A Genetic Optimization Algorithm to Solve the Problem of the Load-Balancing of Network Load

Yongfa Ling,† *and Deyu Meng*††*,*

Faculty of Science, Xi'an Jiaotong University, Xi'an 710049, China

Summary
An optimum genetic algorithm for balanced network load is proposed in this paper; the algorithm employs the natural number-coded and the selection crossover mutation operator. The simulation result shows that the algorithm is effective, and remarkably improves the uneven load distribution that exists in the conventional short-path algorithm (SPF).

Key words:
Traffic engineering, balanced load, genetic algorithm.

1. Introduction

Reference [1] pointes out that, the traditional Internet route protocol uses the shortest paths to forward load which is destination-based, which frequently causes uneven load distribution, resulting in congestion on some links because of over-loaded, while some other links are under-utilized. Load-balancing is an important part of traffic engineering, and 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 a certain algorithms (for instance, minimize the maximum link utilization). It can enhance flow admitting rate of great bandwidth service load, and make the network resource be balanced, therefore increase the network throughput.

IETF defines traffic engineering as "that 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, modelling, 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"[2].

Recently, the research of traffic engineering algorithm has attracted much attention, and MPLS (Multi-Protocol Label Switching) provides an ideal platform for implementation of these various algorithms. The MPLS network employs explicit routes, which provides one or more explicit paths LSP (Label Switching Path) between the routers in the entrance and exit either by static configuration or dynamic routing method, avoiding the bottleneck link, avoiding emulation imposed on bottleneck resource, optimizing the mapping from load to resource, and enhancing the network performance.

In Document [3], the design of traffic engineering with MPLS in an Internet Service Provider's (ISP) network is presented; In Document [4], the mathematic description of four load engineering problems in MPLS based IP networks is presented, but the specific solution algorithm has not been given, it pointed out that all these cases is NP-hard; In Document [5], the explicit restricted LSP heuristic method conforming to Load-balancing principle is proposed; In Document [6], the mathematics description and analysis of Load-balancing problem with multi-constrained condition is presented. The algorithms in Documents [5,6] require solving mathematical programming problem using mathematical tools, and it is often difficult to combine with the practical application environment, and is restricted by the ability of mathematical tools.

We firstly present an optimization mathematic model of load distribution based on integer programming, multi-constraint, the optimization problem is NP-hard, Later, an optimization method based on genetic algorithm is proposed, used for solving problems of balanced network flows with high complexity computation, simulation experiments and analysis of the algorithm is given.

2. Problem Formulation

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 other constraints associated with the nodes and links. Let K be the set of load demands of LSPs. For each $k \in K$, we denote it as (s_k, t_k, λ_k), let s_k, t_k, λ_k be the source node, destination node and bandwidth demand respectively.

Manuscript received July 5, 2006.
Manuscript revised July 25 , 2006.

X_{ij}^{k} represent if LSP k is routed on link $(i, j), (i, j) \in E$, h_k represent hop restriction of LSP k .

The optimization objective is to minimize the maximum of link utilization. This optimization objective 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. Let C_{ij} represent the capacity of link (i, j) , let α represent the maximum of link utilization among all the links. The mathematic description of network load distribution optimization problem is given as follows.

Optimization objective:

$$\min \alpha \qquad (1)$$

Constraints:

$$\sum_{j:(i,j)\in E} X_{ij}^{k} - \sum_{j:(j,i)\in E} X_{ji}^{k} = 0, k \in K, i \neq s_k, t_k \qquad (2)$$

$$\sum_{j:(i,j)\in E} X_{ij}^{k} - \sum_{j:(j,i)\in E} X_{ji}^{k} = 1, \quad k \in K, i = s_k \qquad (3)$$

$$\sum_{j:(i,j)\in E} X_{ij}^{k} - \sum_{j:(j,i)\in E} X_{ji}^{k} = -1, \quad k \in K, i = t_k \qquad (4)$$

$$\sum_{k\in K} \lambda_k \cdot X_{ij}^{k} \leq C_{ij} \cdot \alpha, \quad (i, j) \in E \qquad (5)$$

$$\sum_{(i,j)\in E} X_{ij}^{k} \leq h_k, \quad k \in K \qquad (6)$$

$$X_{ij}^{k} \in \{0,1\}, \quad \alpha \geq 0 \qquad (7)$$

The objective function (1) says the variable to be minimized is the maximum of link utilization. Constraint (2) says that the load flowing into a node has to equal the load flowing out of the node for any node other than the source node and the destination node for each demand. Constraint (3) (4) is the net flow restriction of input node and output node respectively[7]. Constraint (5) is the link capacity utilization constraint. Constraint (6) restricts the number of hops in the path of a LSP. Constraint (7) specifies that all decision variables are either 0 or 1, and only one route is retrieved, here, α is non-negative.

