Applications of Genetic Algorithms in BGP-Based Interdomain Traffic Engineering

Jiyun Yan[1(✉)], Zhenqiang Li[2], and Xiaohong Huang[1]

[1] Beijing University of Posts and Telecommunications, No. 10, Xitucheng Road, Haidian District, Beijing, China
yanjybupt@163.com, huangxh@bupt.edu.cn

[2] China Mobile Research Institute, No. 32, West Street, Xuanwumen, Xicheng District, Beijing, China
lizhenqiang@chinamobile.com

Abstract. With the rapid development of the Internet, widely deployed new services such as high definition videos and voice over IP (VoIP) require higher performance guarantees. Traffic engineering can improve end to end service quality, help large autonomous systems (AS) operators improve network resource utilization and meet the challenge. At the interdomain level, traffic engineering is more challenging. Most network operators still rely on changing routing policies and BGP attributes manually. In this paper, we discuss the existing systematic techniques in BGP-based interdomain traffic engineering and propose an improved algorithm based on a multi-objective genetic algorithm. Our algorithm is scalable and efficient. We apply our solution to a provincial network of China Mobile and discuss the influence of different parameters on the performance and validity of the algorithm.

Keywords: Genetic algorithms · Interdomain traffic engineering · BGP

1 Introduction

Today, the Internet has been developing rapidly. The traffic of the Internet is growing at a high speed while widely deployed new services such as high definition videos and voice over IP (VoIP) require higher performance guarantees. With the rapid growth of traffic and the emergence of various kinds of new services, the network operation is facing a big challenge. Nowadays more and more Internet Service Providers (ISP) rely on traffic engineering to control the flow of traffic into, out of, and across their networks. The main objective of traffic engineering is to optimize the performance of the network. Intradomain traffic engineering is mature and several techniques exist such as MPLS based traffic engineering and OSPF based traffic engineering [1]. Due to the incomplete information of the network topology and state and the complex interactions between the BGP decision process and IGP routing, the current state-of-the-art in interdomain traffic engineering is primitive. An efficient multi-objective evolutionary algorithm to evenly distribute the interdomain traffic has been proposed in [3–5]. As described in [5], the algorithm has one manifest drawback. If multiple eBGP peers in a same AS are attached to one egress router, it cannot guarantee a good balance among those peers. We design an improved genetic algorithm based on the multi-objective evolutionary heuristic. We solved this problem and reduced the running time of the algorithm by adding filtering mechanism for the selection of exit routers. Another key contribution in our paper is that we apply our method to a provincial network of China Mobile. We also discuss the influence of different parameters on the performance and validity of the algorithm which has great practical significance to network operators.

The remainder of this paper is organized as follows. Section 2 introduces the interdomain traffic engineering. Section 3 presents the related work in BGP-based interdomain traffic engineering. Section 4 introduces the improved multi-objective genetic algorithm. Section 5 presents the application of our solution to a provincial network of China Mobile. The last section summarizes the content of this paper.

2 Interdomain Traffic Engineering

The Internet is composed of Autonomous Systems (AS). A transit AS is mainly to transit packets from a neighbor domain to another. A stub AS does not provide transit service and only sends or receives packets produced by or destined to their hosts. Stub ASes represent the majority of the Internet and they have two types: single-homed ones and multi-homed ones. Today, Border Gateway Protocol (BGP) is the de facto standard interdomain routing protocol used in the Internet. BGP is a path vector protocol that works by sending routing advertisements. There are two variants of BGP. The external BGP (eBGP) variant is used to announce the reachable prefixes on a link between routers that are part of distinct ASes. The internal BGP (iBGP) variant is used to distribute the best routes learned from neighboring ASes inside an AS. A key feature of BGP is that it allows each network operator to define its routing policies implemented by configuring the BGP attributes. BGP routing attributes include Next-hop, AS-Path, Local-Preference, MED (Multi-Exit Discriminator), Origin, etc.

