Time-Efficient Algorithms for BGP Route Configuration

Tat Wing Chim and Kwan L. Yeung
Dept. of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong
E-mail: {twchim, kyeung}@eee.hku.hk

***Abstract*—Based on the concept of *most popular prefix first*, two efficient algorithms for BGP route configuration are proposed. The first algorithm MPPF_SES is designed for solving the single egress selection (SES) problem, and the second algorithm MPPF_MES is for multiple egress selection (MES). MPPF_MES has two variants, one aims at minimizing the total amount of resources consumed for carrying the transit traffic, and the other tries to minimize the egress link capacity required. Compared with the existing algorithms, a comparable performance in terms of network resources consumed can be obtained. In case of SES, our MPPF_SES can carry a given traffic load with much lower egress link capacity requirement. In case of MES, our MPPF_MES tends to provide a more stable performance. Last but not the least, our proposed algorithms have a much lower time complexity than the existing approach.**

Keywords – BGP; Egress Selection; OSPF Weight Assignment

I. INTRODUCTION

A major responsibility of an Internet Service Provider (ISP) is to provide transit service for its neighbors. Traffic goes into and out of an ISP through a set of border routers which are managed by the ISP and are connected to its neighbors via a set of peering edge links. It has been noticed that the peering links are often the bottlenecks in the Internet, so it is important that those links can be utilized efficiently.

On the other hand, an ISP wants to minimize its operational cost on carrying the transit traffic. This can be achieved by minimizing the amount of network resources consumed. In this paper, we assume that traffic is characterized by flows, where each flow is identified which ingress router the traffic enters the AS, and where this traffic goes to. The problem of BGP route configuration is to determine a set of egress edge links (thus egress border routers) to carry the transit flows such that the network resources consumed locally is minimized and the egress edge link capacity is not violated.

To fully understand the mechanisms available to control the selection of egress edge links for forwarding transit flows and to have a general picture of the problem, knowledge of Border Gateway Protocol (BGP) [1] is essential. In short, BGP is a path vector protocol under which routing decisions can be made based on policy. BGP divides the Internet into a collection of Autonomous Systems (ASes). An AS is defined as a set of routers under a single technical administration, e.g. an ISP. ASes exchange routing reachability information through external BGP peering sessions. A BGP speaking router (i.e. a border router) receives route advertisements from either external peers (in neighboring ASes) or internal peers (in local AS). Each advertisement contains a destination prefix, an IP address of the next-hop, a multi-exit discriminator (MED) and a list of AS numbers of the ASes along the path going to the destination. Depending on what is configured in the *import policy engine* of a router, some or all of these received routes will be included into its own routing table and advertised to its peers. The MED field can be used by an external peer to differentiate the preference of the local AS among the set of border routers in common with that peer. Within an AS, we may favor one advertisement (from an internal peer) over another by assigning a local preference to it. Such preference will be broadcast to all internal routers, and is only effective within an AS.

BGP import policy engine of a router selects the best routes according to a list of pre-defined criteria [1][2]. For a given prefix (i.e. a destination network), a single egress edge link will be selected to carry all the traffic destined to it. That means no matter where the transit traffic's ingress point is, as long as it goes to the same prefix, it will exit the local AS at the same egress edge link. The problem of determining the best single egress point for each given prefix is known as *single egress selection* (SES) problem. For a given prefix, the import policy engine can also be set to allow flows arrived at different ingress border routers to exit the local AS at different egress edge links. This can improve the network utilization but at the expense of higher BGP configuration overhead. The associated problem of determining the optimal set of egress links for each given prefix is called *multiple egress selection* (MES) problem.

In [2], BGP route configuration problems for both SES and MES are shown to be NP-hard. Based on the linear programming (LP) relaxation of the associated integer programming formulation, heuristic algorithms are designed to round the fractional solutions from LP to the nearby integers. We call the algorithms for solving SES and MES problems as Rounding_SES and Rounding_MES respectively. As the rounding process [3] would lead to a capacity violation on the egress edge links by a factor of up to 2, a feasible solution (i.e. all flows can be forwarded and there is no capacity violation on any egress link) may require more egress link capacity than that obtained by the LP.

