Received: June 17, 2017

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

IP Network Design by Modified Branch Exchange Method

Kairat Jaroenrat[1] **Natchamol Srichumroenrattana**[2*]

[1]*Faculty of Engineering at Kamphaeng Saen, Kasetsart University, Thailand*
[2]*Faculty of Management Science, Nakhon Pathom Rajabhat University, Thailand*
* Corresponding author's Email: k.natchamol@hotmail.com

Abstract: One popular IP network routing protocol is OSPF (Open Shortest Path First), which uses algorithms to find its own path by calculating the path with the least weight from source to destination. As a result, the design and routing of the data flow may not meet the requirements. To meet the efficiency of cable installation and routing, the Branch Exchange for network design process was applied. With the ability to select the number of links to be exchanged with others, we can trade-off between the design efficiency and the processing time. We used the ability of the Branch Exchange method to determine a network with an efficient traffic flow distribution. We calculated the end-to-end delay time using the M/M/1 model, then analyzed the network design efficiency of 8- and 9-node networks compared to the Simplex method. For 8-, 9-, and 10-node networks, we compared the MENTOR-II method to traffic flow and network complexity. The results showed that the efficiency of the Branch Exchange method in terms of delay and cost closely align to the Simplex method but with faster calculation time. However, more exchange links (r) means a processing time closer to the Simplex method.

Keywords: Branch exchange, IP network design, Weight exchange

1. Introduction

The Internet is a very important technology and necessary for connecting worldwide networks together. The devices that are used to connect networks are called routers. The current widely popular routing protocol is OSPF (Open Shortest Path First), which has many embedded advantages, such as using Dijkstra's algorithm to find the shortest route by itself, recognizing change in the topology pattern, and quickly updating the network's route. However, because OSPF routers use this algorithm to find their own paths with the shortest routes, the traffic flow may not match the overall needs of some flow paths. In addition, the efficiency and cost of network cable installation are the most important factors to be considered.

Most network design algorithms are fairly complex, such as the Simplex linear programming method and the Branch Exchange method. The simple Branch Exchange method [1] requires a complexity of $O(N^5)$, where N is the number of nodes, which is prohibitive for moderating to large networks. However, as the Internet becomes the lifeline to business and commercial applications, one must be aware of several issues in designing large data networks (i.e., Internet Service Providers' backbone). One of the most important is that the IP network is a datagram network in which the routing protocols route traffic over the path with the shortest distance—that is, the sum of link weights. However, the link weight setting for an optimum routing pattern is a complex problem that is possibly unfeasible [2-3]. Sridharan, Gúerin, and Diot [4] explored the trade-off that exists between performance and the overhead associated with the additional configuration steps for IP networks. Sqalli, Sait, and Asadullah [5] engineered a Tabu Search Iterative heuristic using two different implementation strategies to solve the OSPF weight setting problem for link failure scenarios. But these conventional techniques are only the IP traffic assignment algorithms for existed networks which cannot be used to design IP networks.

The least complex and high-speed heuristic network design algorithm is MENTOR (Mesh Network Topological Optimization and Routing [6], which selects a link to be installed as assigned traffic flows over it. MENTOR is often used to design virtual circuit packet switching networks, such as Frame Relay, ATM, and MPLS. After that, Cahn [7] proposed a modified version of MENTOR called MENTOR-II in which appropriate link weights are assigned to the installed links such that the traffic always routes to the path with minimum distance and can be applied to IP networks. However, both MENTOR and MENTOR-II give a good performance by using some value of the design parameters [8-9].

Branch Exchange [10] is a distributed design method which consists of adding, dropping, or exchanging links to achieve better performance. In order to determine the efficiency and cost-effectiveness of cable installation and traffic routing, we applied our modified Branch Exchange method for IP network design to get the advantage of the ability to adjust between the design efficiency and the processing time of the design process.