Before designing traffic engineering techniques, we must study the stability of interdomain routing and the characteristics of interdomain traffic. As explained in [2], most of the network prefixes do not change for several days or even a few weeks and the routes modified by the interdomain traffic engineering are guaranteed to remain constant over a long period of time. Many studies show that most of the traffic is concentrated in a small part of the prefixes [2]. Only by selecting popular prefixes reasonably, interdomain traffic engineering can redistribute a large amount of traffic. Popular prefixes with large volume traffic, the relative stability of traffic and interdomain routing are the preconditions of interdomain traffic engineering.

There are two main solutions in interdomain traffic engineering, MPLS (Multi-Protocol Label Switching) based solution and BGP based solution [1]. The MPLS based solution is generally an extension of intradomain traffic engineering based on MPLS and follows its framework. The deployment of the solution needs all the ASes within the traffic engineering, which is very difficult. There are still some unsolved problems in this kind of solution. The BGP based solution is divided into incoming traffic engineering and outgoing traffic engineering. Modifying the flow of the inbound traffic with BGP requires changing how remote ASes choose their best

BGP routes to reach the local AS. Since it is relatively difficult to influence the route policy in remote ASes, more of the current effort has been spent on outbound aspects. The outbound traffic engineering is generally implemented by influencing the decision process of BGP. In this paper, we focus on BGP-based outbound interdomain traffic engineering.

3 Related Work

The traffic engineering objectives that an AS may want to optimize will depend on its size, the type of business it focuses on and the relationships with other ASes. Stub ASes care more about aspects related to the distribution of the traffic over egress points than end-to-end properties. They also want to maintain the lowest communication cost at the same time. The stub ASes whose main business is to provide access to contents may prefer to optimize the outgoing traffic while the stub ASes whose main business is to provide dial-up, broadband access services may want to optimize how the traffic enter their network. For transit ASes, they care about both the distribution of traffic inside their ASes and among their interdomain links. We should build different models for stub ASes and transit ASes.

BGP-based outbound traffic engineering is essentially a problem of egress router selection and a NP problem. So we need heuristic algorithms to solve this problem. A systematic BGP-based traffic engineering for the outbound traffic of stub ASes at a very limited cost in terms of iBGP messages are presented in [3]. They use an efficient multi-objective evolutionary algorithm to evenly distribute the total traffic over the available providers and minimize the cost of the traffic. The cost of the traffic models the volume-based billing performed by transit ISPs. The multi-objective evolutionary heuristic is described in detail in [4]. The objectives considered in [4] are the following: minimizing the burden on BGP and optimizing one or several objectives defined on the traffic exchanged with other ASes or on the distribution of the traffic inside the AS. They implement a NSGA-II like algorithm. To avoid making the search space grow very large by sampling the whole search space, their heuristic iterate over the BGP routing changes by trying to add one BGP routing changes at each generation. The evolutionary heuristic is also used in [5]. The problem tackled in [5] consists in balancing the outbound traffic of a transit AS over its Internet connections while minimizing the cost of the traffic to cross its internal topology. They propose Tweak-it, a tool that based on the steady-state view of BGP routing inside the AS and the traffic demands of the AS, computes the BGP updates to be sent to the ingress routers of a transit AS to engineer its interdomain traffic over time. Tweak-it is based on two components: a scalable BGP simulator (C-BGP) [6] that reproduces the steady-state behavior of BGP routing and the multi-objective evolutionary heuristic. The work of this paper is based on this tool.

Taking into account the stability of BGP routing and the difficulty of real-time collection of traffic and routing updates, it is more practical to provide an offline tool for large ASes. The tool provided by this paper is offline.

4 A Multi-objective Genetic Algorithm

We design our method based on the Tweak-it tool proposed in [5]. Tweak-it is composed of two parts: C-BGP and the multi-objective evolutionary heuristic algorithm. Our contributions in this paper are the optimization and improvements of the algorithm and the re-development of C-BGP to apply a systematic BGP-based interdomain traffic engineering to a real network.