In this paper, based on the concept of most popular prefix first, two time-efficient algorithms for BGP route configuration, MPPF_SES and MPPF_MES, are proposed. In the next section, both SES and MES problems are formally formulated. In Section III and IV, MPPF_SES and MPPF_MES algorithms are presented in detail. In Section V, their performance is compared with the existing algorithms. Finally, we conclude the paper in Section VI.

II. PROBLEM FORMULATION

A similar system model as that in [2] is adopted. We assume that all routers and intra-domain links have infinite capacities while edge links connecting to other ASes are bottlenecks with finite capacities. Each edge link carries traffic in both directions, ingress and egress. The capacity allocated to each direction is pre-determined and dedicated.

Multiple edge links may be connected to the same border router. A neighboring AS may be connected to the local AS through a direct edge link, or indirectly via other ASes. Each neighbor may be connected to the local AS through multiple edge links. For simplicity, we assume that the prefixes received by the AS are non-overlapping and so cannot be aggregated as in [4]. We further assume that route advertisements for any prefix are advertised to all connecting neighbors so that neighbors are able to choose which ingress points to use. (In this paper, we assume that each neighbor makes a random selection.)

Given a set of neighbors $A_1, \ldots, A_H$ and a set of edge links $b_1, \ldots, b_I$, $R(i)$ returns the border router of edge link b_i. For each neighbor A_h, let $In(h)$ denote the set of edge links through which A_h may send in the transit traffic. Each edge link b_j has an *egress* capacity constraint C_j. (We have assumed that the *ingress* capacity on each edge link is sufficient.) The intra-domain topology provides the shortest path distance between any two edge links b_i and b_j, which we denote as $d(i,j)$. External BGP peering sessions at the border routers receive advertisements for network prefixes across the edge links. Let $P_1, \ldots, P_K$ denote the set of prefix advertisements received across all edge links. For each prefix P_k, let $Out(k)$ denote the set of edge links at which an advertisement for P_k has been received.

For each traffic flow going to destination prefix P_k and coming from neighbor A_h via ingress edge link b_i, let $t(h,i,k)$ denote its traffic volume. Then the product $t(h,i,k)\cdot d(i,j)$ is the internal cost (i.e. the resources required) to carry the traffic $t(h,i,k)$ from edge b_i to edge b_j. Finally, let f be the egress edge assignment function and $f(h,i,k)$ returns the assigned egress edge link for the above traffic flow. Table I summarizes the notations used in this paper.

TABLE I NOTATIONS USED IN THIS PAPER

Notation	Description
$A_1, \ldots, A_H$	Set of AS neighbors
$b_1, \ldots, b_I$	Set of edge links
$P_1, \ldots, P_K$	Set of destination network prefixes
$r_1, \ldots, r_X$	Set of border routers
$R(i)$	Border router of edge link b_i
$In(h)$	Set of ingress edge links from neighbor A_h
$Out(k)$	Set of egress edge links for P_k
$d(i, j)$	Intra-domain distance between b_i and b_j
$t(h, i, k)$	The amount of traffic from neighbor A_h via ingress edge link b_i and destined for prefix P_k
C_j	Egress link capacity for edge b_j
$N(j)$	Number of prefixes that has advertisements to b_j
f	Function that maps traffic to an egress point
$t(h,i,k)\ d(i,j)$	Cost of carrying traffic $t(h,i,k)$ from b_i to b_j

A. Problem Statement for SES Problem

Compute an assignment function f:($\{1, \ldots, H\}$, $\{1, \ldots, I\}$, $\{1, \ldots, K\}$)→($\{1, \ldots, I\}$) from (neighbor, ingress edge link, prefix) to egress edge link, such that the total amount of network resources consumed for carrying the transit traffic, called *solution cost*, is minimized

$$\min\left(\sum_{h,i,k} t(h,i,k)\cdot d(i, f(h,i,k))\right). \qquad (1)$$

and f satisfies the following constraints:

- If $f(h,i,k) = j$, then $j \in Out(k)$.
- Egress capacity constraints of edge links are satisfied, i.e.
$$\sum_{h,i,k: f(h,i,k)=j} t(h,i,k) \le C_j \text{ for all } j.$$
- The same egress edge link is assigned to all transit traffic going to the same prefix.

