Genetic Algorithm for Optimizing Network Load Balance in MPLS Network

Conference Paper · November 2012

DOI: 10.1109/CICN.2012.119

2 authors:

Ashish Jain
Manipal University Jaipur
20 PUBLICATIONS **149** CITATIONS

SEE PROFILE

Narendra Chaudhari
Visvesvaraya National Institute of Technology
231 PUBLICATIONS **1,404** CITATIONS

SEE PROFILE

2012 Fourth International Conference on Computational Intelligence and Communication Networks

Genetic Algorithm for Optimizing Network Load Balance in MPLS Network

Ashish Jain
Department of Computer Science & Engineering
IIT Indore, India
ashishjn.mecs@gmail.com

Narendra S. Chaudhari
Department of Computer Science & Engineering
IIT Indore, India
nsc183@gmail.com

***Abstract* — This paper presents a flexible genetic algorithm (FGA) for optimizing network load balance in MPLS network. Along with FGA, minimum cost path constraint is also considered. Multiconstraints optimal Network load balancing is an NP-hard problem and it is an important part of traffic engineering. Thus, in this research we investigate how a genetic algorithm can be employed to solve the network load balancing problem. Minimum cost routing with load balancing method based on proposed genetic algorithm can be applied in internet for dynamic routing.**

***Index Terms*— Genetic Algorithm, Network load balancing, MPLS Networks.**

I. INTRODUCTION

Network Load-balancing is an important part of Traffic Engineering. It allocates load that needs load balancing among the M paths which have been established between the router in the entrance and exit according to certain algorithms. Traffic Engineering can enhance flow admitting rate of great bandwidth service load and make the network resource balanced. Thereby increasing the network throughput [1].

With the phenomenal growth of internet traffic and the increasing need to support engineering applications with strict Quality-of-Service requirement, several new architectures have been proposed. Recently, Multiprotocol Label Switching (**MPLS**) technology has been introduced by the Internet Engineering Task Force (**IETF**) to provide the Internet Services Provider (**ISPs**) with a more flexible and improved support for Traffic Engineering (**TE**) and Quality-of-Service (**QoS**) over the internet [2, 3].

IETF (Internet Engineering Task Force) defines traffic engineering as: "The aspect of network engineering which is concerned with the performance optimization of operational networks"; furthermore, the description adds that "traffic engineering encompasses the application of technology and scientific principles to the measurement, modeling, characterization, and control of internet load, and the application of such knowledge and techniques to achieve specific performance objective, including the reliable and expeditious movement of load through the network, the efficient utilization of network resources, and the planning of network capacity" [4].

Recently the research of traffic engineering algorithm has attracted much attention and MPLS (Multi Protocol Label Switching) provides an ideal platform for implementation of various algorithms. The MPLS network employs an explicit route which provides one or more explicit **LSPs** (Label Switching Paths) between the routers in the entrance and exit either by static configuration or dynamic routing method, avoiding the bottleneck link and emulation imposed on bottleneck resource, optimizing the mapping from load to resource and enhancing the network performance [5].Thus, Load balancing algorithms are designed specially to equally spread the load on link in MPLS network. In order to achieve these goals, the load-balancing method should be "fair" in distributing the load across the link. This implies that the difference between the heaviest-loaded and the lightest-loaded links should be minimize, therefore the load information on each MPLS router must be updated constantly so that the load-balancing mechanism can be more effective.

In section II we present an optimization mathematical model of load distribution based on integer programming. Experiments which we have done are represented in section III and in section IV FGA method is proposed. Section V presents the result of several experiments conducted using proposed FGA method followed by conclusion in section VI.

II. PROBLEM DESCRIPTION

Let digraph G = (V, E, C) represent the network topology, where V is the set of nodes, E is the set of links and C is the set of capacity and constraints associated with the nodes & links.