C-BGP is an efficient open-source BGP simulator written in C programming language which can model networks as large as the Internet. The network is represented as a graph where nodes are routers, and edges are links between routers. Each edge is weighted by the IGP metric of the corresponding link. C-BGP uses Dijkstra's Shortest Path First (SPF) to model IGP routing instead of modeling the specific implementation details in IGP. C-BGP is aimed at computing the outcome of the BGP decision process and it computes for each router the routes selected toward all the interdomain prefixes. This output can then be used to replay how the traffic was routed by the routers of the AS. To model BGP, the nodes which are considered as BGP routers are fitted out with additional data structures: a local RIB (Loc-RIB) and adjacent RIBs (Adj-RIBs). The Loc-RIB is used to store the best BGP routes while the Adj-RIBs contain alternative routes. These data structures form the basis of the algorithm. C-BGP is easily configurable through a CISCO-like command-line interface. To verify the simulation results, we add a command to C-BGP which will be explained in Sect. 5.

The second part of the tool is the multi-objective genetic algorithm. As practical traffic engineering objectives can be conflicting, the interdomain traffic engineering problem is intrinsically a multi-objective optimization problem. The algorithm we use as a basis can deal with one or several objectives definede tffic ehanged wher ASes or on the distribution of the traffic inside the AS. We define the imbalance ratio to measure the balance of interdomain traffic, specified as

$$(\max(\mathrm{tr_i}) - \min(\mathrm{tr_i}))/\sum\nolimits_{i=1}^{n} \mathrm{tr_i} i = 1, \ldots, n \tag{1}$$

where n denotes the number of eBGP peers and $\mathrm{tr_i}$ denotes the amount of traffic sent to the peer i. In this paper, for the provincial network of China Mobile, our first traffic objective is to reduce the imbalance ratio as much as possible to evenly distribute the interdomain traffic. The second objective is minimizing the IGP cost of the traffic inside the transit AS, specified as

$$\sum\nolimits_{\forall(i,p)}^{n} \mathrm{traffic}(i,p) * \mathrm{IGP\_cost}(i,p,e) \tag{2}$$

where $\mathrm{traffic}(i,p)$ denotes the amount of traffic received by ingress router i having as destination external prefix p and $\mathrm{IGP\_cost}(i,p,e)$ represents the IGP cost for ingress i to reach egress router e.

Unlike the general genetic algorithms, this algorithm doesn't encode the possible solutions and has no crossover and mutation. Instead, the steady-state policy routing made by BGP constitutes the search space of the multi-objective genetic algorithm. We do not describe this algorithm in detail but refer to [5] for the original idea and to [3, 4] for a thorough explanation. The core points of the algorithm are briefly mentioned here. This algorithm is implemented as a Perl script. The first phase of the script is the initialization interacting with C-BGP. It takes the internal topology, IGP weights, and BGP routing policies of each BGP router. Then the Perl script can build the C-BGP configuration file which will be injected into C-BGP. And after we inject RIBs of the border routers, C-BGP will populate the BGP routing tables and reproduce the steady-state policy routing made by BGP which constitutes the search space of the genetic algorithm. Then the multi-objective genetic algorithm will compute the tweakings to be performed to engineer the interdomain traffic. A tweaking is a <ingress, prefix, exit> tuple. The genetic algorithm relies on a population of individuals. Each individual is a potential solution, i.e. a set of tweakings to be applied. At the first generation, we start with a population of individuals initialized at the default solution found by BGP routing containing no BGP routing changes. At each generation, we parse the whole population and for each individual we choose an additional randomly <ingress, prefix> pair. Then, we try to apply the tweaking by computing the new values of the traffic objectives for every reachable exit peer. If the objectives have been optimized, we accept the tweaking. The number of generations is upper bounded by MAXGEN, the maximum number of tweakings allowed. Finally, after MAXGEN generations, we can select the best solution according to our optimization objectives.

