Proceedings of the 10th WSEAS International Conference on COMMUNICATIONS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp419-424)

A Study on the Flow Assignment Efficiency of MENTOR Algorithm

KAIRAT JAROENRAT, CHOM KIMPAN, PISIT CHARNKEITKONG
Faculty of Information Technology
Rangsit University
52/347 Muang-Ake, Phaholyothin Rd., Lak-Hok, Pathumtani,
THAILAND
kairat@kairat.com, chom@rangsit.rsu.ac.th, pisitc@rangsit.rsu.ac.th

Abstract: - Mesh Network Topological Optimization and Routing (MENTOR) algorithm is a low complexity and efficient partial mesh networks design algorithm. This study explores the relation between design parameters and performance of traffic assignment of MENTOR algorithm. We analyze 384 networks designed by MENTOR for 4 sets of 50 nodes each with equivalently distributed demand and randomly generated locations. For each of these networks, the performances at normal load and at congestion threshold of MENTOR flow assignment are calculated and compared with the optimum solution obtained by solving the linear programming. It is found that the performances of MENTOR strongly depend on “slack”, a design parameter represented the different between the maximum and minimum utilization. For small value of slack, s=0.1, the performances of MENTOR keep very close to that of the optimum solution, while the maximum utilization hardly impact the performances. As the slack get larger the performances of MENTOR become worse and more depend on the maximum utilization.

Key-Words: - MENTOR, Network Design, Mesh Networks, Flow assignment, Network optimization

1 Introduction

The ultimate goal of network design is to obtain network that is able to support given traffic demands with highest performance and lowest cost. However, as Internet become life-line of business and commercial application, designers of large data network, i.e. Internet Service Providers (ISPs) backbone, have to aware of rapid growth and reserve the capacity for the future. Reserved capacity of a network depends on a number of network parameter such as maximum and minimum allowable link utilization. For example, setting high maximum link utilization often leads to a network with lower cost and reserved capacity, and vice versa.

A heuristic network design algorithm called MENTOR (Mesh Network Topological Optimization and Routing) [1] is a high-speed and very efficient design algorithm. MENTOR is flexible and can be used to design virtual circuit packet switching networks such as Frame Relay, ATM or even MPLS. When MENTOR decides to installed a link, at the same time, traffic flow over it is assigned. Flow assignment of MENTOR is not always optimal and strongly depends on network design parameters such as maximum and minimum link utilization.

This study investigates the relation between design parameters and performance of flow assignment of MENTOR algorithm. We analyze 384 networks designed by MENTOR for 4 sets of 50 nodes each with equivalently distributed demand and randomly generated locations. For each of these networks, the performances at normal load and at threshold of congestion of MENTOR flow assignment are calculated and compared with the optimum solution obtained by solving the linear programming.

2 Problem Formulation

2.1 MENTOR Algorithm

MENTOR algorithm is a low complexity heuristic network design algorithm. This low complexity is achieved by doing implicit routing over a link at the same time it is considered to be installed. For a given set of nodes N, demand matrix D and link cost matrix X, let $d_{s,t}$ and $x_{s,t}$ are the amount of traffic flow and link installation cost from s and t, respectively. The characteristics of network obtained by MENTOR algorithm are (1) traffic demands are routed on relatively direct paths (2) links have reasonable utilization and (3) relatively high capacity links are used.

MENTOR starts with clustering process. In this stage, nodes are classified in to end nodes and backbone nodes using a clustering algorithm. Examples of possible clustering algorithms are threshold clustering and K-mean clustering. However, we consider in this paper only the case where traffic demands are distributed equivalently among all nodes. Therefore, all nodes can be considered as backbone node.

Next, a good tree is formed to interconnect all (backbone) nodes. Kershenbaum et. al. [1] suggests to a use a heuristic, which can be thought of as a modification of Prim and Dijkstra algorithm. The algorithm almost the same manner as Dijkstra algorithm, but instead, each round, tree is expanded by connecting a tree node i to an out of tree node j to that minimize $\alpha \cdot L_i + x_{ic}$, where $0 \leq \alpha \leq 1$, and L_i is the cost of path from root along the tree to node i. Note that $\alpha = 0$ and 1 is corresponding to Minimum Spanning Tree and Shortest Path Tree, respectively.