We denote the route set corresponding to the terminal nodes (s_k, t_k, λ_k) that are met the constraints above as the feasible route set of the kth LSP, $Q_k = \{q_k^1, \cdots, q_k^j, \cdots, q_k^{N_k}\}, k \in K$, where N_k is the number of feasible routes of the kth LSP. Therefore the optimization problem above is equivalent to the solution to the optimization route set $P = (p_1, \cdots, p_k, \cdots, p_{|K|})$, $p_k \in Q_k$ which are met the equation (1). Let l represent link (i, j) , define

$$\delta_{kl}^{p_k} = \begin{cases} 1, & \text{if LSP } i \in F \text{ is routed on link } l \in E \\ 0 & \text{otherwise} \end{cases} \qquad (8)$$

then, link load $\gamma_p(l)$ and utilization rate $\alpha_p(l)$ are given as follow.

$$\gamma_p(l) = \sum_{k=1}^{|K|} \delta_{kl}^{p_k} \lambda_k \qquad (9)$$

$$\alpha_p(l) = (1/C_l) \times \gamma_p(l) \qquad (10)$$

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[8]. So far, there is no optimization solution through polynomial algorithm, in engineering, we tend to balance the algorithm efficiency and optimization solution, and get the ideal solution. The characteristic of the optimization problem above.

3. Solution to the Algorithm

Genetic algorithms (GA) were proposed by Professor Holland, Michigan University in 1969, summarized by De Jong in 1975, Goldberg in 1989, it is a category of Simulated Evolutionary Algorithm (SEA), which mimic the natural evolution process and mechanism of the biology, it is a category of self-organized, adaptive artificial intelligence technology of optimization solution and search problem [9,10,11]. When applying genetic algorithms, the essential factor below should be determined: (1) coding method for chromosome. How the chromosome coding can be applied in manifesting solution to the problem?. (2) The Fitness Function. The probability to choose a certain individual is proportional to its fitness. (3) How to generate the initial population? (4) Determine the genetic operator, how to carry on genetic operators, such as selection, crossover, mutation and produce the next generation population.(5) determines the running parameter, including population size (popSize), maximum generations, crossover probability p_c and mutation probability p_m .

3.1 Chromosome Coding

Firstly, we assign a natural number beginning with 1 to each route of the route set of every LSP, the serial number permutation selected from each feasible route set is namely 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 kth pair of entrance/exit nodes $(s_{|K|}, t_{|K|}, \lambda_{|K|})$ as $y_{|K|}$,hence $(y_1, y_2, \cdots, y_i, \cdots, y_{|K|})$ constitute a chromosome, where yi corresponds the route $q_i^{y_i}$ of route set Q_i of the LSP i .

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 increase, the scope of each gene bit value change, but the code length remain fixed, thus overcome the disadvantage of binary coding which has low encoding/decoding efficiency, network scale sensitive, 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, arbitrary crossover operation, as well as mutation operation inside the scale value of gene bit lead no invalid chromosome appear. It guarantees a feasible solution after carrying on genetic operation, avoids carrying on the search in the invalid space, hence, improve the algorithm efficiency.

3.2 Initial Population

The algorithm uses uniform random selection strategy. Assume the largest route serial number of the feasible route set Q_i is N_i , then, in each individual $(y_1, y_2, \cdots, y_i, \cdots, y_{|K|})$ of initial population, y_i is obtained by generating a random number between 1 and N_i .

This method is simple and general, but is unable to guarantee the globality and sparsity of the population. Here, we uses improved method based on the search space partition: First, generate certain area in solution space uniformly, then constitute initial population by generating possible solution in each sub-area randomly. Specific procedure as follows: First, generate random integer k between 1 to $|K|$, designated the kth 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 selects 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.

3.3 Fitness Measure

Since equation (1) is a minimum problem, the fitness function can be given as:

$$F(Y) = (\max_{l \in E}(\alpha_l))^{-1} \tag{11}$$

Where α_l can be calculated from equation (10).

3.4 Select Crossover Mutation Operator

The algorithm uses proportional selection method combined with optimal individual preserving strategy. In this way, the selection probability of each individual is proportion to the fitness value. Suppose the population size is popSize, the individual fitness as F_i , then the selection probability of individual i is:

$$ps_i = F_i \left(\sum_{i=1}^{popSize} F_i \right)^{-1} \tag{12}$$

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.