Here we focus on our improvements of the algorithm. The paper improves the algorithm on two aspects: adding filtering mechanism and expanding the scope of application of the algorithm by solving the defects of the original algorithm.

4.1 Adding Filtering Mechanism

At each generation, the original algorithm tries to add one tweaking. It randomly chooses a <ingress, prefix> and tries every reachable exit peer to test whether the traffic objectives are optimized. Taking into account the fact that we usually tweak the traffic to the exit routers whose traffic on its interdomain link is smaller than average, we can add filtering mechanism for the selection of exit routers. In our algorithm, we only try the reachable exit peers whose traffic is below average. This will narrow the search space largely and shorten the running time of the algorithm.

4.2 Expanding the Scope of Application of the Algorithm

The original algorithm cannot efficiently deal with the following kind of scenario. Multiple eBGP peers belonging to a same AS are attached to one egress router, as shown in Fig. 1. The router R1 and R2 are egress routers. The router R3 and R4 are ingress routers. The best BGP routes chosen by each ingress router in the transit AS to reach prefix p are also shown in Fig. 1. As we can see in Fig. 1, the router R2 has two eBGP peers R6 and R7 in AS A.

To tackle the problem exists in the original algorithm, we must also record the corresponding eBGP peers when recording the best and alternative routes the ingress routers use to reach the prefixes. For instance, both R2 and R3 use R6 as the exit peer. Since C-BGP only stores the Next-hop and AS-Path attributes for the BGP routes, we need to do additional work in the algorithm. For routers like R2 whose Next-hop is just

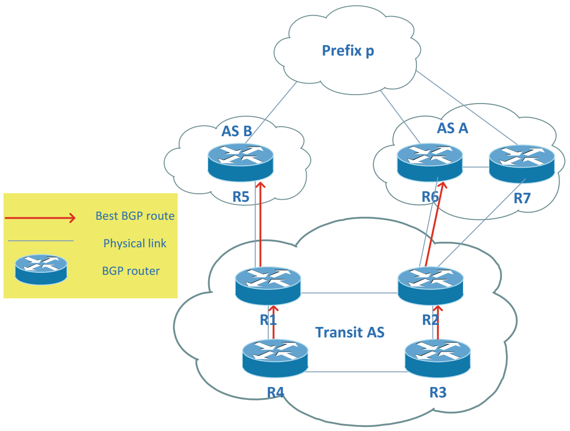


Fig. 1. The scenario that the original algorithm cannot deal with

the eBGP peer, we can simply record the Next-hop as the exit peer. But for routers like R3 whose Next-hop is a router inside the AS, we should find the exact exit peer. Taking into account the propagation of BGP and the interactions between IGP and BGP, we find the fact that if the BGP Next-hop is a router inside the AS, the exit peer the Next-hop router use to reach the prefix is actually the exit peer we are looking for. This is a key point here. For instance, we can find the exit peer R6 for R3 through R2. In the original algorithm, an egress router has only one eBGP peer in a neighboring AS by default and tweakings are defined in terms of egress routers. In our algorithm, through the additional information of exit peers, we can define tweakings in terms of exit peers, tweak the traffic more accurately and expand the scope of application of the algorithm.

5 Applications of the Genetic Algorithm

After we solve the problem exists in the original algorithm, we can apply our method to a provincial network of China Mobile. AS shown in Fig. 2, the simplified architecture is an abstraction of one provincial network of China Mobile. All the routers in the provincial network are both ingress routers and egress routers at the same time. The traffic objectives we consider here are evenly distributing the traffic over the interdomain links and minimizing the IGP cost of the traffic inside the provincial network as described in Sect. 4.

The first thing we should do is gathering the information of the network to inject into C-BGP which will build the search space of the genetic algorithm. It is necessary to collect the information such as: internal topology (including nodes, links and link weights), BGP routing policies and the RIBs of the border routers to be injected into C-BGP.