Given a tree, the objective of MENOTR is to consider adding a direct link between each pair of nodes if the utilization is between the predefined maximum and minimum utilization. Let the maximum utilization be ρ, and the minimum utilization be $(1-s)\rho$, where slack s, $0 \leq s \leq 1$. Consider a pair of nodes A and B, let C_{AB} and l_{AB} be link capacity and accumulated load between A and B, respectively. If $l_{AB} < \rho C_{AB}\ (\tilde{1}-s)$, no link is added and all traffic l_{AB} is overflowed to the next most direct path. A link is added and no over flow traffic if $\rho\ C_{AB}\ (\tilde{1}\text{-s}) \leq l_{AB} \leq \rho C_{AB}$. However, if $l_{AB} > \rho C_{AB}$, a direct link is added only when splitting traffic among multiple route is possible. In this situation and a portion of traffic $l_{AB} - \rho C_{AB}$ is overflow to the next most direct path. Otherwise, if splitting traffic is not possible, no link is added and l_{AB} is overflowed to the next most direct path. Node pairs are sequenced such that a link between a pair is considered only when all traffic flows that could overflow to the ink are already considered.

MENTOR gives fairly good results and widely used to many type of networks, e.g. Frame Relay, ATM as well as MPLS. However, the impact of design parameters, e.g. ρ, s and α, on efficiency of traffic routing are not yet studied before.

2.2 Objective Function

Consider a directed network graph G = (N, A) with a capacity c_a for each $a \in A$ and as define in previous section, d_{st} denote the amount of traffic flow between s and t. Let f_a^t indicate how much of the traffic flow to t over arc a, traffic load l_a over link $a \in A$ is the sum of all f_a^t. It is suggested in [4] to measure the performance of network by cost function

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

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

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

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, i.e. $l_a \geq c_a$. To overcome this problem, $l_a /(c_a - l_a)$ is approximated by a piece-wise linear function $\phi_a(0) = 0$ and derivative

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

2.3 Optimum Solutions

With piece-wise linear cost function define by (3), the general routing problem can be formulated as the following linear programming [5].

$$\text{Min } \Phi = \sum_{a \in A} \phi_a \qquad (4)$$

Subject to:

$$\sum_{u:(u,v)\in A} f_{(u,v)}^t - \sum_{u:(v,u)\in A} f_{(v,u)}^t = \begin{cases} D^t, \text{ if } v = t. \\ -\ d_{s,t}, \text{ if } v = s \\ 0, \text{ otherwise}, \end{cases} \quad v,s,t \in N, \qquad (5)$$

$$\phi_a \geq l_a \qquad a \in A, \qquad (6)$$

$$\phi_a \geq 3l_a - 2/3c_a \qquad a \in A, \qquad (7)$$

$$\phi_a \geq 10l_a - 16/3c_a \qquad a \in A, \qquad (8)$$

$$\phi_a \geq 70l_a - 178/3c_a \qquad a \in A, \qquad (9)$$

$$\phi_a \geq 500l_a - 1468/3c_a \qquad a \in A, \qquad (10)$$

$$\phi_a \geq 5000l_a - 19468/3c_a \qquad a \in A, \qquad (11)$$

$$l_a = \sum_{t \in N} f_a^t \qquad a \in A, \qquad (12)$$

$$D^t = \sum_{s \in N} d_{s,t} \qquad (13)$$

$$f_a^t \geq 0 \qquad a \in A;\ t \in N. \qquad (14)$$

Constraints (5) are flow conservation constraints, constraints (6) – (11) describe the cost function, constraints (12) define the load on each arc, and constraint (13) defines D^t that represents all traffic headed toward destination node t.