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, mutation can adaptively changing with individual fitness, then the two targets above can well achieved[12]. We can calculate p_c and p_m as equations (13) and (14).

$$p_c = \begin{cases} k_1 (F_{\max} - F)/(F_{\max} - \bar{F}), F \ge \bar{F} \\ k_2, F < \bar{F} \end{cases} \tag{13}$$

$$p_m = \begin{cases} k_3 (F_{\max} - F)/(F_{\max} - \bar{F}), F \ge \bar{F} \\ k_4, F < \bar{F} \end{cases} \tag{14}$$

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

$0 < k_1, k_2, k_3, k_4 \le 1$. In order to guarantee the multiplicity, individuals with low fitness have high probability of reconstruction and mutation, we make $k_1 = k_2 = 1, k_3 = k_4 = 0.5$. These adaptive crossover mutation operators can hold the whole population evolve to the direction with high fitness and improve convergence efficiency of the algorithm. On the other hand, when the individuals of the population tend to be identical with each other excessively, increases probability of crossover, mutation, hence, avoiding precocity.

Moreover a question is: How to determine position of crossover and mutation point? In order to improve the convergence speed and enhance local seeking best ability, we use some heuristic information as assistant. As the optimization objective is to balance network load flow, minimize maximum utilization of link, only the load of maximum utilization rate link changes, the performance of the solution can be improved. Supposes two parent string: $A = (p_1, p_2, \cdots, p_{|K|})$, $B = (p_1, p_2, \cdots, p_{|K|})$, calculate $F(A)$ and $F(B)$, select the bigger one, if $F(A) = F(B)$, select any one, suppose $F(A) > F(B)$, among all $a_A(l)$,find serial number of link l which has the maximum link utilization, and then obtain the serial number i of LSP which has least contributes to load of link l , takes i as the crossover point. Seemly, we can determine the mutation point. We select the serial number of LSP which has least contribution in order to have less affect on load of other links as far as possible.

4. Implementation of the Algorithm

In order to have a direct-viewing effect, we compare the new algorithm and the conventional algorithm. Denote the new algorithm as NGA, the traditional algorithm as SPF. Suppose the network topology is show as Figure 1. Carry on computer simulation to it. Use the maximum link utilization to measure the performance. Supposing maximum hop restriction of LSP 6 hops, the LSP nodes which need optimisation, and the corresponding load constraints are showed in Table 1, where all the links capacity is 155.

If we apply SPF algorithm to calculate route, we obtain the maximum link utilization factor α = 0.716, and there are two links both of which achieve this maximum link utilization (maxlink load=111), meanwhile there are a few links of which the load is 0, and the network load distribution is very unreasonable. As indicated below, we employ SGA and NGA to optimize the network route respectively.

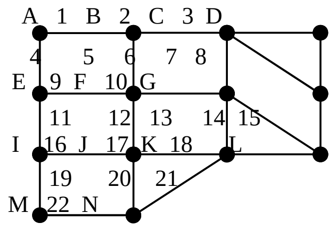

Fig. 1 Network topology

LSP entrance/exit nodes	AN	AL	AH	BM	CM
Load requirement	32	40	27	33	34
LSP entrance/exit nodes	DI	DM	DN	HI	MH
Load requirement	48	25	21	15	31

Table 1： LSP Entrance/exit Nodes Load Requirement

Firstly, we use Dijkstra the kth shortest path algorithm to obtain possible route set of LSP that meets delay constraints. Choice population size: popSize = 200, the maximum evolution generation: maxGen = 100, p_c =0.8, p_m =0.065 in SGA algorithm. Using the maximum link utilization as the measurement of performance, we have carried on more than 100 experiments, the data are statistical results by averaging the outcome data, see table 2. The simulation results indicate that, compared with SPF, the NGA algorithm performance is more stable, and it enhances the algorithm efficiency as well as the optimized performance.

Algorithm	Best performance	Worst performance	Average performance	Convergent speed
NGA	0.523	0.606	0.574	58
SPF	0.555	0.652	0.613	77

Table 2：Statistical Table of Simulation Result

Table 2 Statistical Table of Simulation Result When the maximum link utilization α =0.523,the paths of LSP are(denoted by serial number of routed link): AN=（1,5,12,20）, AL=(4,9,10,14), AH=(1,2,7,8), BM=(5,12,16,19), CM=(6,13,21,22), DI=(8,15,18,17,16), DM=(3,6,10,9,11,19) , DN=(3,6,13,17,20) , HI=(7,2,1,4,11), MH=(22,21,18,15)。

It can be seen that after route optimization of LSP by NGA algorithm, compared with the calculation results of SPF, the maximum link load is reduce by 15 percent, the network load distribution is more reasonable, and the purpose of balanced load flow is achieved.

Then, we use the widely applied Waxman model [13] to generate the simulation network. In figure of network model, the generation probability of (u,v) is given by

$$prob(u,v) = \alpha e^{-d(u,v)/\beta L}, 0 \le \alpha \le 1, \beta \ge 1 \qquad (15)$$

where, d is the Euclid distance between u,v, L is the largest distance of any node pair. For (s,t) node pair, $O_s, D_t, C_{(s,t)} \in [0,1]$ are three random values, and the load demand between s,t is given by

$$\lambda(s,t) = \gamma O_s D_t C_{(s,t)} e^{-d(s,t)/2L}, 0 \le O_s, D_t, C_{(s,t)} \le 1 \quad (16)$$

where g is a positive constant. Since $\lambda(s,t)$ is decided by multiplying three random numbers, the variability range is large.