2. Modified branch exchange method

Branch Exchange is a distributed design method that can be applied to IP network design. It starts by connecting the station with any topology and then adding, dropping, or exchanging links to achieve better performance through a link exchange called permutation. Thus, the default link weights are assigned and exchanged to achieve better performance, as shown in Fig. 1.

An overview of the modified Branch Exchange method is shown in Fig. 1. It starts with the following user's requirements: (1) The number of network nodes. (2) The node degree, which is the ratio of the number of signal lines to the number of nodes. (3) Traffic requirement to flow between the two nodes. (4) Umax, which is the maximum acceptable traffic utilization of links. (5) Umin, which is the minimum acceptable traffic utilization of links.

The initial network with default weight links is created with the node degree required from the user. The network links are then exchanged to get the best network delay. Finally, the link weights are exchanged to get the best network with the best link weights.

The initial network creation process begins by installing links between node pairs with the most traffic requirements in order to meet the required node degree. Then the links' load, suitable bandwidth, and delay cost are calculated. If there is any breach of utilization conditions, links with breach of conditions must be dropped and the link with the next largest traffic requirement will be added instead until the network meets the user's node degree.

The links' bandwidth exchange process starts by calculating each link's network load and exchanging each link's bandwidth, which has the r-rank highest traffic load with other links. When each new network is created by the links' bandwidth exchange process, then the new delay cost and the links' network load is calculated. But if there is any link with breach of utilization conditions, this network will be dropped. After calculating all forms of exchange, the network with the least delay cost is selected; if there is not an acceptable network, a network with minimal condition violations is chosen to be the solution.

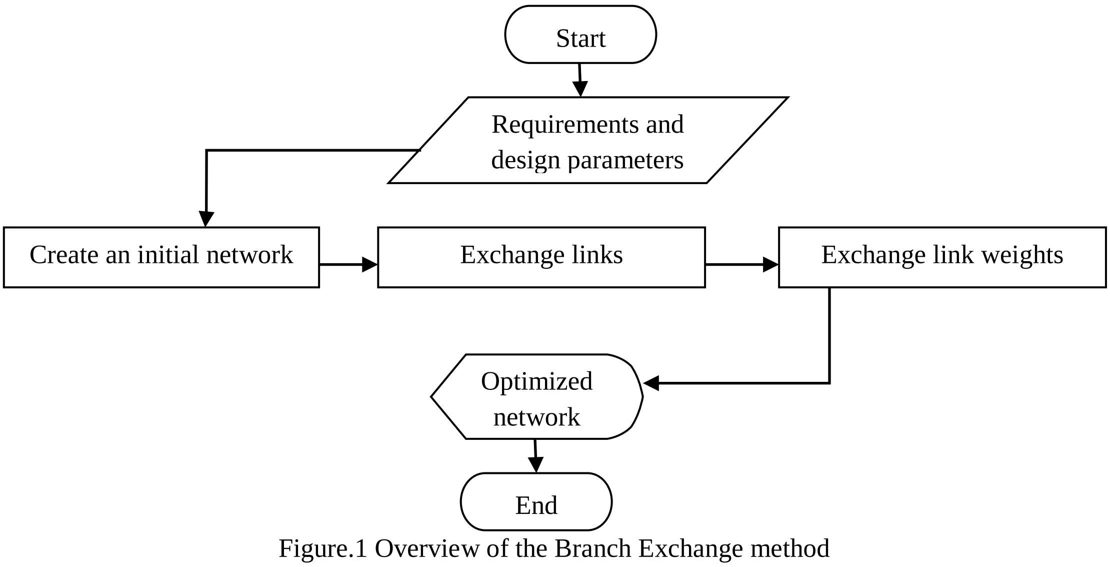

Figure.1 Overview of the Branch Exchange method

After the solution network is created by the links' bandwidth exchange process, a default weight is assigned to each link. But if the default weight is not suitable for the solution network, then we used a link's weight exchange process to produce a suitable link weight for each link. We begin by calculating each link's network load and exchanging each link's weight, which has the r-rank highest traffic load with the other links. When each new network is created by the links' weight exchange process, then the new delay cost and the links' network load is calculated. But if there is any breach of utilization conditions, this network will be dropped. After calculating all forms of exchange, the network with the least delay cost is selected; if there is not an acceptable network, a network with minimal condition violations is chosen to be the final solution.