However, general optimum solution is not fairly comparable with other traffic routing that have limits maximum link utilization, e.g. MENTOR algorithm. This is because the purpose of limiting maximum link utilization is to reserve capacity to handle more traffic load when network get congest. To take in to account the capacity reservation, the optimum solution with maximum link utilization ρ is obtained by solving (4) subject to (5) – (14) and additional constraint

$$l_a/c_a \leq \rho. \qquad (15)$$

2.4 Normalizing Routing Cost

Fortz and Thorup [3] proposed a normalizing scaling factor for the routing cost that makes possible comparisons across different network sizes and topologies:

$$\Phi_{\mathrm{UNCAP}} = \sum_{s,t\in NxN} d_{s,t} h_{s,t} \qquad (16)$$

where h_{st} = minimum hop count between s and t.

For any routing cost Φ, the scaled routing cost or normalized routing cost is defined as

$$\Phi^* = \Phi \,/\, \Phi_{\mathrm{UNCAP}} \qquad (17)$$

The above program is a complete linear programming formulation of the general routing problem. We shall use Φ to denote the optimal general routing cost.

3 Experiments

In order to evaluate the efficiency of flow assignment calculated by MENTOR algorithm, we analyze the performances of a number synthesized network and observe the relation between design parameters and performances.

3.1 Experiment Set Up

DELITE [6] is used to synthesize 4 sets of 50 nodes each with different node distribution obtained by varying SEED parameter. We shall refer to these set of nodes as N1, N2, N3 and N4. The traffic demand matrix for each set of nodes is also generated by DELITE with default setting and total traffic in and traffic out of each node are 100 Mbps.

By varying design parameters, a total of 384 MENTOR networks are generated for N1-N4 using the full-duplex link of capacity 45 Mbps. For each node sets, two groups of networks corresponding to Minimum Spanning Tree (α=0) and Shortest Path Tree (α=1) are generated. For each type of spanning tree, 48 networks are generated by varying of ρ, $\rho \in$ (0.4, 0.5, 0.6, 0.7, 0.8, 1.0) and s, $s \in$ (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8).

3.2 Experiment Results

For each of 384 networks, optimum routing solution with maximum link utilization constraint proposed in section 2.3 is solved by GLPK[7]. On Intel Pentium IV Xeon 3.3 GHz machine, it takes maximum 2 hours to solve the optimal routing problem. Normalized cost Φ^* of MENTOR flow assignment and optimal solution are calculated for different scaling of projected demand matrix.

Given network demands, performance of MENTOR flow assignment at normal load is measured by % of cost different from optimality

$$\Delta C = \frac{\Phi^*_M - \Phi^*_O}{\Phi^*_O} \times 100,$$

where Φ^*_M and Φ^*_O are normalized cost of MENTOR flow assignment, and that of optimum solution, measured at demand used to design the network respectively.

As seen in section 2, the cost function increase rapidly toward 5000 after the ϕ_a = 10 ⅔.The performance of MENTOR flow assignment at the threshold of congestion is measured by % of demand different from optimality

$$\Delta D = \frac{D_M - D_O}{D_O} \times 100,$$

where D_M and D_O are the scaling traffic demand of MENTOR flow assignment, and that of optimum solution measured when the cost Φ^* = 10⅔, respectively

The results are presented in Fig.1-16.

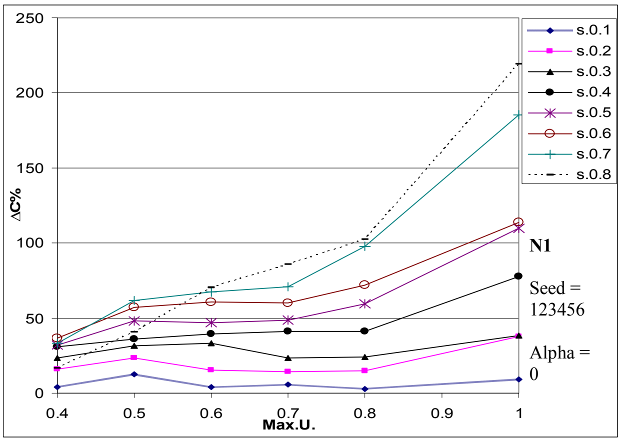

Fig.1. ΔC of networks with Alpha = 0 for N1.