Let K be the set of load demands or LSPs where for each $k \in K$, (s_k, t_k, λ_k) denotes load demand: s_k, t_k, λ_k is the source node, destination node & bandwidth demand respectively. Let LSP k is routed on link (i, j) denoted by X_{ij}^k and h_k represents hop restriction of LSP k; where $(i, j) \in E$.

The optimization objective is to minimize the maximum of link utilization which ensures that the load is moved away from congested hot spots to less utilized parts of the network and the distribution of load is balanced across the network. Minimizing the maximum of link utilization also leaves more space for future load growth. When the maximum of link utilization is minimized, the percentage of the residual bandwidth on links is also maximized. Therefore, the growth in load in the future is more likely to be accommodated and can be accepted without requiring the re-arrangement of connections.

A. Mathematical Description

Let C_{ij} represent the capacity of the link (i, j) and α represent the maximum of link utilization among all the links therefore the mathematical description of network load distribution optimization problem can be given as follows [6]:

978-0-7695-4850-0/12 $26.00 © 2012 IEEE
DOI 10.1109/CICN.2012.119

- Optimization objective is to Minimize α

Where α represents "maximum of link utilization"

- Subject to following constraints:

1. For each demand the load flowing into a node should be equal to the load flowing out of the node for any node other than the source node and the destination node.

$$\sum_{j:(i,j)\in E} X_{ij}^{k} - \sum_{j:(j,i)\in E} X_{ji}^{k} = 0, k \in K, i \neq s_k, t_k$$

2. The net flow out of the source node should be 1.

$$\sum_{j:(i,j)\in E} X_{ij}^{k} - \sum_{j:(j,i)\in E} X_{ji}^{k} = 1, k \in K, i = s_k$$

3. The net flow entering on the destination node should be -1.

$$\sum_{j:(i,j)\in E} X_{ij}^{k} - \sum_{j:(j,i)\in E} X_{ji}^{k} = -1, k \in K, i = t_k$$

4. Link capacity utilization constraint says that the total amount of bandwidth consumed by all the demands routed on a link should not exceed the maximum utilization rate times the total capacity of the link.

$$\sum_{k\in K} \lambda_k . X_{ij}^{k} \leq C_{ij} . \alpha, (i,j) \in E$$

5. The number of hops in the path of a LSP must not exceed the hop restriction.

$$\sum_{(i,j)\in E} X_{ij}^{k} \leq h_k, k \in K$$

6. All decision variables are either 0 or 1 and only one route is retrieved, here α is non-negative.

$$X_{ij}^{k} \in \{0,1\}, \alpha \geq 0$$

B. Feasible Route Set, Link Load and Utilization Rate:

The feasible-route-set (FRS) for the k^{th} LSP corresponding to the demand (s_k, t_k, λ_k) that met the constraints given above can be denoted as:

$$Q_k = \{ q_k^1, ..., q_k^j, ..., q_k^{N_k} \}$$

Where $k \in K$ and N_k is the number of feasible routes of the k^{th} LSP. The optimization problem above is equivalent to the solution to the optimization route set, where:

$$P = (p_1, ... p_k, ..., p_{|K|})$$

$p_k \in Q_k$ which met the optimization objective (i.e. min α).

* FRS obtains using Dijkstra kth shortest path algorithm.

Now, we define utilization factor:

$$\delta_{kl}^{pk} = \begin{cases} 1, if\ LSP\ k\ is\ routed\ on\ link\ l \in E \\ 0, otherwise \end{cases}$$

Where l represent the link (i, j)

Therefore link load $\gamma_p(l)$ and utilization rate $\alpha_p(l)$ can be evaluate using following formula:

$$\gamma_p(l) = \sum_{k=1}^{|K|} \delta_{kl}^{p_k} . \lambda_k$$

$$\alpha_p(l) = (1 / c_l) \times \gamma_p(l)$$

The essential step of the optimization problem above is to choose the exact variable (p_k) form each multi-choice domain (Q_k), namely multi-choice assignment problem; it is NP-hard. So far there is no optimization solution through polynomial algorithm. However genetic algorithms have gained immense popularity over the last few years as a robust and easily adaptable search and optimization technique [7]. In this paper we investigate standard genetic algorithm based on that we proposed FGA method for algorithmic efficiency and optimization of solution.