Finally, the output network consists of the following: (1) Good link connections and bandwidths among nodes. (2) Link weights, which allow for efficient OSPF routing. (3) Total delay cost of the network. (4) Overall time involved in the process. (5) Installation cost of the network.

3. Experimental design

We conducted an experiment to compare the traffic flow performance of networks designed by the modified Branch Exchange method with the same requirement networks designed by the Simplex method and the MENTOR-II algorithm. The experiment was conducted with three node numbers—8, 9, and 10—and six node degree values, 1.0, 1.2, 1.4, 1.6, 1.8, and 2.0. We used DELITE [7] to generate three different data sets for each node number and traffic requirement, with total inbound and outbound traffic for each data set equal to 100 Mbps, 150 Mbps, and 200 Mbps. A total of 162 data sets were used in this experiment.

In parts of the MENTOR-II algorithm, we used the design parameters [9] of $\alpha \in (0, 0.5, 1)$, $\rho \in (0.6, 0.7, 0.8)$, and s $\in$ (0.3, 0.4, 0.5) and then calculated the performance in the following areas.

3.1 Delay efficiency

Consider a directed network graph G = (N, A) with a capacity c_a for each $a \in A$ and, as defined in previous section, d_{st} denotes the amount of traffic flow between s and t. Let f_a^{st} indicate how much traffic flow from s to t over arc a, traffic load l_a over link $a \in A$ is the sum of all f_a^{st}. It is suggested by Fortz et al. [3] to measure the network performance by delay cost function in Eq. (1).

$$\Phi = \sum_{a \in A} \phi_a(l_a, c_a), \qquad (1)$$

where $\phi_a(l_a, c_a)$ is an M/M/1 queuing theory style link cost function given by Eq. (2).

$$\phi_a(l_a, c_a) = l_a / (c_a - l_a) \qquad (2)$$

With this function, it is more expensive to send flow along arcs whose loads approach capacity, which is what we want. However, the function does not deal with overloaded links—that is, $l_a \geq c_a$. To overcome this problem, $l_a /(c_a - l_a)$ is approximated by a piece-wise linear function [3] $\phi_a(0) = 0$ and derivative in Eq. (3).

$$\phi'_a(l_a, c_a) = \begin{cases} 1 & \text{for} \quad 0 \leq l_a / c_a < 1/3, \\ 3 & \text{for} \quad 1/3 \leq l_a / c_a < 2/3, \\ 10 & \text{for} \quad 2/3 \leq l_a / c_a < 9/10, \\ 70 & \text{for} \quad 9/10 \leq l_a / c_a < 1, \\ 500 & \text{for} \quad 1 \leq l_a / c_a < 11/10, \\ 5000 & \text{for} \quad 11/10 \leq l_a / c_a < \infty. \end{cases} \qquad (3)$$

Then we can calculate the delay efficiency as a difference between the total network delay [3] of the original method and the new method from Eq. (4).

$$\text{Delay Efficiency} = \frac{D_{L\ or\ M}}{D_B} \times 100\ \% \qquad (4)$$

When delay efficiency is a performance in terms of total link delay in the network, $D_{L\ or\ M}$ is a total links' network delay designed by the Simplex or MENTOR-II method and D_B is a total links' network delay designed by the Branch Exchange method.

3.2 Processing time efficiency

We can calculate the efficiency of the time spent in the network design process as shown in Eq. (5).

$$\Delta Times = |T_B - T_{L\ or\ M}| \qquad (5)$$

When $\Delta Times$ is a processing time efficiency of the network, $T_{L\ or\ M}$ is the time spent on the design process of the Simplex or MENTOR-II method and T_B is the time spent on the design process of the Branch Exchange method.