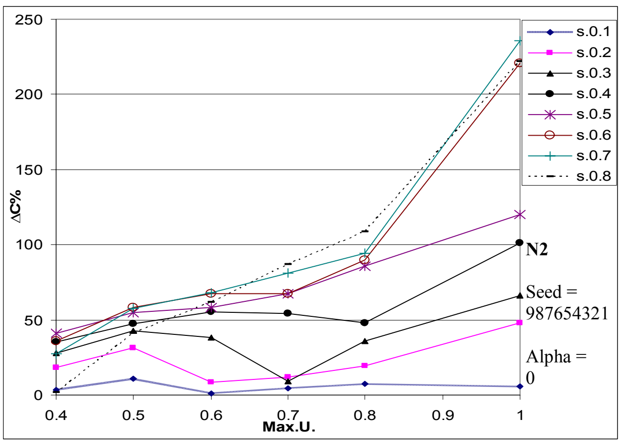

Fig.2. ΔC of networks with Alpha = 0 for N2.

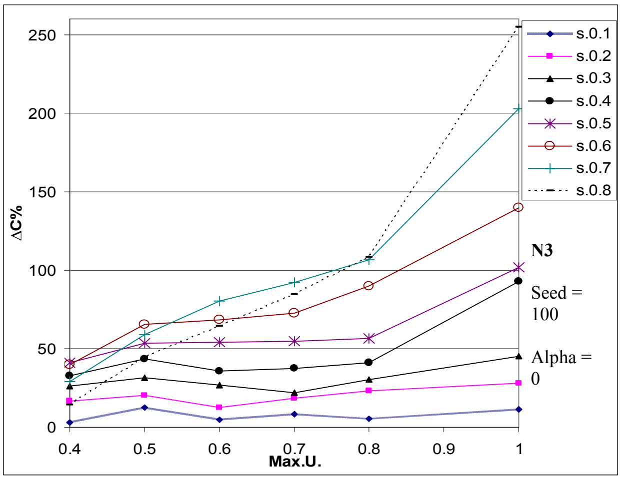

Fig.3. ΔC of networks with Alpha = 0 for N3.

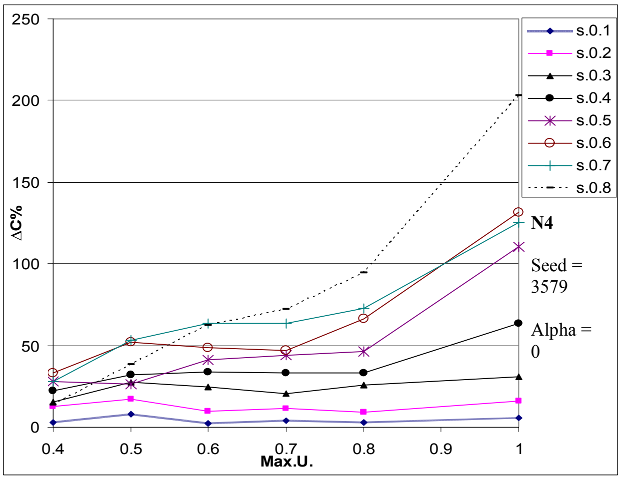

Fig.4. ΔC of networks with Alpha = 0 for N4.