B. Problem Statement for MES Problem

The problem statement for MES problem is the same as that for SES except that different ingress routers may choose different egress edge links for transit flows going to the same destination prefix. On the other hand, the same ingress router will always choose the same egress edge link for all transit traffic going to the same destination prefix.

III. ALGORITHM FOR SES PROBLEM

A. MPPF_SES Algorithm

Let p_k be the total amount of traffic destined to prefix P_k. Our proposed algorithm aims at giving the highest route selection priority to the prefix with the largest amount of traffic destined to it, i.e. $\max_k\{p_k\}$. The idea is that if no priority is given to the prefix with the largest value of p_k, it is very likely that the most desirable egress link leading to this prefix would have been occupied by others. The potential extra cost of carrying this traffic on alternative egress link would be very high. Since the route configuration priority is based on p_k, we call our algorithm *Most Popular Prefix First* (MPPF). For solving the single egress selection problem, we call the resulting algorithm MPPF_SES. Its detailed operations are shown in Fig. 1. In SES problem, the assignment function f is independent of the neighbor A_h and the prefix P_k and so we use the short form $f(k)$ for $f(h,i,k)$.

MPPF_SES Algorithm

Inputs: $t(h, i, k)$ for all h, i, k; $d(i, j)$ for all i, j
Outputs: f

1. For all j, set $U_j = 0$. /* U_j records the amount of traffic assigned to b_j. */
2. Compute
$$p_k = \sum_{h=1}^{H}\sum_{i=1}^{I} t(h,i,k) \text{ for all } k \in [1,K]$$
Sort k in non-increasing order of p_k to form an ordered list **K**.
4. For all k, set $f(k) = 0$. /* Initialize assignment to null. */
5. For each k in the ordered list **K**
Sort $j \in Out(k)$ in non - decreasing order of $\sum_{h=1}^{H}\sum_{i=1}^{I} d(i,j)\cdot t(h,i,k)$
to form an ordered list **J**.
For each j in the ordered list **J**
If $U_j + p_k \le C_j$ then /* Check for capacity violation */
Set $f(k) = j$. /* Egress link b_j is selected */
Set $U_j = U_j + p_k$.
If $f(k) = 0$, quit program. /* No feasible solution */

Fig. 1 The MPPF_SES algorithm

Step 2 calculates the amount of traffic destined to each prefix. Step 3 sorts them into an ordered list of non-increasing order. Step 5 is responsible for selecting the egress link with the lowest cost for each entry in the ordered list. Starting from the first prefix P_k in the list, we select the egress edge in $Out(k)$ which yields the minimum resulting cost and yet has enough residual capacity to accept all traffic for P_k (i.e. p_k). Note that

$\sum_{h=1}^{H}\sum_{i=1}^{I} d(i,j)\cdot t(h,i,k)$ is the cost of using b_j to carry all the traffic destined for P_k. If the minimum cost egress link does not have enough capacity, we select the second minimum one and so on. In the worst case that all egress links in $Out(k)$ do not have enough capacity, then no feasible solution can be found.

B. Time Complexity

In practice, advertisements for any prefix P_k will only be broadcast to a small number of egress edges (usually less than 50) no matter how large an AS is. This can be observed from the real BGP routing table data collected by the Route Viewer server [6]. Therefore we can treat $|Out(k)|$ as a constant. As such, the time complexity of MPPF_SES is dominated by the sorting algorithm in Step 3. Assume Quick Sort [5] is used. The resulting time complexity of MPPF_SES is $O(K^2)$, where K is the number of prefixes. From [2], the time complexity of Rounding_SES algorithm is $O((K^2I)\log(K+I))$, where I is the number of edge links. So our MPPF_SES is more efficient.

IV. Algorithm for MES Problem

A. Proximity Constraint & MPPF_MES Algorithm

The major departure of MES problem from SES is that the transit traffic going to the same prefix may exit from different egress links if they enter the AS at different ingress routers. That is, the egress edge link is determined jointly by which ingress router the traffic enters the AS, and where this traffic goes to.

