

GENETIC ALGORITHM BASED CONCEPT DESIGN TO OPTIMIZE NETWORK LOAD BALANCE

Ashish Jain¹ and Narendra S. Chaudhari²

Department of Computer Science and Engineering, Indian Institute of Technology Indore, India
E-mail: ¹ashishjn.meecs@gmail.com and ²nsc183@gmail.com

Abstract

Multiconstraints optimal network load balancing is an NP-hard problem and it is an important part of traffic engineering. In this research we balance the network load using classical method (brute force approach and dynamic programming is used but result shows the limitation of this method) but at a certain level we recognized that the optimization of balanced network load with increased number of nodes and demands is intractable using the classical method because the solution set increases exponentially. In such case the optimization techniques like evolutionary techniques can employ for optimizing network load balance. In this paper we analyzed proposed classical algorithm and evolutionary based genetic approach is devise as well as proposed in this paper for optimizing the balance network load.

Keywords:

Genetic Algorithm, Evolutionary Techniques, Meta-heuristic, Network Load Balancing

1. INTRODUCTION

One most important thing in the internet is to provide the best path to data (load) to reach its destination which is done by routing algorithms executed between routers. If we talk about traditional internet routing algorithms, which use shortest path to forward load which is destination-based but this approach frequently causes uneven load distribution resulting congestion on some links due to over-loading while some other links are not utilized or under-utilized. The solution of this problem is balancing of load on the network.

Balancing Network Load is an important part of Traffic Engineering in which load allocates among the M paths of the network which have been established between the router in the entrance and exit using certain algorithms having objective is to minimize the maximum link utilization in this way traffic engineering enhances flow admitting rate of bandwidth service and make the network resource balanced which increases throughput of the network [1]. Therefore, there is a need of developing optimal load balancing algorithms which equally spread load on the link of MPLS network. In Fact Multi Protocol Label Switching (MPLS) provides an ideal platform for deployment of these algorithms because MPLS network employs an explicit route which provides one or more explicit Label Switching Paths (LSPs) in the entrance and exit of the routers either by static or dynamic routing method, avoiding the emulation imposed on bottleneck resource and bottleneck link thereby optimize mapping from load to resource which enhances the network performance [2].

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 more flexible and improved support for Traffic Engineering (TE) and Quality-of-Service (QoS) over the internet [3]-[4].

Problem and integer programming based optimization mathematical model for network load distribution is described in section 2. In section 3 we discussed experiments in detail and in section 4 formulation of genetic algorithm based concept for network load balancing is proposed. Section 5 presents the result of several experiments conducted using classical method followed by conclusion and future work in section 6.

2. PROBLEM DESCRIPTION

We can represent the network topology as the digraph $G = (V, E, C)$; 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 links and nodes.

Let K is the set of load demands or LSPs; Let $k \in K$ and (s_k, t_k, λ_k) denotes source node, destination node and 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. When the maximum of link utilization is minimized, the percentage of the residual bandwidth on links is 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 [5].

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

- Optimization objective is to Minimize α
- Subject to 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.

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

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.

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

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

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

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:

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

where, $p_k \in Q_k$ which met the optimization objective i.e. $\min \alpha$.

Now, we define:

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

where, l represent the link (i, j) .

However, Link load $\gamma_p(l)$ and utilization rate $\alpha_p(l)$ can be given as follow:

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

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

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 [6]. So far there is no optimization solution through polynomial algorithm. Thus in engineering we tend to balance the algorithm efficiency, optimize solution and get the ideal solution.

3. EXPERIMENT DONE

To solve the network load balancing problem we performed some experiments. In our experiments we used the Brute force approach and dynamic programming techniques (classical method). JAVA is chosen as the programming language. This method produced satisfying results for limited number of nodes but as the number of nodes and demands were increased it became difficult to produce optimize results because for obtaining optimal solution we find all possible combination of feasible routes which increases exponentially.

