Proceedings of IAM, V.1, N.2, 2012, pp.121-128

GENETIC ALGORITHM FOR NETWORK PERFORMANCE OPTIMIZATION

Sergiy D. Pogorilyy[1], Roman V. Bilous[1]

[1]Taras Shevcenko National University, Kiev, Ukraine
e-mail: sdp@univ.net.ua

Abstract. A genetic algorithm for load-balancing problem in large heterogeneous computer networks is proposed. Encoding techniques and genetic operations features are substantiated. Simulation results show algorithm efficience in comparison with conventional algorithm.

Keywords: genetic algorithm, traffic engineering, routing, load-blanacing.

AMS Subject Classification: 60F05, 60G50, 60E10.

1. Introduction

Traffic engineering problems play a significant role in constructing and maintaining computer networks. Parallel routing and traffic-flow distribution algorithms have been designed to improve network convergence and performance [1, 2]. Load-balancing problems arise in many applications, but, most importantly, they play a special role in the operation of parallel and distributed computing systems and networks. Traffic engineering is concerned with optimizing the performance of operational networks. For dynamic networks there is a need to recognize load changes of individual channels and to form mechanisms of overall network routing process control [3]. Classical algorithms are not applicable to such problems, which belongs to the NP-hard [4] class. So far, there is no optimization solution through polynomial algorithm. A genetic algorithm is an alternative approach for solving the load-balancing problem of communication channels in the network. The main objective is to reduce congestion hot spots and improve resource utilization. This can be achieved by setting up explicit routes over the physical network in such a way that the traffic distribution is balanced across several traffic trunks.

2. Problem formulation

Communication network is represented as a weighted oriented graph $G=(V,E,C)$, where V is the set of vertices, $E \in V \times V$ is the set of edges of the graph and $C_{i,j} : E \to R$ is a throughput of each link (graph edge). Proposed algorithm employs explicit routes, which provides several explicit paths between two arbitrary nodes (routers) in the network. Let L be a set of network traffic distributions. For each $l \in L$ we denote it as (s_l, d_l, b_l), where s_l, d_l, b_l be the source node, destination node and traffic requirement respectively. Set L defines which network nodes will be taking into account by genetic algorithm. For each pair (s_l, d_l) we define a set

$$P_l = \{p_l^i\}, \quad i = 1 \ldots k_l \tag{1}$$

of alternative paths between nodes s_l and d_l provided by static configuration or dynamic routing method. Load balancing technique is accomplished by applying each alternative path p_l^i with a certain probability α_l^i so that:

$$\sum_i \alpha_l^i = 1. \tag{2}$$

During routing process data packets are routed according to coefficients α_l^i thus dividing traffic between alternative paths p_l^i proportionally.

The optimization objective is to minimize the maximum of link utilization [5]. 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. In order to calculate ordinary link $m \to n \in E$ utilization we define

$$X_l^i(m,n) = \begin{cases} 1, & if\ p_l^i \text{ is routed on } m \to n, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Utilization of a specific link $m \to n$ can be calculated as:

$$U(m,n) = \frac{1}{C_{m,n}} \sum_l \left(\sum_i X_l^i(m,n) \cdot \alpha_l^i \right) \cdot b_l, \tag{4}$$

where $i = 1 \ldots k_l$, $\quad l = 1 \ldots |L|$.

Let $U_{max} = max(U(m,n)),\ \forall (m,n) \in E$ represent the maximum of link utilization among all the links. The mathematic description of network load distribution optimization problem is given as follows:

$$min(U_{max}). \tag{5}$$

3. Genetic algorithm

A genetic algorithm (GA) is a search algorithm based on the principles of evolution and natural genetics. GAs combine the exploitation of past results with the exploration of new areas of the search space. By using survival of the fittest techniques combined with a structured yet randomized information exchange, a GA can mimic some of the innovative flair of a human search [6].

Using genetic algorithm for traffic engineering problems such as load-balancing provides a set of advantages:

• controlled convergence and response time;

Selecting different termination criterion and parameters can change convergence and response time of proposed algorithm. This may be well employed by dynamic routing control.

• adaptive;

As load requirements are changed or added into algorithm it changes chromosome length and fitness value calculation in order to find best problem solution with new requirement. Optimization parameters of algorithm itself can be changed during data processing as soon as new restrictions are met.