III. Experiment Done

To solve the network load balancing problem we have done some experiments in [6] using brute force approach and dynamic programming techniques (classical methods) which is fail to balance network load however formulation of genetic algorithm for balancing network loads was proposed in [6], this work is extension of our previous work. In this paper we proposed FGA that fits our requirements which is possible by using fitness function and other optimization operators provided by standard genetic algorithm. Chromosome representation, evaluation and selection algorithm and how other operators to be used are given in section IV. JAVA is chosen as the programming language for FGA implementation and producing results which is presented in section V.

IV. Proposed Solution: FGA

A. Chromosome Generation

Firstly, we assign a natural number beginning with 1 to each route of the feasible route set of every LSP therefore the serial number permutation selected randomly from each feasible route set will be a possible solution (a chromosome) for the original problem. For instance, select the serial number of route of the 1st pair of entrance/exit nodes(s_1, t_1, λ_1) as y_1, select the serial number of route of the 2nd pair of entrance/exit nodes (s_2, t_2, λ_2) as y_2,…, select the serial number of route of the k^{th} pair of entrance/exit nodes ($s_{|k|}$, $t_{|k|}$, $\lambda_{|k|}$) as $y_{|k|}$, hence (y_1 ,y_2 ,….,y_i ,….,$y_{|k|}$) constitute a chromosome, where y_i corresponds the route $q_i^{y_i}$ of route set Q_i of the i^{th} LSP.

As we use the natural number encoding method for chromosomes, the length of code is fixed, equal to the total number of LSPs but has no relation with the entire possible route number. When the number of possible route increases, the scope of each gene bit value changes, but the code length remain fixed, this approach overcomes the disadvantage of binary coding which has low encoding/decoding efficiency; low network scale sensitivity and search space is large. Meanwhile, the chromosome genes position has no ordinal response. The gene position is independent, it makes design of genetic operator flexible and guarantees a feasible solution converges to optimum solution thus improve the algorithm efficiency.

B. Evaluation

To generate initial population the algorithm uses uniform random selection strategy. Assume the largest route serial number of the feasible route set is Ni. Then in each individual (y_1, y_2,….,y_i,….,$y_{|k|}$) of initial population, y_i is

obtained by generating a random number between 1 and N_i. This method is simple and general but unable to guarantee the globality and sparsity of the population. Here, we use improved method based on the search space partition: First, generate certain area in solution space uniformly, and then constitute initial population by generating possible solution in each sub-area randomly. Specific procedure is as follows: First, generate random integer k between 1 to |K|, designate the k^{th} LSP' route, the routes of other LSP remain search variable and then uniformly split the solution space into N_k subspace. If the population size is popsize, then select at least $|popsize/N_k|$ sample in each subspace. This may expand diversity of the initial population of genetic algorithms; reduce the possibility of converging into local minimum solution.

In genetic algorithm, fitness is used to allocate reproductive traits to the individuals in the population and thus act as some measure of goodness to be maximized. The fitness function of each chromosome is evaluated by examining the soft constraints. Each soft constraint is assigned a penalty value which contributes to the fitness function for each constraint violated. Since the optimization objective is minimizing the problem, the fitness function can be given as:

$$\text{Fitness} = \left(\max_{l \in E} (\alpha_l)\right)^{-1}$$

Where α is calculated with the objective "minimize α". Chromosomes with the lower fitness have higher probability to reproduce and to survive in next generation.

Algorithm for Fitness Calculation for each chromosome

1. Initialize utilization matrix and fitness vector to zero.

2. Set utilization matrix[i][j] to 1 where match is found in feasible-route-set matrix for corresponding investigated chromosome (let m).