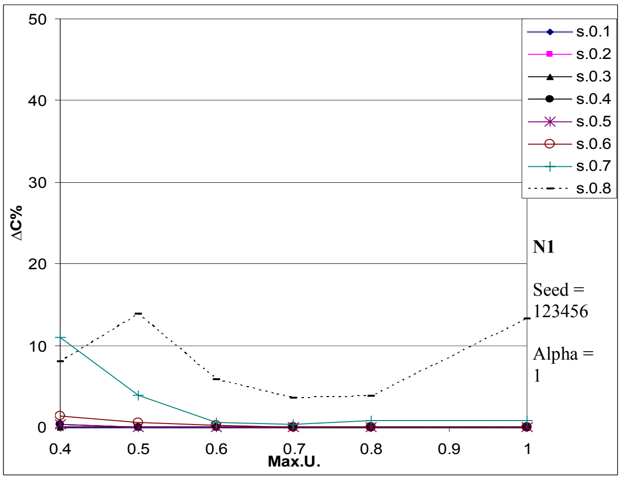

Fig.5. ΔC of networks with Alpha = 1 for N1.

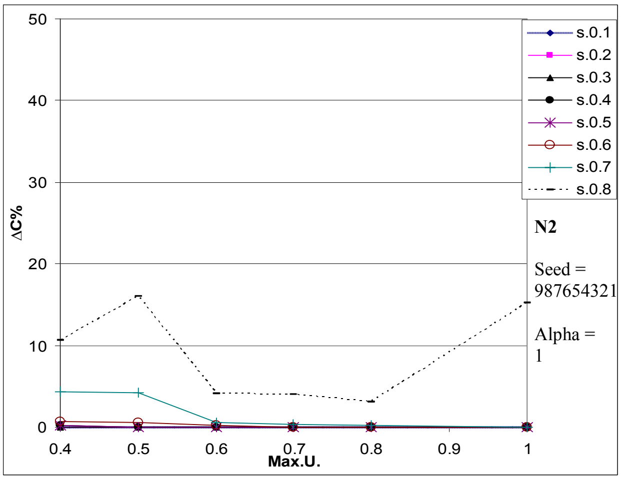

Fig.6. ΔC of networks with Alpha = 1 for N2.

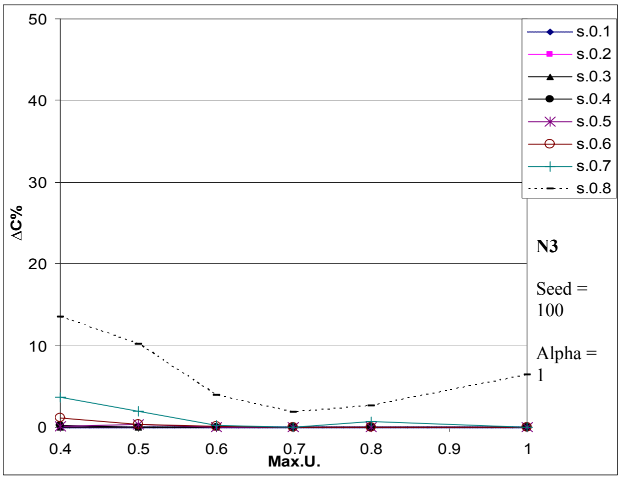

Fig.7. ΔC of networks with Alpha = 1 for N3.

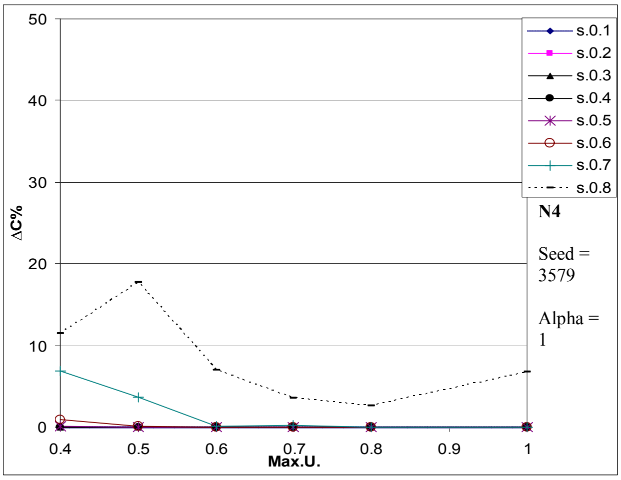

Fig.8. ΔC of networks with Alpha = 1 for N4.