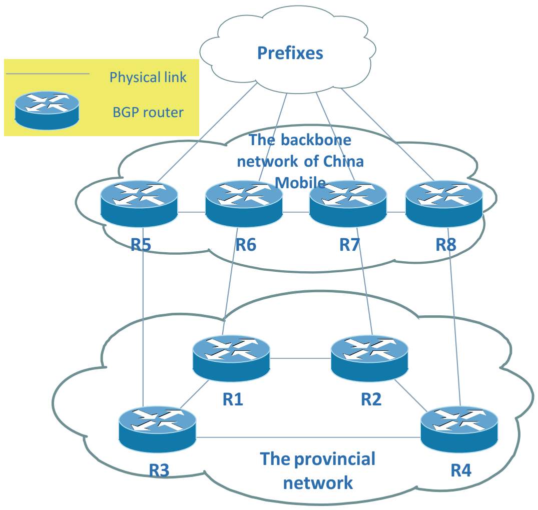
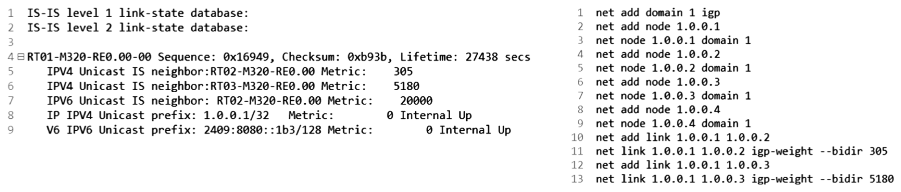


Fig. 2. The provincial network

5.1 Gathering the Information of Topology and IGP Weights

Large scale ISPs usually use IS-IS as the intradomain routing protocol and the provincial network of China mobile is no exception. We can extract information of all the nodes and links inside this network and the corresponding IGP weights from the IS-IS link-state database. We can get the database from any of the routers inside the AS. For reasons of confidentiality, the IP addresses, node names and link weights are changed.

We design a Python script to parse the file and output a C-BGP configuration file as shown in Fig. 3 by which C-BGP will compute the shortest paths from one router to another. In order to save space, we will not introduce the parsing process in detail.



```
IS-IS level 1 link-state database:
IS-IS level 2 link-state database:

RT01-M320-RE0.00-00 Sequence: 0x16949, Checksum: 0xb93b, Lifetime: 27438 secs
   IPV4 Unicast IS neighbor:RT02-M320-RE0.00 Metric:    305
   IPV4 Unicast IS neighbor:RT03-M320-RE0.00 Metric:    5180
   IPV6 Unicast IS neighbor: RT02-M320-RE0.00 Metric:    20000
   IP IPV4 Unicast prefix: 1.0.0.1/32   Metric:        0 Internal Up
   V6 IPV6 Unicast prefix: 2409:8080::1b3/128 Metric:        0 Internal Up
```

```
net add domain 1 igp
net add node 1.0.0.1
net node 1.0.0.1 domain 1
net add node 1.0.0.2
net node 1.0.0.2 domain 1
net add node 1.0.0.3
net node 1.0.0.3 domain 1
net add node 1.0.0.4
net node 1.0.0.4 domain 1
net add link 1.0.0.1 1.0.0.2
net link 1.0.0.1 1.0.0.2 igp-weight --bidir 305
net add link 1.0.0.1 1.0.0.3
net link 1.0.0.1 1.0.0.3 igp-weight --bidir 5180
```