3.3 Installation cost efficiency

When it is necessary to consider the cost of installing cables, then we can calculate the installation cost efficiency [11] of the design process as shown in Eq. (6):

$$\text{Cost Efficiency} = \frac{C_{L\,or\,M}}{C_B} \times 100\ \% \qquad (6)$$

When Cost Efficiency is an installation cost efficiency of the network, C_B is the total network installation cost that was designed by the Branch Exchange method and $C_{L\,or\,M}$ is the network installation cost that was designed by the Simplex or MENTOR-II method.

4. Experiment results and analysis

4.1 Experiment results

We analyzed the performance among the networks designed by the Branch Exchange, Simplex, and MENTOR-II methods in the three areas of delay, processing time, and installation cost [11]. The results are shown in Tables 1–9.

Tables 1–3 show the average delay cost of the networks designed by the Branch Exchange, Simplex, and MENTOR-II methods. The results for each method were averaged from the results of 162 networks, except for Branch Exchange, which are separated for each value of r.

Tables 4–6 show the average processing time of networks designed by the Branch Exchange, Simplex, and MENTOR-II methods. The results for each method were averaged from the results of 162 networks. As with the delay cost, the Branch Exchange results are separated for each value of r.

Table 1. Average delay cost of 8-node networks

Delay Cost			
r	Branch Exchange	Simplex	MENTOR-II
1	1679.18	1463.48	1895.67
2	1652.54		
3	1640.71		
4	1622.6		
5	1608.69		
6	1589.23		
7	1582.77		

Table 2. Average delay cost of 9-node networks

Delay Cost			
r	Branch Exchange	Simplex	MENTOR-II
1	2106.04	1899.43	2485
2	2104.71		
3	2079.83		
4	2067.54		
5	2062.04		
6	2058.63		
7	2058.63		

Table 3. Average delay cost of 10-node networks

Delay Cost		
r	Branch Exchange	MENTOR-II
1	3642.91	3849.75
2	3424.09	
3	3272.63	
4	3142.57	
5	3085.29	
6	3047.43	
7	2377.33	

Table 4. Average processing time for designing 8-node networks

Processing Time			
r	Branch Exchange	Simplex	MENTOR-II
1	0.01	5558.03	0.02
2	0.08		
3	0.32		
4	15.41		
5	21.02		
6	203.32		
7	880.79		

Table 5. Average processing time for designing 9-node networks

Processing Time			
r	Branch Exchange	Simplex	MENTOR-II
1	0.03	338283.55	0.024
2	0.11		
3	0.53		
4	5.81		
5	89.26		
6	956.67		
7	7208.6		

Table 6. Average processing time for designing 10-node networks

Processing Time		
r	Branch Exchange	MENTOR-II
1	0.07	0.025
2	0.26	
3	3.39	
4	61.37	
5	2521.02	
6	16990.53	
7	49215.95	

Table 7. Average installation cost of 8-node networks

Installation Cost			
r	Branch Exchange	Simplex	MENTOR-II
1	2476.42	2189.96	2243.6
2	2476.42		
3	2476.42		
4	2476.42		
5	2476.42		
6	2476.42		
7	2476.42		

Table 8. Average installation cost of 9-node networks

Installation Cost			
r	Branch Exchange	Simplex	MENTOR-II
1	3226	2825.38	2902
2	3226		
3	3226		
4	3226		
5	3226		
6	3226		
7	3226		

Table 9. Average installation cost of 10-node networks

Installation Cost		
r	Branch Exchange	MENTOR-II
1	5163.56	3716.67
2	5163.56	
3	5163.56	
4	5163.56	
5	5163.56	
6	5163.56	
7	5163.56	

Tables 7–9 show the average network installation cost designed by the Branch Exchange, Simplex, and MENTOR-II methods. The results for each method were averaged from the results of 162 networks. As with the delay cost and processing time, the Branch exchange results are separated for each value of r.