Proceedings of the 10th WSEAS International Conference on COMMUNICATIONS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp419-424)

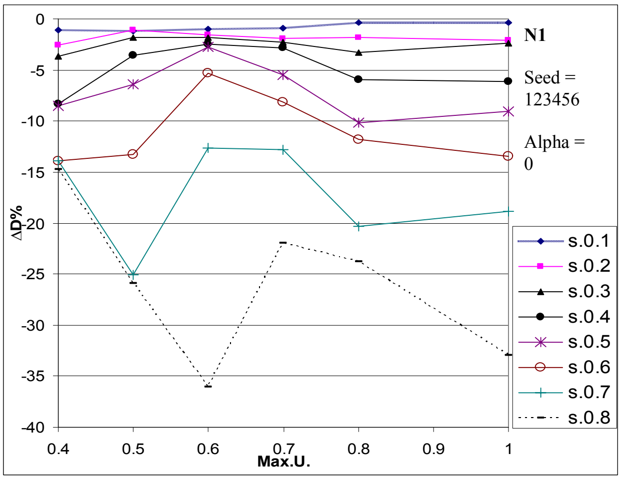

Fig.9. ΔD of networks with Alpha = 0 for N1.

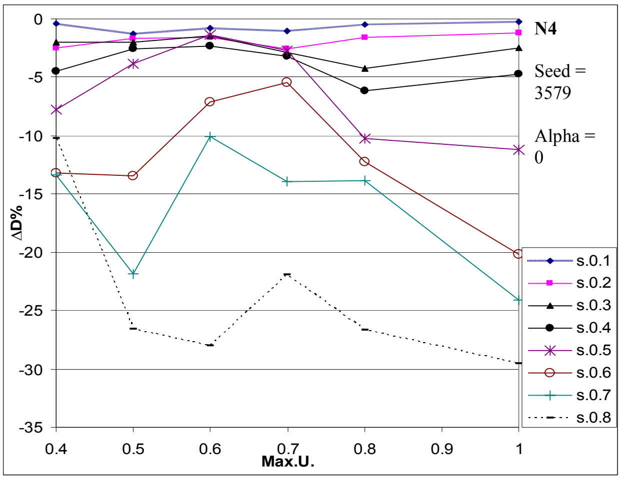

Fig.12. ΔD of networks with Alpha = 0 for N4.

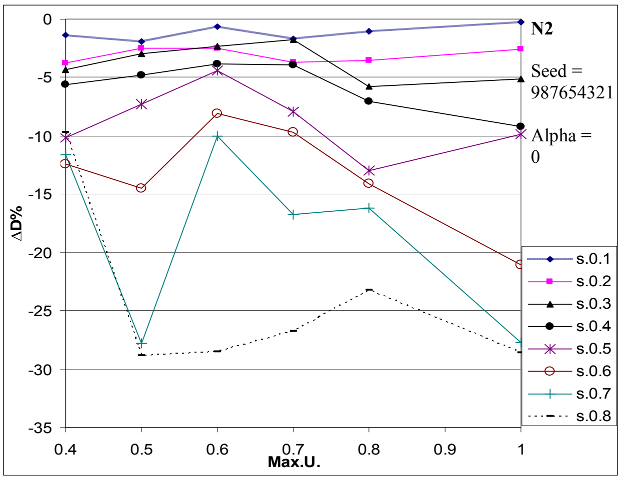

Fig.10. ΔD of networks with Alpha = 0 for N2.