Complexity of Classical Method: To obtain the optimal route set we find possible combination of routes from each LSP. Let the number of demand is K , number of nodes is n and the number of

feasible routes in each LSP k is Q_k , where; Q_k is a multi choice domain for each demand k . The number of feasible routes in some Q_k are greater than n while in some Q_k it is less than n and in some Q_k it may be equal to n thus on an average, number of feasible routes in any Q_k be n . So, possible combinations of routes to obtain optimal route set will be. In average case total number of demand may exceed to n (when each node have some data to transfer to any node then $k = n$). So average case complexity will be $O(n^n)$. In worst case each node have some data to transfer to every other node so $k = n!$ So worst case complexity will be $O(n^n!)$.

We recognized that for optimizing complexity of this problem we need a metaheuristic approach that fits our requirements which is possible by using fitness function and other operators provided by genetic algorithm. Formulation of chromosome, evaluation criteria and how other operators will be used is presented in the next section.

4. CONCEPT DESIGN USING GENETIC ALGORITHM

[7] The genetic algorithm works as follows:

1. Initialization of parent population
 2. Evaluation (fitness function)
 3. Selection
 4. Crossover/recombination
 5. Mutation
 6. Evaluate child and Go to step 3 until termination criteria satisfies.
- Initial population is generated randomly.

4.1 CHROMOSOME REPRESENTATION

First, we assign a natural number beginning with 1 to each route to the route set of every LSP therefore the serial number permutation selected for each feasible route set will be a possible solution (a chromosome) for the original problem. For example, 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, \dots , 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 $(y_1, y_2, \dots, y_i, \dots, y_{|k|})$ constitute a chromosome, where y_i corresponds to 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 change but the code length remain fixed this overcome 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 because genes position is independent which makes design of genetic operator flexible, arbitrary crossover operation, as well as mutation operation inside the scale value of gene bit leads no invalid chromosome appear which guarantees a feasible solution as well as genetic operations avoids search in the invalid space resulting improvement of algorithm efficiency.

4.2 EVALUATION

To generate initial population the algorithm uses uniform random selection strategy. Assume the largest route serial number of the feasible route set is N_i . Then in each individual $(y_1, y_2, \dots, y_i, \dots, y_{|k|})$ 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 minimum of $\lfloor popsize/N_k \rfloor$ sample in each subspace. This may expand diversity of the initial population of genetic algorithms; reduce the possibility of converging into local minimum solution. Fitness in biological sense is a quality value which is a measure of the reproductive efficiency of chromosomes. 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:

$$Fitness = \left(\max_{i \in E} (\alpha_i) \right)^{-1}$$

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.

4.3 GENETIC OPERATORS

Selection, Crossover and Mutation are genetic operators:

Selection (or Reproduction) is an operator that aid to select better strings in a new population because in each successive generation, a proportion of existing population is selected to breed a new generation. Individual solutions are selected using fitness function, where fitter solutions are typically more likely selected.

Common Methods of Selection are:

- Roulette wheel method
- Tournament selection
- Stochastic remainder selection

We are using Roulette wheel for our research work.

Roulette wheel Selection:

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 of 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.

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

- a. Do not destroy too many fine patterns that represent the good properties, in case to make algorithm convergent.
- b. 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 adaptively change with the individual fitness, then the two targets above can be well achieved. We can calculate p_c and p_m as:

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

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

where, F_{max} represent the highest individual fitness of current population, \bar{F} is the average individual fitness value, F is some individual fitness, and $0 < k_1, k_2, k_3, k_4 \leq 1$. In order to guarantee the multiplicity, individuals with low fitness have high probability of reconstruction and mutation, we consider $k_1 = k_2 = 1$ and $k_3 = k_4 = 0.5$.

5. RESULTS

In this research, we analyzed the results of classical method on the network topology as shown in Fig.1 by carry on computer simulation to it; Let the maximum hop restriction of LSP is only 1 hop. Table.1 show LSPs which need optimization with load requirements; let all links capacity is 125.