5. Results analysis

This section analyzes the total delay cost, processing time, and installation cost of 8-, 9-, and 10-node networks. Figs. 2–4 show the delay cost comparison of the modified Branch Exchange method to the Simplex and MENTOR-II methods. By selecting r links to be exchanged in the Branch Exchange method, we found that the delay efficiency of the Branch Exchange method will fluctuate based on the number of selected links (r). The greater the value of r, the less network delay, which makes this method better than the MENTOR-II method and more closely aligned to the Simplex method. As shown in Tables 1–3, with r equal to 1, we can see that the average delay cost of 8-, 9-, and 10-node networks is 1,679.18, 2,106.04, and 3,642.91, respectively, but when r equals 7, the average delay cost of 8-, 9-, and 10-node networks is reduced to 1,582.77, 2,058.63, and 2,377.33, respectively.

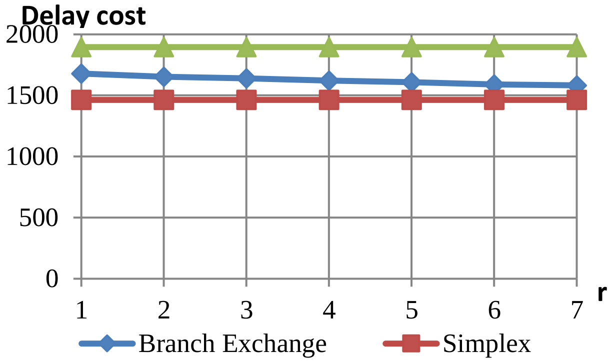

Figure.2 Average delay cost in 8-node networks

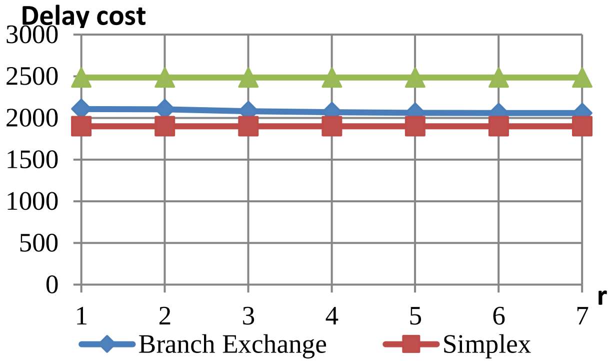

Figure.3 Average delay cost in 9-node networks

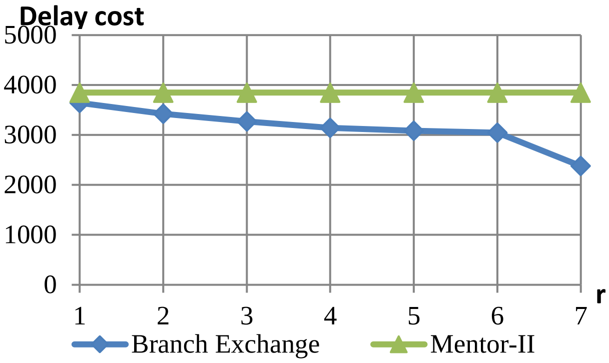

Figure.4 Average delay cost in 10-node networks

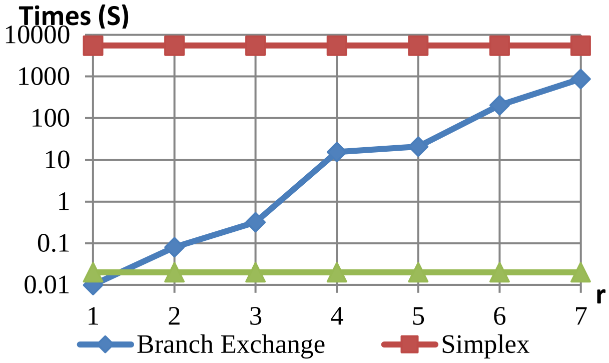

Figure.5 Average processing time of 8-node networks

Figs. 5–7 show the processing time comparison of the modified Branch Exchange method to the Simplex and MENTOR-II methods. By selecting r links to be exchanged in the Branch Exchange method, we found that the processing time of the Branch Exchange method will also fluctuate based on the number of selected links (r). The greater the value of r, the more processing time needed by the Branch Exchange method, which makes it worse than the MENTOR-II method and more closely aligned to the Simplex method. As shown in Tables 4–6, with r equal to 1, the average processing time of 8-, 9-, and 10-node networks is 0.01, 0.03, and 0.07 seconds, respectively, but when r equals 7, the average processing time of 8-, 9-, and 10-node networks is increased to 880.79, 7,208.6, and 49,215.95 seconds, respectively.

Figs. 8–10 show the installation cost comparison of the modified Branch Exchange method to the Simplex and MENTOR-II methods. By selecting r links to be exchanged in the Branch Exchange method, we found that the installation cost of the Branch Exchange method did not fluctuate based on the number of selected links (r). As shown in Tables 7–9, the average installation cost of 8-, 9-, and 10-node networks is 2,476.42, 3,226, and 5,163.56, respectively.