Based upon the above assumptions, we have carried on massive simulation experiments of simulation network, in the network the number of generation nodes is 60, the number of link is 96 (namely, $|V| = 60, |E| = 96$), and the load demand is variable. The simulation result showed that, if the load flow slight loaded, SPF algorithm could also meet the network load demand. But when the load is over loaded, the performance of SPF degrades rapidly. As the load flow increases, the effect by applying NGA to the network load flow become more distinct, and the distribution of network load is well balanced, thus enhance the service ability of network greatly. Figure 2 shows the load distribution (link utilization) of the network when the total load demand is 3283; it shows the significance of applying NGA to network optimization for load balancing. Moreover, it is discovered that as scale of the network increases, the running time of the algorithm has not remarkable increased, and still can maintain good performance.

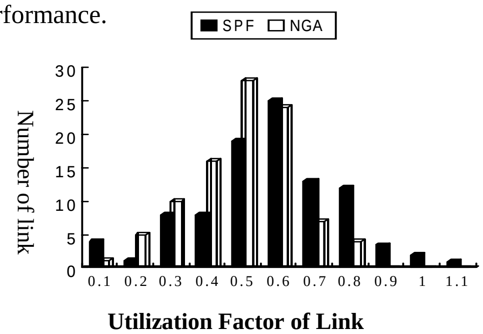

Fig. 2 Schematic Diagram of Network Load Distribution

5. Conclusion

A genetic optimization algorithm is proposed to solve the problem of the load-balancing of network load. The algorithm carries on the entire spatial parallel search which concentrates on the high performance parts in order to improve the efficiencies and robustness of the search with no restriction to the optimization objective function. The algorithm employs natural number coding to efficiently reduce coding space and searching space, meanwhile it does not lead to appearance of invalid chromosome, thus improve the searching efficiency of the algorithm. It chooses crossover mutation operator with strategies of preserving the optimization individual, thus improves its entire searching performances; it improves its efficiencies and optimization performances by merging automatic information into it which guides the searching direction and efficiencies. The simulation results have verified that the algorithm is efficient and evidently improve the status that traditional SPF route algorithm tends to unbalanced network load distribution.

References

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

[2] Ikctani A, Kuno Y. Real-time Surveillance System Detecting Persons in Complex Scenes In: Proc. Image Analysis and Processing. 1999: 1112-1115.

[3] Xiao X P. Traffic engineering with MPLS in the Internet[J]. IEEE Networking, 2000, 14(2): 28-33.

[4] Girish M K, Zhou B, Hu J Q. Formulation of the load engineering problems in MPLS based IP networks [A]. The Fifth IEEEISCC[C]. Antibes, France, 2000. 214-219.

[5] Wang Y F, Wang Z. Explicit routing algorithms for Internet Traffic engineering [A]. IEEE ICCCN'99[C]. Boston, MA, 1999. 582-588.

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

[7] Bertsekas D, Galager R. Data Networks [M]. NJ : Printice Hall, 1992.

[8] Alouane A B, Bean J C. A generic algorithm for the multiple-choice integer program [J]. Operations Research, 1997, 45 (1):92-101.

[9] De Jong K. The Analysis of the Behavior of A Class of Genetic Adaptive Systems [M]. Ph. D. thesis, University of Michigan, Ann Harbor. Dissertation Abstract International, 36(10), 5140B.

[10] Goldberg D E. Genetic Algorithms in Search, Optimization and machine Learning[M]. Reading, MA: Addison-Wesley,1989.

[11] Holland J H. Adaptation in Natural and Artificial Systems[M]. Ann Arbor, MI: University of Michigan Press. 1975.

[12] Srinivas M, Patnaink L M. Adaptive probabilities of crossover and mutation in genetic algorithms [J]. IEEE Transactions on Systems, MAN and Cybernetics,1994, 24 (4): 656-667.

[13] Waxman B M. Routing of multipoint connections [J]. IEEE Journal on Selected Areas in Communications, 1988, 6(9): 1617-1622.

Yongfa Ling, received the B.S. and M.S. degree from Yunnan Minzu University. Now he is the Postdoctoral students in Xi'an Jiaotong University. His scientific interests are in the fields of network algorithm and network control.

Deyu Meng, received the B.E. degree in Information Science in 2001 and M.E. degree in applied mathematics in 2004 from the Xi'an Jiaotong University. He is currently a doctoral student at School of Electronic and Information Engineering in Xi'an Jiaotong University. His scientific interests are in the fields of machine learning and data mining.