An important consideration in MES is the proximity constraint [2]. The purpose of it is to ensure that the distance between the ingress router where a transit flow arrives, and the egress link found for this flow is the shortest among all other possible egress links. This helps to ensure that the local preference among routers in the local AS can be set properly. However, the local preference in BGP may not rely on the *actual* distance between two border routers. If the egress link selected is not the shortest in terms of distance $d(i, j)$, we can use the technique of OSPF weight assignment [7][8][9] to specify the desired paths and set the local preference in routers based on the assigned weights. As such, we can ignore the proximity constraints while solving for the MES problem. The overall solution cost thus found can be further reduced.

We can generalize the concept of *most popular prefix first* to MES. The resulting MPPF_MES algorithm also aims at giving the *prefix* with the largest amount of traffic destined to it the highest route selection priority. For a given prefix, priority is given to the *ingress router* that has the largest aggregated flow volume. This is because large traffic flows tend to be more difficult to assign. MPPF_MES algorithm is summarized by the pseudo codes in Fig. 2. In MES problem, the assignment function f depends on only the ingress router r_X and the prefix P_k and so we use the short form $f(x,k)$ for $f(h,i,k)$.

Following the similar argument before, the worst case time complexity of MPPF_MES can be found as $O(K^2 + KX^2)$. From [2], Rounding_MES algorithm has a complexity of $O((HIK)\log(HIK))$, where H is the number of neighboring ASes and I is the number of edge links.

B. A Variant of MPPF_MES Algorithm

In solving the MES problem, the probability that an egress link for a certain prefix is occupied by traffic for other prefixes is much higher than that in the SES problem. This is because the transit traffic destined to a prefix can be forwarded onto more than one egress links. This increases the minimum egress link capacity required to forward a given traffic pattern. To address this, we propose another variant of MPPF_MES algorithm for minimizing the egress link capacity required. We call it MPPF_MESv2.

MPPF_MES Algorithm

Inputs: $t(h, i, k)$ for all h, i, k; $d(i, j)$ for all i, j
Outputs: f

1. For all j, set $U_j = 0$.
2. Compute
$$p_k = \sum_{h=1}^{H}\sum_{i=1}^{I} t(h,i,k) \text{ for all } k \in [1, K]$$
3. Sort k in non-increasing order of p_k to form an ordered list **K**
4. For each prefix P_k, compute
$$c_{k,x} = \sum_{h=1}^{H}\sum_{i:R(i)=x} t(h,i,k) \text{ for all } x \in [1, x]$$
5. For all (x, k), set $f(x, k) = 0$. /* Initialize assignment to null */
6. For each k in the ordered list **K**
   Sort x in non-increasing order of $c_{k,x}$ to form an ordered list **X**
   For each x in the ordered list **X**
   Sort $j \in Out(k)$ in non - decreasing order of
   $\sum_{h=1}^{H}\sum_{i:R(i)=x} d(i,j)\cdot t(h,i,k)$ to form an ordered list J
   For each j in the ordered list **J**
   If $U_j + c_{k,x} \le C_j$ then
   Set $f(x, k) = j$.
   Set $U_j = U_j + c_{k,x}$.
   If $f(x, k) = 0$, quit program.

Fig. 2 The MPPF_MES algorithm

Both MPPF_MESv1 (i.e. the MPPF_MES algorithm in Fig. 2) and MPPF_MESv2 give the route selection priority to the most popular prefix. And for a given prefix, priority is given to the ingress router with the largest flow destined to it. The difference is only at choosing the suitable egress link. In MPPF_MESv1, the egress link that gives the lowest cost will be selected first; whereas in MPPF_MESv2, the egress link that has the least number of prefix advertisements is selected first. The number of prefix advertisements implies the potential number of flows/load an edge link needs to carry. Giving selection priority to the edge with the lowest load tends to balance the traffic on all egress links. This can also lower the probability that an egress link for a certain prefix is occupied by traffic for other prefixes. As a result, the minimum capacity required to forward a given traffic pattern can be reduced.