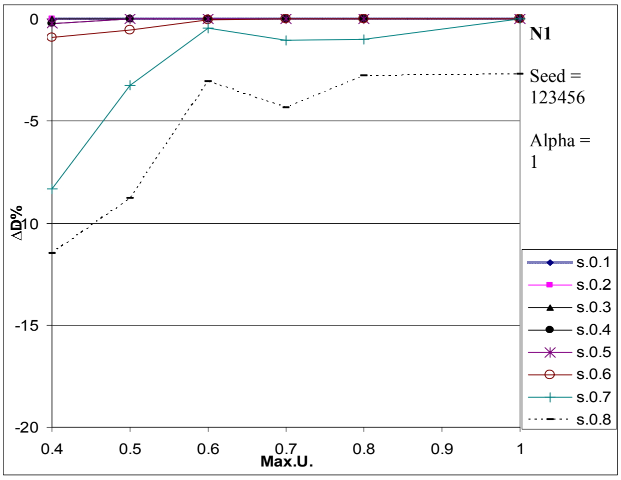

Fig.13. ΔD of networks with Alpha = 1 for N1.

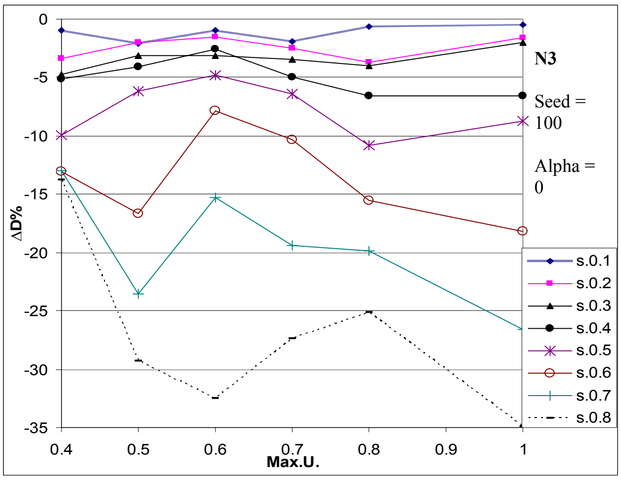

Fig.11. ΔD of networks with Alpha = 0 for N3.

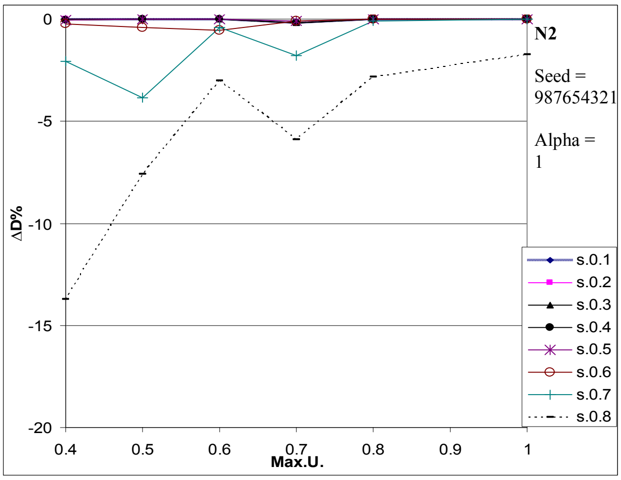

Fig.14. ΔD of networks with Alpha = 1 for N2.

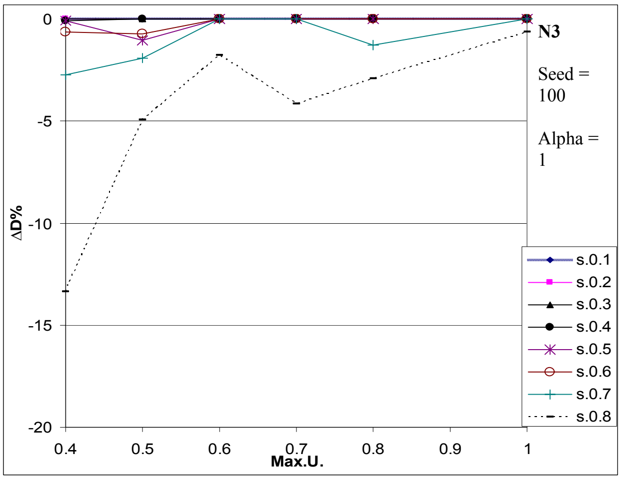

Fig.15. ΔD of networks with Alpha = 1 for N3.

