSPF heuristic, increase network traffic by 70% until the network	It is better than Genetic and local search	It depends on the size of the network
Local minima	The main problem	often gets stuck in suboptimal local minima	rarely gets stuck in suboptimal local minima	avoiding local minima
performance	competitive in terms of solution quality	It is better than local search	it is more robust	better in maximum utilization and extra load
Running time	5000 iterations took about one hour	the algorithm will run for approximately 672 seconds	less computation times	Not mentioned

Table 1, Comparison between algorithms

Evaluation

Evaluation is the bottle neck of any work, Evaluation of the proposal has to be made to measure the quality of EM method, we start to evaluate our proposal and compare the results to other methods (Cisco method, unity method, random method) (tool command language) TCL language is used for implement the new algorithm.

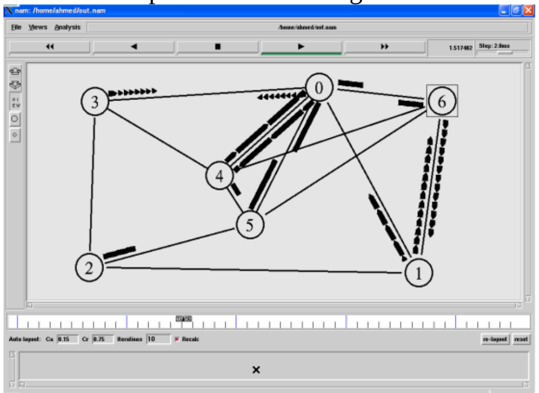

Figure 7, 14 nodes with 14 links network topology

We take a network in figure.4 as an example for simulate our idea

Discussion

EM is compared with three legacy method for setting OSPF weights (unit all weights equal to 1 , random all weights take random value, Cisco weights is setting inversely with bandwidth) EM perform better than all three network in both cost function and link utilization comparison .

9. Conclusion

This paper presents and compares different algorithms used to solve OSPFWS problem such as ***local search***, ***genetic algorithm***, ***simulated annealing***, and ***hybrid genetic algorithm***. It discusses each algorithm and depicts the advantage and disadvantage of each one. Also the comparison is made between algorithms it illustrate that each algorithm is better than other according to the case that used in, finally we described our new proposal for solving OSPWS problem.

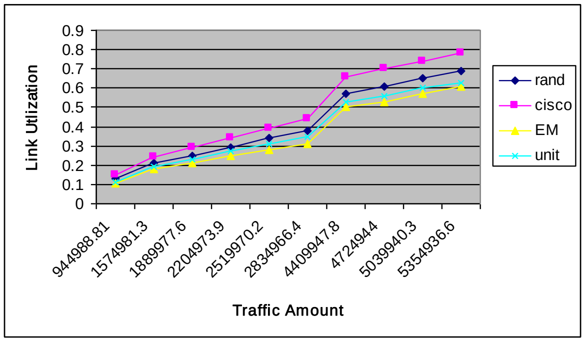

Figure 5, 7 nodes with 14 links traffic with utilization

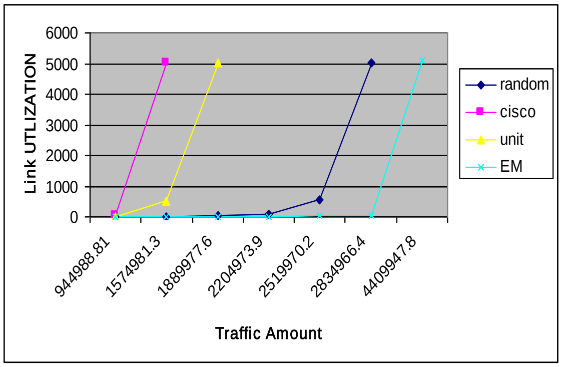

Figure 6, 7node with 14 links traffic with cost function

Traffic amount	944988.8	1574981	1889978	2204974	2519970	2834966	4409948	4724944
Random	0.138	0.23	0.27	0.32	0.368	0.41	0.62	0.66
Cisco	0.16	0.26	0.32	0.375	0.429	0.477	0.71	0.75
EM	0.116	0.193	0.232	0.27	0.309	0.34	0.539	0.57
unit	0.126	0.21	0.25	0.295	0.337	0.379	0.57	0.608

Table 2, 7 node 14 link traffic amount with link utilization

Traffic amount	944988.8	1574981	1889978	2204974	2519970	2834966	4409948	4724944
Random cost	15	21	30	92	538	5040	10601	15538
Cisco cost	24	5025	5087	5517	10017	10017	10162	10652
EM cost	13	15	17	19	25	31	609	5537
Unit cost	15	518	5020	5020	5027	5034	10588	11018

Table 3, 7 node 14 link traffic amount with link utilization

Reference

[1] K.G. Coffman and A.M. Odlyzko, “Internet growth: Is there a "Moore's Law" for data traffc?”, In J. Abello, P.M. Pardalos, and M.G.C. Resende, editors, Handbook of Massive Data Sets, Kluwer Academic Publishers, 2001

[2] Internet Engineering Task Force, Ospf version 2, Technical ,Report RFC 1583, Network Working Group, 1994.

[3] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, “Deriving traffic demands for operational ip networks: Methodology and experience”, IEEE/ACM Transactions on Networking, pages 265--279, June 2001.

[4] Bernard Fortz and Mikkel Thorup,” Internet traffic engineering by optimizing “, OSPF weights. IEEE Conference on Computer Communications,(INFOCOM), pages 519–528, 2000.

[5] URL http://en.wikipedia.org/wiki/Local_search_%28optimization%29

[6] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in Proceedings of IEEE INFOCOM, Tel-Aviv, Israel, March 2000, pp. 519--528

[7] F. Glover, “Future paths for integer programming and links to artificial intelligence,” Computers & Operations Research, vol. 13, pp. 533–549, 1986

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

[9] J.H. Holland, “Adaptation in Natural and Artificial Systems”, MIT Press, Cambridge MA, 1975.

[10] M.Ericsson and RESENDE and PARDALOS " A Genetic AlgorithmL FOR the Weight Setting Problem IN OSPF Routhing " October 9, 2001

[11] B. Fortz and M. Thorup, ”Increasing internet capacity using local search.” Technical report, AT&T Labs Research, 2000.

[12] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, “Equation of State Calculations by Fast Computing Machines”, J.Chem. Phys., vol. 21, no. 6, pp. 1087-1092, 1953.

[13] Xiao-Ying Wang, Jonathan M. Garibaldi, “Simulated Annealing Fuzzy Clustering in Cancer Diagnosis”, Informatica 29 (2005). 61–70

[14] S. Kirkpatrick, C. D. Gelatt Jr. and M. P. Vecchi, “Optimisation by Simulated Annealing”, IBM Research Report, RC 9355, 1982.

[15] Mohammed H. Sqalli, Sadiq M. Sait, and Mohammed Aijaz Mohiuddin “An Enhanced Estimator to Multi-objective OSPF Weight Setting Problem”, 2006

[16] L.S. Buriol, M.G.C. Resende , C.C. Ribeiro And M. Thorup A Hybrid Genetic Algorithm for the Weight Setting Problem in OSPF/IS-IS Routing March 2005 (AT&T Labs Research Technical Report TD-5NTN5G)