Since this assignment process does not aim at minimizing the solution cost, unacceptably high solution cost may be resulted. Therefore an additional phase is designed to reassign some traffic flows to less expensive edges. In this phase, we consider the prefixes one by one in non-increasing order of p_k. Then for each prefix, we consider the traffic flows for that prefix one by one, again in non-increasing order of their traffic volumes. Then for each of such flows, we try to reassign it to another egress edge (in $Out(k)$) such that it has enough residual capacity and has a minimum cost.

The time complexity of MPPF_MESv2 is still $O(K^2 + KX^2)$.

V. EXPERIMENTS AND PERFORMANCE

A. Network model

Given the number of border routers for the local AS X, the number of neighboring ASes H, and the number of prefixes that transit traffic addressed to K, the network topology for simulation is generated as follows:

- The intra-domain distance between any two border routers is uniformly distributed (with integer value) over the range {10…100}. We assume that $d(i, j) = d(j, i)$ for any two routers r_i and r_j.
- The multihoming degree of each border router is randomly selected from 1 to 3. Each border router is then associated with the corresponding number of edge links. All edge links are uniquely numbered to form the edge set.
- The size of set $In(h)$ for each neighbor A_h is randomly selected from 1 to 3. The elements of $In(h)$ are randomly selected from the edge set.
- For each prefix P_k, the size of $Out(k)$ is randomly selected from 2 to 5. The elements of $Out(k)$ are again randomly selected from the edge set.

B. Traffic model

Assume that every neighboring AS has some traffic destined for every destination prefix. This gives HxK traffic instances/flows, forming an HxK traffic matrix. Each entry of the matrix represents the traffic volume of a flow. Its value is uniformly distributed between 0 and 20. Looping is not allowed. So if a neighboring AS has forwarded an advertisement for prefix P_k to the local AS, this AS cannot inject traffic for prefix P_k into the local AS.

In Figs. 3-6, each point of simulation results is obtained by taking the average of 10 independent experiments, each with a randomly generated network topology (with $X = 25$, $H = 12$ and $K = 35$) and traffic matrix.

C. MPPF_SES vs other algorithms

We first study the single egress selection problem. In addition to our proposed MPPF_SES algorithm, the following algorithms are implemented for comparison:

- BTF (Biggest Traffic First): In BTF, the set of traffic is sorted in non-increasing order. An attempt is then made to assign an egress link for each traffic $t(h, i, k)$. If prefix P_k has already been assigned to an egress point, which still has sufficient capacity, we simply send the traffic to this already selected egress point. If prefix P_k has not been assigned, we find the closest egress point that has the capacity to accept the traffic.
- Rounding_SES: Please refer to [2] for details.

Let the (egress) capacity of all egress edge links be equal. Fig. 3 shows the percentage of total traffic sent against the capacity of each egress link. Less than 100% means that not all traffic generated/arrived can be carried by the egress links. In other words, no feasible solution can be found. Fig. 4 shows the corresponding solution cost normalized by the infinite capacity solution, which is obtained by assuming the capacity of each egress link is infinite. This serves as a lower bound. From Fig. 3, we can see that the corresponding minimum capacity required to send 100% offered traffic are 210, 240, and 420 for using MPPF_SES, Rounding_SES and BTF respectively. Our MPPF_SES gives a save of 50% as compared with BTF, and 12.5% as compared with Rounding_SES.