• easy to parallelize;

Genetic algorithms are hardly decentralized and provide a good platform for creating parallel algorithms which improves performance and gain speedup.

When designing crossover operation and mutation operation, two principles as follow should be met [7]:

• do not destroy too many fine patterns that represent the good properties, in case to make algorithm convergent;

• 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. Employing genetic algorithm in large computer network with various network links bandwidth can slightly decrease network overall performance because of specific objective. We need to ensure that load is balanced across the network but don't decrease its performance.

3.1. Chromosome representation

Ordinary chromosome Y consists of $|L|$ genes which represent a feasible solution to the problem formulated above. Each gene g_l is a set of coefficients

$$\alpha_l^i \; g_l = \{\alpha_l^i\}, \quad i = 1 \ldots k_l \tag{6}$$

Chromosome length remains fixed, thus overcome the disadvantage of binary coding which has low encoding/decoding efficiency [4], network scale sensitive, and search space is large. It guarantees a feasible solution after carrying on genetic operation, avoids carrying on the search in the invalid space, hence, improve the algorithm efficiency.

Such representation also simplifies applying genetic operations without additional chromosome decoding procedure.

3.2 Initial population

Each explicit path p_l^i has its own cost in terms of routing algorithm.

We define a single aggregated metric $Q(p_l^i)$ through a combination of weighting objectives for each path which determines chromosome fitness according to several parameters:

• path cost in terms of SPF algorithm,
• hop count,
• propagation delay,
• maximum bandwidth.

Calculated aggregated metrics defines initial values for coefficients α_l^i:

$$\alpha_l^i = \frac{Q(p_l^i)}{\sum_i Q(p_l^i)}. \tag{7}$$

These initial coefficients determines first chromosome of initial population. Complete chromosomes set is formed by applying mutation operation to first chromosome. Such approach employs dynamic or static routing information and ensures that open phase of genetic algorithm will not change network performance greatly.

3.3. Genetic operations: mutation

Mutation operation for the proposed algorithm is a random change in chromosome gene g_l (one-point mutation) or in set of genes simultaneously (multipoint mutation). Either ways mutation technique defines points of mutation and can be easily propagated through population. For arbitrary chromosome Y which is related to traffic requirement (s_l, d_l, b_l) we define mutation operation as follows:

• selecting mutation points,
• changing chromosome genes.

We use heuristic approach for selecting mutation points. Since the objectives of the algorithm is to minimize the largest consumer channels through the network, changing the load on this particular channel with the largest consumption can improve the solution of the problem in the next generation of the genetic algorithm.

Assuming that within the current generation largest consumption is observed on link $m \to n \in E$, only genes that define this link consumption value should be mutated in every chromosome of the population. We select genes that satisfy next conditions:

$$\begin{aligned} &\exists i \in (1 \ldots k_l), \quad so\ that \\ &\alpha_l^i \neq 0 \quad and \quad X_l^i(m,n) = 1. \end{aligned} \tag{8}$$

Mutation procedure of arbitrary gene in chromosome is changing coefficients α_l^i in order to achieve a better solution for the problem. We define this procedure as follows:

$$\alpha_l'^i = \alpha_l^i + \delta_l^i, \tag{9}$$

$$\sum_i \delta_l^i = 0, \tag{10}$$

where $\alpha_l'^i$ - next generation coefficient, $-1 < \delta_l^i < 1$.

Constraint (10) ensures that for next generation genes condition (2) will be met. Selecting δ_l^i for each generation is based on chromosome fitness value and normal distribution. For preserving fine patterns within population and generating new solutions affectively we introduce adaptive mutation probability [7].