3. Repeat step 4 for each demand (let k Nos. demand exist in the network at a time quanta)

4. Check the utilization matrix; if links which comes in the kth gene vector are already set to 1 in the utilization matrix then update fitness[m]++ and set the corresponding utilization matrix entry according to kth gene vector.

Fig.1:Pseudo-code for finding fitness of chromosomes

In the above algorithm utilization, fitness and feasible-route-set are name of data structures which is used during algorithm implementation.

C. Genetic Operators

Reproduction (or selection) is an operator that makes more copies of better strings in a new population. Reproduction is usually the first operator applied on a population. During each successive generation, a proportion of existing population is selected to breed a new generation. Individual solutions are selected through fitness-based process, where fitter solutions are typically more likely to be selected [8]. Selection methods rate the fitness of each individual and preferentially select the best solution.

Algorithm for Selection

1. Search minimum fitness value in fitness [] and set threshold fitness.

2. Repeat step 3 to 5for each chromosome.

3. If minimum fitness is 0 then go to step 4 else go to step 5.

4. Current chromosome is optimal solution and break.

5. If fitness of current chromosome is below or equals to threshold then put the chromosome into new population.

6. Replace the initial population with the new population.

7. Generate two random numbers between 1 to size of new population and put the corresponding chromosomes into child population for reproduction.

Fig. 2: Pseudo-code for selecting chromosome

The i^{th} string in the population is selected with a probability proportional to F_i. Since the population size is usually kept fixed in a simple GA, thus the sum of the probability of each string being selected for the mating pools must be one. Therefore, the probability for selecting the string is:

$$ps_i = F_i \left(\sum_{i=1}^{popsize} F_i \right)^{-1}$$

Because this method is based on the probability selection, there exists statistical error. Therefore, we combine it with optimal preserving strategy, and guarantee the current individual with highest fitness can evolve to next generation and will not be destroyed by the randomness of genetic operation, thus achieving the convergence of the algorithm.

Crossover and Mutation:

When designing crossover operator and mutation operator, two principles as follow should be met:

(1) Do not destroy too many fine patterns that represent the good properties, in case to make algorithm convergent.

(2) Generate some new individual patterns effectively, maintain the population diversity, and avoid falling into the local optimal solution.

According to these two principles, if probability of crossover and mutation can adaptively change with the individual fitness, then the two targets above can be well achieved. We can calculate p_c and p_m as:

$$p_c = \begin{cases} k_1 (F_{\max} - F) / (F_{\max} - \bar{F}), F \geq \bar{F} \\ k_2, F \leq \bar{F} \end{cases}$$

$$p_m = \begin{cases} k_3 (F_{\max} - F) / (F_{\max} - \bar{F}), F \geq \bar{F} \\ k_4, F < \bar{F} \end{cases}$$

Where F_{max} represent the highest individual fitness of current population, is the average individual fitness value, F is some individual fitness, and $0 < k1, k2, k3\ k4 \leq 1$. In order to guarantee the multiplicity, individuals with low fitness have high probability of reconstruction and mutation, we make k1= k2=1 and k3=k4=0.5.

V. RESULTS

In this paper, we compare performance of the proposed method FGA and SPA (shortest-path algorithm: Conventional method) by measuring maximum link utilization. The network topology is shown in figure 3 carry on computer simulation to it; let maximum hop restriction of LSP is 5 hops. Table I show LSPs which need optimization with load requirements; let all links capacity is 125.

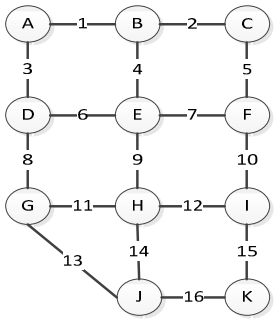

Fig. 3 Network Topology

Table I: LSP Entrance/Exit Nodes and Load Requirements

LSP Source-Sink Pair	AF	AK	BG	CJ	DK	HF
Bandwidth Demand	30	25	35	40	28	32

If we apply SPA algorithm to calculate route, we obtain the maximum ink utilization factor $\alpha = 0.612$ and there is one link which achieve this maximum link utilization (maximum load = 88), meanwhile there are few links of which the load is 0 which show the network load distribution is very unreasonable.