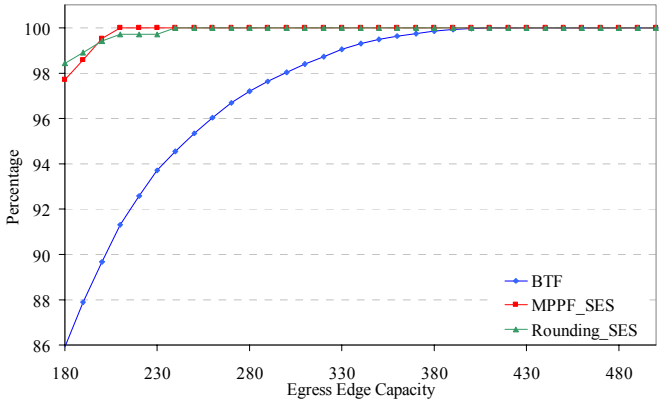

Fig. 3 Percentage of traffic sent in the SES experiments

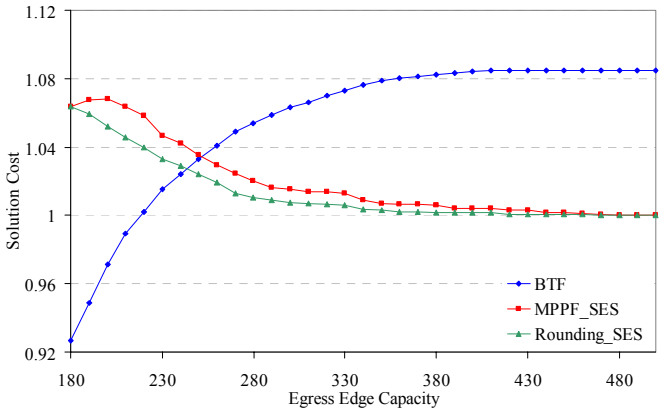

Fig. 4 Normalized solution cost in the SES experiments

From Fig. 4, we can see that the normalized solution cost for BTF is the worst and in fact, cannot converge to the case of infinite capacity. The solution cost for Rounding_SES has a small gain over MPPF_SES algorithm when egress link capacity is small. At the point where Rounding_SES can send 100% of the offered traffic (so is MPPF_SES), the solution cost for MPPF_SES is only 1.26% higher than that of Rounding_SES. It should be noted that if an algorithm cannot send 100% of the offered traffic, it is in general not meaningful to compare solution cost.

D. MPPF_MES vs other algorithms

Next we focus on the performance of the algorithms for multiple egress selection. Our proposed two variants of MPPF_MES are compared with the following algorithms:

- HPR (Hot Potato Routing): It sends all incoming traffic to the closest allowable egress point while assuming all egress links have infinite capacity. Just like the infinite capacity solution for single egress selection, this is to serve as a lower bound for comparison.
- EBTF (Extended Biggest Traffic First): In EBTF, the set of traffic is sorted in non-increasing order. An attempt is then made to assign an egress link for each traffic $t(h, i, k)$. If the pair $(R(i), k)$ has already been assigned to an egress point, we simply send the traffic to the already selected egress point, if capacity permits. If the pair $(R(i), k)$ has not been

assigned, we find the closest egress point that has the capacity to accept the traffic.

- Rounding_MES: We modified the original Rounding_MES in [2] by dropping its proximity constraint.

Fig. 5 shows the percentage of traffic sent against the egress link capacity. Fig. 6 shows the corresponding solution cost normalized to the infinite capacity solution. From Fig. 5, we can see that the corresponding minimum capacity required to send 100% offered traffic are 150, 220, 190 and 320 using Rounding_MES, MPPF_MESv1, MPPF_MESv2 and EBTF respectively. For MPPF_MESv1, MPPF_MESv2 and EBTF, continuously 100% of offered traffic can be forwarded with capacities greater than their minimum values. However, for Rounding_MES, although 100% of the offered load can be forwarded when the egress link capacity is 150, only about 99% of offered load can be forwarded with capacity of 230. Only when the capacity is larger than 240, 100% of the offered traffic can be sent. This unstable performance can be explained by the fact that the Rounding_MES algorithm consists of two phases: the first phase produces fractional assignments while the second phase rounds fractional assignments into integer assignments. Assume a solution is found when egress link capacity is small. As the egress link capacity continues to increase, the first phase tries to change the fractional assignments to obtain an even lower solution cost. However, this may cause the second phase incapable of finding a "good" integer assignment, which causes the percentage of traffic sent drops below 100%. From the experiments we conducted, it is observed that such fluctuations in performance are unlikely to occur in EBTF and MPPF_MES algorithms.

From Fig. 6, we can see that as egress link capacity increases, the solution costs obtained from all 4 algorithms converge very quickly to that of HPR. At egress link capacity of 240, the solution cost for MPPF_MESv1 is only 0.72% higher than that of Rounding_MES.