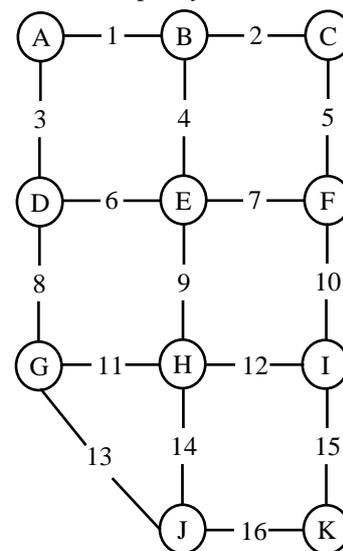


Fig.1. Network Topology

Table.1. LSP Entrance/Exit Nodes and Load Requirements

LSP Source-Sink Pair	AC	AE	BG	CI	DK	EI
Bandwidth Demand	30	25	35	40	28	32

When we apply our designed classical algorithm to calculate route, we obtain results of AC, AE, CI and EI LSPs in polynomial time with proper link utilization but we didn't get results of BG and DK demands because this LSPs require more than one hop however we improved algorithm to work for many hops results in running time of algorithm increases exponentially. One possible solution for this problem is to use Dijkstra kth shortest-path algorithm to obtain possible route set of LSP that meets delay constraints but this approach will results in unreasonable link utilization.

Table.2. Statistical Table of Simulation Result

Algorithm	Best Performance	Average Performance	Worst Performance	Convergent Speed
Classical Method	0.900	0.770	0.710	70

When the maximum link utilization $\alpha = 0.900$, the paths of LSP are: AC = (1, 2), AE = (3, 6), BG = (4, 9, 11), CI = (5, 10), DK = (8, 13, 16), EI = (9, 12). The simulation results indicate that results is reasonable but it degrades algorithmic performance very fast which is the main drawback of this algorithm as well as for large network the classical algorithm intractable to work due to many hops. In the above case the time taken by algorithm is approx 1 hour on i3, 2.20 GHz processor which is about $O(n^n)$.

6. CONCLUSION AND FUTURE WORK

In order to optimize network load balance, we first solved this problem with the help of classical approach. This algorithm first evaluates the feasible route sets for each LSP, and then select optimal route within each LSP for balancing network load which is chosen to forward the load but when the number of nodes and demands in the network along with the constraints increases, the problem domain becomes intractable to solve using classical method because the solution set increases exponentially, in the worst case the time required is $O(n^n)$ it shows terrible performance of the classical algorithm. Therefore we conclude that when network and demands grow the classical approach fails to minimize the network load balance; however we formulated the solution with the help of genetic approach for minimizing the maximum link utilization so that network load balance can be optimize. Due to DOP nature of the problem on MANETs, the

recently found efficient scheme "GA with Immigrants and memory based schemes [8]" is to be investigated in the future for solving the same problem in MANETs by designing and modifying proposed genetic approach. 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.

REFERECES

- [1] Awduche D, Malcolm J, Agogbua J, O'Dell M and McManus J, "Requirements for Traffic Engineering over MPLS", *Requirements for Traffic Engineering Over MLPS*, 1999.
- [2] Lee Y, Seok Y, Choi Y and Changhoon K, "A constrained multipath Traffic Engineering scheme for MPLS networks", *IEEE International Conference on Communications*, Vol. 4, pp. 2431-2436, 2002.
- [3] Rosen E, Viswanathan A and Callon R, "Multiprotocol Label Switching Architecture", *RFC 3031*, 2001.
- [4] Awduche D, Chiu A, Elwalid A, I. Widjaja and X. Xiao, "Overview and Principles of Internet Traffic Engineering", *RFC3272*, 2002.
- [5] Yongfa Ling and Deyu Meng, "A Genetic Optimization Algorithm to Solve the Problem of the Load-Balancing of Network Load", *International Journal of Computer Science and Network Security*, Vol. 6, No. 7, pp. 63-68, 2006.
- [6] Hadj-Alouane A B and Bean J C, "A Generic Algorithm for the Multiple-Choice Integer Program", *Operations Research*, Vol. 45, No. 1, pp. 92-101, 1997.
- [7] David E. Goldberg, "Genetic Algorithms in search, optimization and machine learning", Pearson Education, 1989.
- [8] 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 Systems, Man and Cybernetics-Part C: Application and Reviews*, Vol. 40, No. 1, pp. 52-63, 2010.