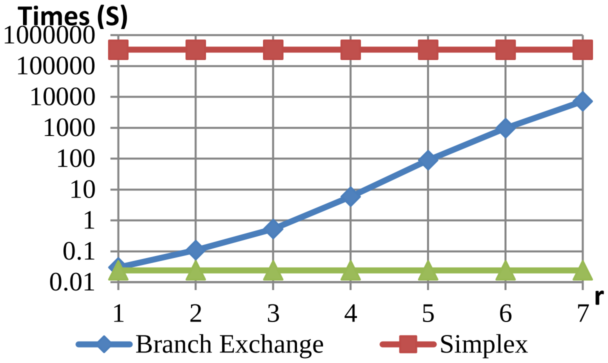

Figure.6 Average processing time of 9-node networks

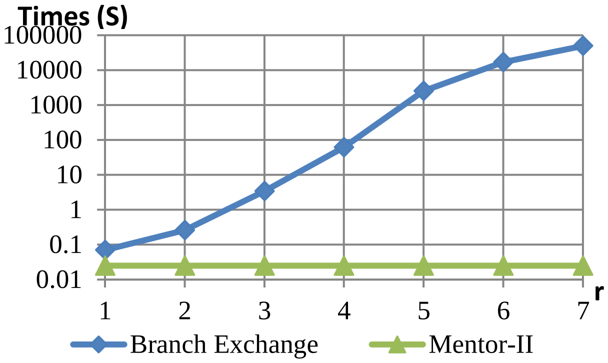

Figure.7 Average processing time of 10-node networks

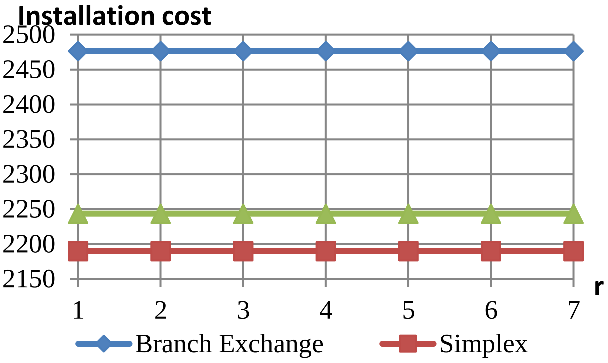

Figure.8 Average installation cost in 8-node networks

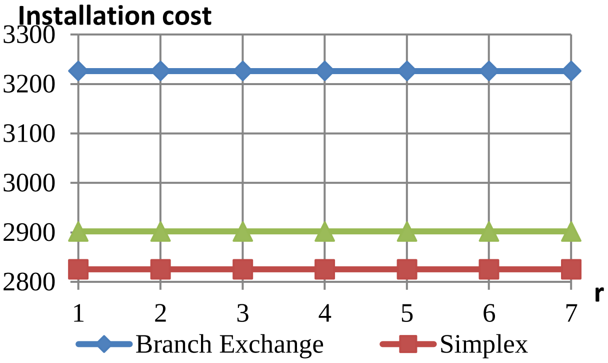

Figure.9 Average installation cost in 9-node networks

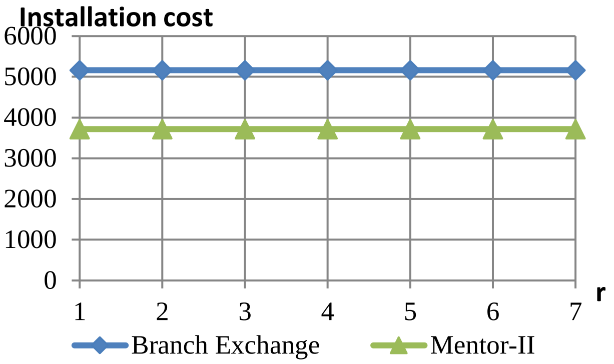

Figure.10 Average installation cost in 10-node networks

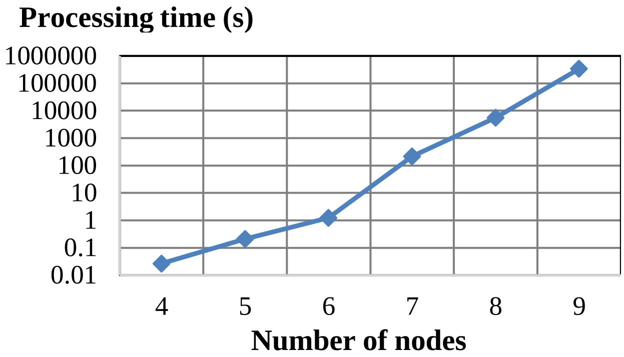

Figure.11 Simplex method processing time

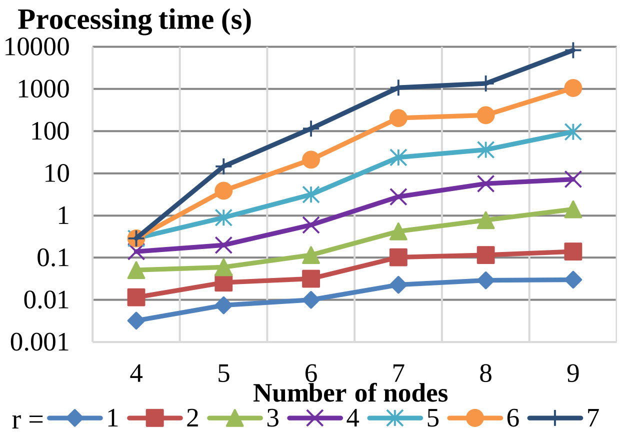

Figure.12 Branch Exchange method processing time

The Simplex method network design and routing process takes a very long time, especially for large networks. Fig. 11 shows the relationship between the number of nodes and the processing time, which can be approximated to an exponential equation of $y = 0.0003e^{3.3572x}$ when x and y are the number of nodes and the processing time, respectively.

The Branch Exchange network design and routing process takes less time than the Simplex method. Fig. 12 shows the relationship between the number of nodes and the processing time for each r. These relationships can be approximated to exponential equations of $y = 0.0026e^{0.4584x}$, $y = 0.0055e^{0.5413x}$, $y = 0.0098e^{0.8177x}$, $y = 0.0313e^{0.9469x}$, $y = 0.0527e^{1.2992x}$, $y = 0.2684e^{1.3965x}$, and $y = 0.7678e^{1.5465x}$, for r equals 1 to 7, respectively.

From these estimated relationships, we can conclude that the Simplex method is not suitable for large networks. Using the example of a 1-month processing time, the Simplex method could provide only a 7-node network, while the Branch Exchange method could provide a maximum 45-node network with r=1.