Fig. 3. The IS-IS database and the C-BGP configuration file

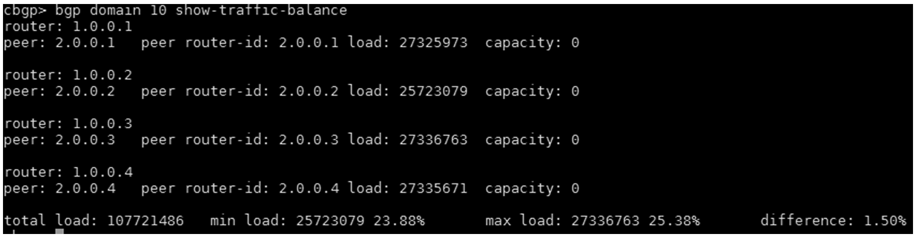
5.2 Gathering the Information of BGP Routing

To model the routing of BGP, we must get the information of all the BGP sessions established by the routers first. In our example, the iBGP sessions are full mesh and each BGP router has an eBGP peer belong to the backbone network of China Mobile as we can see in Fig. 2. C-BGP can load into one BGP router a dump of RIB captured on a real router and the RIB dump must be provided in ASCII MRT format. In the same way, we get the RIBs of the real routers and write Python scripts to parse the files. To ensure the correctness of BGP routing, we should get RIBs of all the BGP routers inside the AS.

5.3 Collecting Traffic and the Validation of the Algorithm

After gathering the information of the network, we also need the information of traffic. We collected a day of the traffic of the province and we can see that a limited fraction of the prefixes carry the vast majority of the interdomain traffic and these popular prefixes have stable BGP routes.

Then we can use the multi-objective genetic algorithm to compute the BGP updates to optimize the traffic objectives. C-BGP can load traffic but cannot analyze the traffic. To verify the simulation results, we add a command to C-BGP as shown in Fig. 4. With the comparison of imbalance ratio before and after the BGP updates, we can verify the effectiveness of the algorithm.



```
cbgp> bgp domain 10 show-traffic-balance
router: 1.0.0.1
peer: 2.0.0.1   peer router-id: 2.0.0.1 load: 27325973  capacity: 0

router: 1.0.0.2
peer: 2.0.0.2   peer router-id: 2.0.0.2 load: 25723079  capacity: 0

router: 1.0.0.3
peer: 2.0.0.3   peer router-id: 2.0.0.3 load: 27336763  capacity: 0

router: 1.0.0.4
peer: 2.0.0.4   peer router-id: 2.0.0.4 load: 27335671  capacity: 0

total load: 107721486   min load: 25723079 23.88%       max load: 27336763 25.38%       difference: 1.50%
```

Fig. 4. The command added to C-BGP

5.4 Comparison with the Original Algorithm

Under the same parameter condition, we run the original algorithm and our algorithm separately. In Table 1, we show the results of the two algorithms where we optimized the balance of the outgoing traffic of the provincial network while minimizing the cost of the traffic to cross its network (IGP cost metric), by relying on as few BGP routing changes as possible.

Table 1. Results of two algorithms.

Results	Original algorithm	Improved algorithm
Imbalance ratio	2.49%	1.18%
Running time	19.89 s	16.24 s

The imbalance ratio is 43.17% by default in our provincial network. We will say the interdomain traffic is well balanced when the ratio is under 10%. As we can see, the traffic can both be well balanced by the two algorithms but the running time of our algorithm is decreased because of the reduced search space of the algorithm.

5.5 Discussion on Different Parameters

As most of the traffic is concentrated in a small part of the prefixes, we just need to tweak some popular prefixes carrying the vast majority of the traffic. The algorithm takes as a parameter a percentage of the total traffic. This will greatly reduce the search space. But percentage is not as small as possible to ensure adequate search space and good optimization results. In this section, we focus on discussing the influence of different percentages on the performance and validity of the algorithm.

First, we talk about the relationship between the percentage parameter and the number of prefixes needed to tweak. As presented in Fig. 5 we can see a sharp decrease in the numbers of early period and a good percentage will greatly reduce the search space of the algorithm. When the percentage of the total traffic is decreased from 100 to 95, the number of prefixes is almost down to 1/5. When the percentage is below 70%, there are not enough prefixes.