VI. CONCLUSIONS

In this paper, based on the concept of most popular prefix first, two new algorithms for BGP route configuration have been proposed. The first algorithm MPPF_SES was designed for solving the single egress selection (SES) problem; whereas the second algorithm MPPF_MES was for multiple egress selection (MES). MPPF_MES has two variants, one aims at minimizing the overall solution cost, and the other tries to minimize the egress link capacity required. We also showed that the proximity constraint [2] is not necessary, which in fact could worsen the solutions found. Compared with the existing BGP route configuration algorithms, we found that a comparable performance in terms of solution cost can be obtained. In case of SES, our MPPF_SES can carry a given load with a much lower egress link capacity requirement. In case of MES, our MPPF_MES provides a more stable performance than the existing ones.

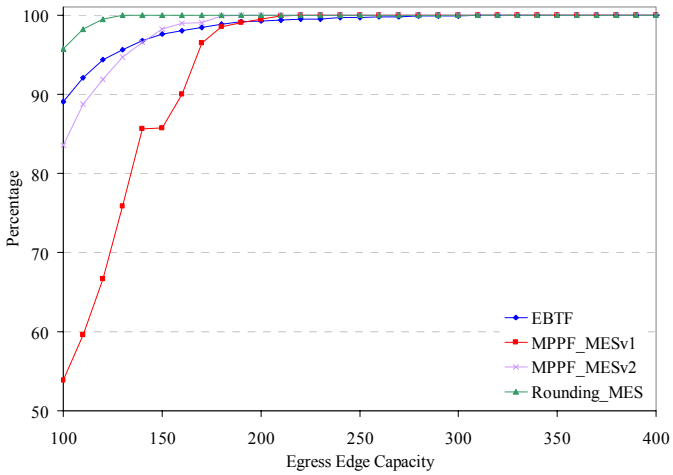

Fig. 5 Percentage of traffic sent in the MES experiments

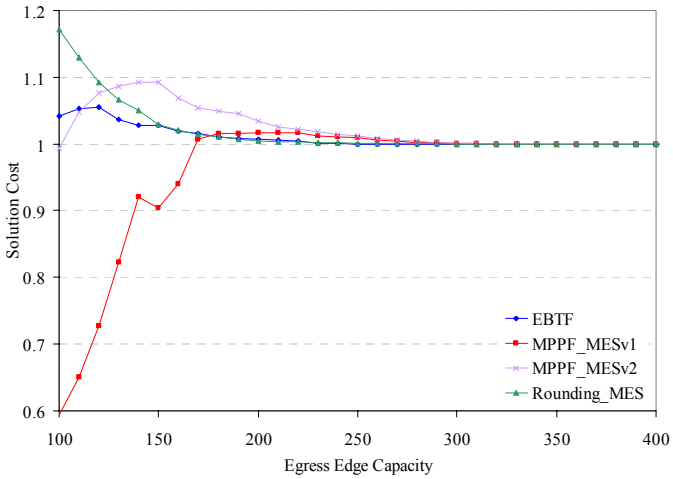

Fig. 6 Normalized solution cost in the MES experiments

REFERENCES

[1] Y. Rekhter and T. Li, "A border gateway protocol 4," Internet-Draft (RFC1771), February 1998.

[2] T. C. Bressoud, R. Rastogi and M. A. Smith, "Optimal Configuration for BGP Route Selection," *IEEE INFOCOM*, April 2003.

[3] D. B. Shymoys and E. Tardos, "An approximation algorithm for the generalized assinment problem," *Mathematical Programming A*, vol. 62, pp. 461 – 474. 1993.

[4] R. P. Draves, C. King, S. Venkatachary and B. D. Zill, "Constructing Optimal IP Routing Tables," *IEEE INFOCOM*, March 1999.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, "Introduction to Algorithms," Second Edition, The MIT Press, 2001.

[6] http://www.antc.uoregon.edu/route-views/

[7] B. Fortz and M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights," *IEEE INFOCOM*, March 2000.

[8] Y. Wang, Z. Wang and L. Zhang, "Internet Traffic Engineering without Full Mesh Overlaying," *IEEE INFOCOM*, April 2001.

[9] M. Ericsson, M. G. C. Resende and P. M. Pardalos, "A Genetic Algorithm for the Weight Setting Problem in OSPF Routing," *J. Combinatorial Optimization*, vol. 6, no. 3, pp. 299 – 333, September 2002.