Now we employ proposed FGA method to optimize the network route. Firstly, we use Dijkstra kth shortest-path algorithm to obtain possible route set of LSP that meets delay constraints. Let population size = 100, maximum generation = 50, $Pc = 0.7$, $Pm = 0.06$. Using the maximum link utilization as the measurement of performance we have carried out more than 50 experiments, the data are statistical results shown in table II by averaging the outcome data.

Table II: Statistical Table of Simulation Result

Algorithm	Best Performance	Average Performance	Worst Performance	Conv ergent Speed
FGA	0.521	0.570	0.602	60
SPA	0.557	0.615	0.650	80

When the maximum link utilization $\alpha = 0.521$, the paths of LSP are: AF = (1, 2, 5), AK = (3, 6, 7, 10, 15), BG = (1, 3, 8), CJ = (2, 4, 9, 14), DK = (8, 13, 16), HF = (9, 7).

The simulation results indicate that compared with SPA the proposed FGA algorithm performance is more stable and it enhances the algorithm efficiency as well as optimized performance hence the purpose of balanced load flow is achieved.

VI. CONCLUSION AND FUTURE WORK

In this paper, we investigate the standard genetic approach however FGA method is proposed which is based on genetic algorithm and linear programming which solves traffic engineering problem in MPLS network. Optimal load balancing along with minimum routing cost objective is taken into account. The problem is solved under several constraints including bandwidth requirements, network resource constraints, and user constraints on number of hops (in this method maximum 5 hop is considered). The proposed approach is used to find the single optimal solution. The proposed method is flexible because other objective functions and constraints can also be incorporate as well. Due to DOP nature of the problem on MANETs , the recently found efficient scheme "GA with Immigrants and memory based schemes[9]" is to be investigated in the future for solving the same problem in MANETs by modifying proposed FGA method along with multihop constraint will also considered. One future work can also be considered by researcher to solve the same problem in under water wireless sensor networks (UWSNs) because its characteristics are different than terrestrial sensor networks.

ACKNOWLEDGEMENT

This work is supported by department of computer science and engineering, IIT Indore. The authors would like to thank to IIT Indore for supporting this work and also grateful to associate editor and reviewers for their thoughtful suggestions and constructive comments.

REFERENCES

[1] D. Awduche , Malcolm J, Jagogbua J, et al. Requirements for Traffic Engineering over MPLS[S], RFC 2702, 1999.

[2] Rosen, et al. "Multiprotocol Label Switching Architecture," RFC3031, IETF, January 2001.

[3] D. Awduche, A. Chintu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering," RFC 3272, IETF, May 2002.

[4] Ikctani A, Kuno Y. Real-time Surveillance System Detecting Persons in Complex Scenes in Proceedings of Image Analysis and Processing, 1999, 1112-1115.

[5] Lee Y, Seok Y, Choi Y. A constrained multipath Traffic Engineering scheme for MPLS networks [A]. ICC'02[C]. New York, 2002, 2431-2436.

[6] Jain Ashish, Chaudhari Narendra S., Ashapure Akash: Genetic algorithm based concept design to optimize network load balance, accepted in WOCN2012.

[7] Albert Y. Zomaya and Yee-Hwei Teh. Observations on Using Genetic Algorithms for Dynamic Load-Balancing, IEEE transaction on parallel and distributed systems, vol. 12, no. 9, September 2001, 899-911.

[8] David E. Goldberg, 1989, "Genetic Algorithms in search, optimization and machine learning".

[9] Yang Shengxiang, Cheng Hui and Wang Fang. Genetic Algorithms with Immigrants and Memory schemes for Dynamic Shortest Path Routing in Mobile Ad Hoc Networks, IEEE Transactions on System, Man and Cybernatics-Part C: Application and Reviews, vol. 40, No. 1, January 2010, 52-63.