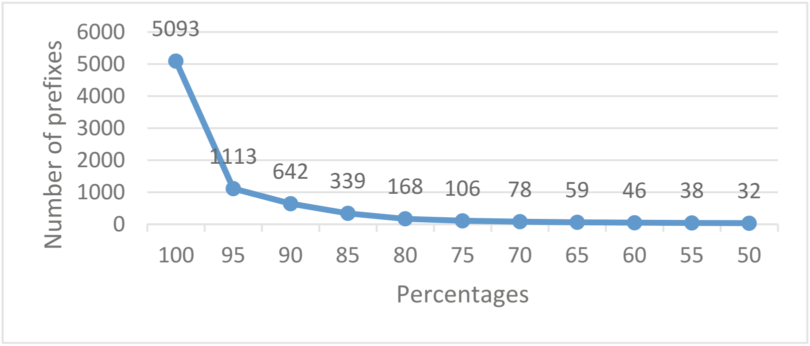


Fig. 5. Relationships between the percentage parameter and the number of prefixes

In Fig. 6, we can see the relationships among the percentage parameter, the imbalance ratio and the running time. All the different parameters can get a good balance of the interdomain traffic and the results are all acceptable. In our example,

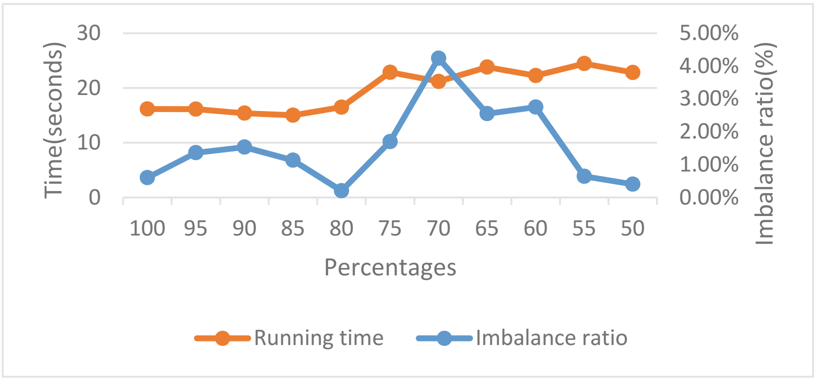


Fig. 6. Relationships among the percentage parameter, the imbalance ratio and the running time

85% to 95% of total traffic to be considered have relatively less running time. In a word, 85% to 95% can get relatively good optimization results and less time used for our example network. The percentage parameter should not be too big nor too small. Network operators can adjust the parameter according to their actual demands.

6 Conclusions

In this paper, we introduce the genetic algorithms in BGP-based interdomain traffic engineering. Then we propose an improved genetic algorithm and apply it to a provincial network of China Mobile. At last, we discuss the influence of different parameters in practical application. This systematic method in BGP-based interdomain traffic engineering has great practical significance for the network operators.

References

1. Luo, W., Wu, J., Xu, K.: Survey on inter-domain traffic engineering research. Tsinghua University (2006). (in simplified Chinese)
2. Wang, D.: Study on BGP-based interdomain traffic engineering. Northeastern University (2006). (in simplified Chinese)
3. Uhlig, S., Bonaventure, O.: Designing BGP-based outbound traffic engineering techniques for stub ASes. Comput. Commun. Rev. **34**(5), 89–106 (2004)
4. Uhlig, S.: A multiple-objectives evolutionary perspective to interdomain traffic engineering in the internet. In: Workshop on Nature Inspired Approaches to Networks and Telecommunications (2004)
5. Uhlig, S., Quoitin, B.: Tweak-it: BGP-based interdomain traffic engineering for transit ASs. In: Next Generation Internet Networks, pp. 75–82. IEEE (2005)
6. Quoitin, B., Uhlig, S.: Modeling the routing of an autonomous system with C-BGP. IEEE Netw. **19**(6), 12–19 (2005)