6. Conclusion

Adopting the Branch Exchange method for network design and routing is a highly flexible option. With the ability to select the number of links (r) to be exchanged with others, we can trade-off between the design efficiency and the processing time. We used the ability of the Branch Exchange method to add, drop, and exchange links in order to determine a network with an efficient traffic flow distribution. We then analyzed the performance of network delay, processing time, and installation cost of the Branch Exchange method. The results showed the average delay performance compared to the Simplex method is equal to 91.77%, the average processing time compared to the Simplex method is 337,103.4 seconds faster, and the efficiency of installation cost is equal to 89.72% compared to the Simplex method. In comparison with the MENTOR-II method, the results showed that the average delay performance is equal to 118.27%, the average processing time is 1,180.12 seconds slower, and the efficiency of installation cost equals 74.42%. In addition, we found that given the greater the number of r, the delay and time performance of the Branch Exchange method would be even better. However, given more numbers of links (r), the Branch Exchange method required more processor time than the MENTOR-II method and thus aligned more closely to the Simplex method.

In conclusion, the Branch Exchange method offers better processing time but with slightly worse delay and installation cost efficiency compared to the Simplex method. But, in comparison with the MENTOR-II method, the Branch Exchange method offers better efficiency in network delay but with slightly worse processing time and higher installation cost. Thus more research works need to be done in the future to find a suitable technique to be added in our method for decreasing the processing time and the installation cost.

References

[1] A. Kershenbaum, *Telecommunications network design algorithms*, McGraw-Hill, New York, N.Y. 1993.

[2] W. Ben-Ameur, E. Gourdin, B. Liau, and N. Michel, “Dimensioning of Internet networks”, In: *Proc. of the second international workshop on the design of reliable communication networks*, Munich, Germany, pp. 56–61, 2000.

[3] B. Fortz, J. Rexford, and M. Thorup, “Traffic engineering with traditional IP routing protocols”, *IEEE Communications Magazine*, Vol.40, No.10, pp. 118–124, 2002.

[4] A. Sridharan, R. A. Gúerin, and C. Diot, “Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks”, *IEEE/ACM Transactions on Networking*, Vol.13, No.2, pp. 234–247, 2005.

[5] M. H. Sqalli, S. Sait, and S. Asadullah, “OSPF weight setting optimization for single link failures”, *International Journal of Computer Networks & Communications*, Vol.3, No.1, pp. 168–183, 2011.

[6] A. Kershenbaum, P. Kermani, and G. A. Grover, “MENTOR: An algorithm for mesh network topological optimization and routing”, *IEEE Transactions on Communications*, Vol.39, No.4, pp. 503–513, 1991.

[7] R. S. Cahn, *Wide area network design: Concepts and tools for optimization*, Morgan Kaufmann, San Mateo, CA, 1998.

[8] K. Jaroenrat, and P. Charnkeitkong, “On routing performance of MENTOR algorithm”, *WSEAS Transaction on Communications*, Vol.9, No.5, pp. 1769–1776, 2006.

[9] P. Charnkeitkong, K. Jaroenrat and P. Keatthalerenglit, “On IP Network Routing Performance of MENTOR-II Algorithm”, In: *Proc. of IT Towards Empowerment International Conference*, Bangkok, Thailand, pp.138-143, 2007.

[10]C. Ababei, and R. G. Kavasseri, “Efficient network reconfiguration using minimum cost maximum flow-based branch exchanges and random walks-based loss estimations”, *IEEE Transactions on Power Systems*, Vol.26, No.1, pp. 30–37, 2011.

[11]K. Jaroenrat, S. Chimmanee and P. Charnkeitkong, “Algorithms for IP networks design with ECMP routing enable”, In: *Proc. of 7th International Conference on Computing and Convergence Technology*, Seoul, South Korea, pp. 420–425, 2012.