Let F_{max} denotes best chromosome fitness function value within the population, $\overline{F}$ - average chromosome fitness function value, $0 < p_1, p_2, p_3, p_4 \le 1$. For arbitrary chromosome Y of current population with its fitness function $F(Y)$ mutation probability can be defined as follows:

$$p_m = \begin{cases} p_1 (F_{max} - F(Y)) / (F_{max} - \overline{F}), & if \; F(Y) \ge \overline{F} \\ p_2, & if \; F(Y) < \overline{F}. \end{cases} \tag{11}$$

3.4. Genetic operations: crossover

We use same heuristic approach to define crossover points. Because of the chosen solution representation method chromosome length remain constant, so the algorithm crossover operation is to share units or individual genes of parental chromosomes [4]. We define adaptive crossover probabilities for algorithm convergence:

$$p_c = \begin{cases} p_3 (F_{max} - F(Y)) / (F_{max} - \overline{F}), & if \; F(Y) \ge \overline{F} \\ p_4, & if \; F(Y) < \overline{F}. \end{cases} \tag{12}$$

To ensure the diversity of chromosomes within a population, the probability of crossover and mutation of the least adapted chromosomes should be high, so in proposed algorithm $p_1 = p_2 = 1$, $p_3 = p_4 = 0.5$. Such an adaptive approach for crossover and mutation operations provides the evolution of populations and improves the convergence of the algorithm. On the other hand, with increasing number of identical chromosomes in the population probability of mutation and crossover are increasing, thus the algorithm avoids premature convergence to local extremum. In proposed algorithm we use single-point, multipoint and pattern crossover techniques depending on heuristic search algorithm results.

3.5. Selection and convergence

According to algorithm objective which is to find a minimum (5) fitness function for arbitrary chromosome Y can be calculated as follows:

$$F(Y) = (max(U(l,m)))^{-1}, \quad \forall (l,m) \in E. \tag{13}$$

Proposed algorithm uses proportional selection method in conjunction with the method of preserving the best individuals of the population (elitism). Thus the probability of selecting a chromosome is proportional to fitness value of this chromosome. Let current population size is N and fitness function of ordinary chromosome Y_i, $i = 1 \dots N$ is equal $F(Y_i)$, then selection probability of chromosome Y_i is calculated as:

$$p_i^s = F(Y_i) \cdot \left(\sum_i F(Y_i) \right)^{-1}. \tag{14}$$

Since approach described above can produce statistical errors and loss of the best chromosomes we apply preserving best solutions technique in proposed algorithm. This method ensures that best individuals pass to the next generation of genetic algorithm and improves convergence.

4. Numerical results

For realization of the genetic algorithm we developed appropriate software realization and used a computer simulation for network performance measurement. The aim of these numerical experiments was to study network link consumption behaviour depending on system parameters (number of nodes, links and load distribution). We compared genetic algorithm described above with the conventional algorithm (SPF). It should be also noted that we used maximum link utilization U_{max} as the measurement of network performance. Simulation results showed that, if the network is slightly loaded and there is no need to balance redundant traffic, conventional algorithm could also meet the network demands (Fig. 1).

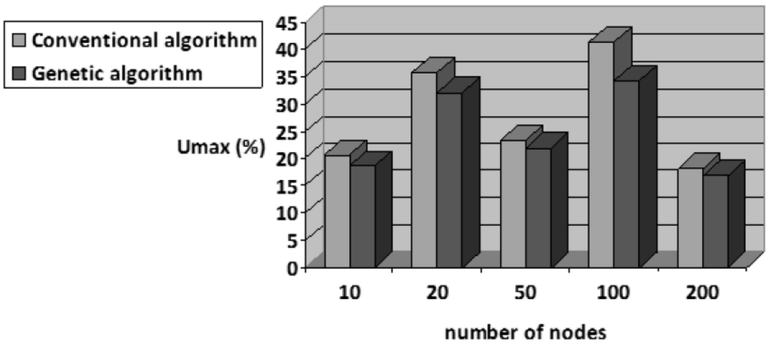

Figure 1: Algorithm performance for small load.

When the network load becomes excessive, the performance of SPF degrades rapidly. Increasing traffic flow in simulation network improves genetic algorithm efficiency thus the distribution of network load becomes balanced. This enhances the service ability of the network.

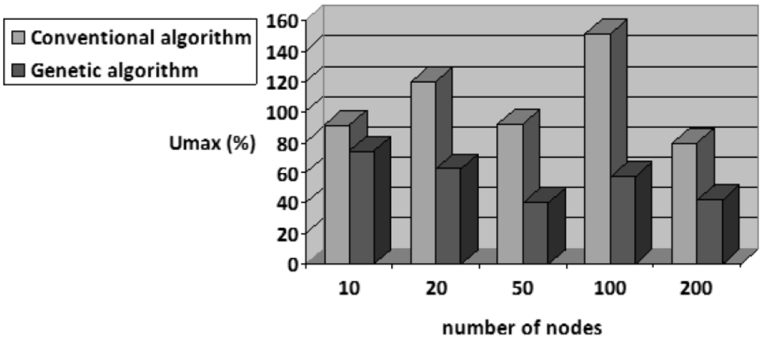

Figure 2: Algorithm performance for high load.

Figure 2 shows the load distribution (link utilization) of the network for conventional algorithm and with applying genetic load-balancing technique. It shows the significance of applying described approach to network optimization for load balancing in high loaded networks. Simulation results indicated that increasing number of network nodes and load demands has no significant influence on overall genetic algorithm performance.

5. Conclusions

We formalized the problem of balancing the load on communication channels in computer network and proposed genetic algorithm for solving it as an optimization problem. Using genetic approach in large dynamic transport networks was substantiated. For proposed genetic algorithm we denoted chromosome representation features to improve encoding/decoding efficiency.

Features of genetic operations: mutation and crossover were described. Proposed techniques include adaptive probabilities and heuristic mutation points search which increase genetic algorithm productivity and improves convergence. Selection includes proportional method with preserving best chromosomes which guarantees feasible solution in any algorithm generation. In order not to decrease network performance additional constraints together with correspondent chromosome destroying technique was applied.

Proposed approach is applicable for building and optimizing transport and computer networks. The simulation results have shown that the algorithm is efficient and can greatly improve network performance in terms of maximum link utilization and balance load distribution.

References

1. Pogorillyy S.D., Boyko Yu.V, Lozitskiy S.I., Gusarov A.D. Formalized methods of paralleling the Goldberg-Tarjan algorithm, Journal of Automation and Information Sciences, vol. 40, N.9, 2008, pp.64-75.

2. Pogorilyy S.D., Gusarov A.D. Paralleling of Edmonds-Karp net flow algorithm, Applied and Computational Mathematics, vol. 5, N.2, 2006, pp.121-130.
3. Pogorilyy S.D., Bilous R., Computer network performance optimization using genetic algorithms, 3rd International Conference on Control and Optimization with Industrial Applications, Bilkent University, Ankara, Turkey, 22-24 August, 2011, pp.19-20.
4. Girish M.K., Zhou B., Hu J.Q., Formulation of the load engineering problems in MPLS based IP networks [A], The 5th IEEEISCC[C], Antibes, France, 2000, pp.214-219.
5. Lee Y., Seok Y., Choi Y., A constrained multipath traffic engineering scheme for MPLS networks [A]. ICC'02[C], New York, 2002. pp.2431-2436.
6. Goldberg D.E. Genetic Algorithms in Search, Optimization and machine Learning[M], Reading, MA: Addison-Wesley, 1989.
7. Srinivas M., Patnaink L.M., Adaptive probabilities of crossover and mutation in genetic algorithms [J], IEEE Transactions on Systems, MAN and Cybernetics, 1994, vol. 24, N.4, pp.656-667.

Şəbəkə göstəricilərinin optimallaşdırılması üçün qenetik alqoritm

S.D. Poqorelıy, P.V. Bilous

XÜLASƏ

Böyük qetoroqen hesablama şəbəkələrində yüklənmənin balansı məsələsinin genetik həll alqoritmi təklif olunur. Kodlaşdırma metodları və əməliyyatın genetik xüsusiyyətləri əsaslandirilir. Simulyasiya nəticələri adi alqoritmlə müqayisədə təklif olunan alqoritmin effektivliyini göstərir.

Açar sözlər: genetik alqoritm, marşrutlaşdırma, yükləmə balansı, trafikin idarə olunması.

Генетический алгоритм для оптимизации показателей сетей

С.Д. Погорелый, Р.В. Билоус

РЕЗЮМЕ

Предлагается генетический алгоритм для задачи балансировки нагрузки в больших гетерогенных вычислительных сетей. Методы кодировки и генетические особенности операций являются обоснованными. Результаты симуляции показывают эффективность алгоритма по сравнению с обычными алгоритмами.

Ключевые слова: Генетический алгоритм, управление трафиком, маршрутизация, нагрузки балансировки.