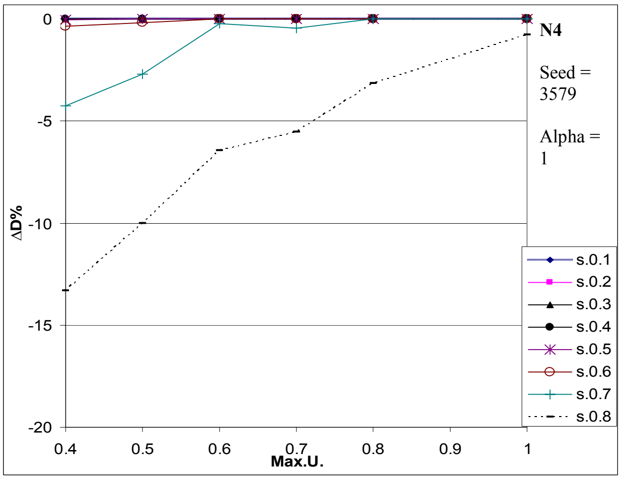

Fig.16. ΔD of networks with Alpha = 1 for N4.

3.3 Experiment Analysis

Fig.1-4 present ΔC of MENTOR networks with α=0 designed for N1-N4, respectively. It is clear that ΔC is more depend on s than ρ. For small s, s=0.1, ΔC is small with average 5.5244%, and hardly change with ρ. As s increase, ΔC get worse and more depend on ρ, ΔC achieve maximum of 250% at s=0.8 and ρ=1.

Fig.5-8 present ΔC of MENTOR networks with α=1 designed for N1- N4, respectively. For small s, s=0.1, ΔC is very small with average 0.0037%, and also hardly change with ρ. ΔC get worse as s increase and has no obvious relation with ρ.

Fig.9-12 present ΔD of MENTOR networks with α=0 designed for N1-N4, respectively. For small s, s=0.1, ΔD is small with average 0.9629%, and also hardly change with ρ. As s increase, ΔD get worse and more depend on ρ, ΔD achieve maximum of ~35% at s=0.8 and ρ=1

Fig.13-16 present ΔD of MENTOR networks with α=1 designed for N1- N4, respectively. For small s, s=0.1, ΔD is very small with average 0.0077%, and hardly change with ρ. As s increase, ΔD get worse as s increase and has no obvious relation with ρ.

4 Conclusion

In this study, the relations between design parameters and performance of traffic assignment of MENTOR algorithm have been explored. 384 MENTOR networks have been analyzed. Our results indicate that the performances of MENTOR strongly depend on a parameter called "slack". For small value of slack, e.g. s=0.1, the performances at normal load and at congestion threshold of MENTOR keep very close to that of the optimum solution, while the maximum utilization hardly impact the performances. As the slack get larger the performances of MENTOR become worse and more depend on the maximum utilization. Note that as slack get large, MENTOR tends to install more links; hence, the obtained networks have more node degree. In conclusion, the performance of MENTOR network gets worse as number of links and node degree get large as well.

References:

[1] Aaron Kershenbaum, Parviz Kermani, and George A. Grover, MENTOR: An Algorithm for Mesh Network Topological Optimization and Routing, *IEEE Transaction on Communications,* Vol.39, April 1991, pp.503-513.

[2] Robert S. Cahn, *Wide Area Network Design.* Morgan Kaufmann Publisher, San Francisco, CA, 1998.

[3] B. Fortz and M. Thorup, Internet traffic engineering by optimizing OSPF weights, *in Proc. IEEE INFOCOM, vol. 2*, Mar. 2000, pp.519–528.

[4] B. Fortz, J. Rexford, and M. Thorup, Traffic engineering with traditional IP routing protocols, IEEE Commun. Mag., vol. 40, Oct. 2002, pp. 118–124.

[5] A. Sridharan, R. Gu´erin, and C. Diot, Achieving Near-Optimal Traffic Engineering Solutions for Current OSPF/IS-IS Networks, *In Proceedings of IEEE INFOCOM, San Francisco, USA*, April 2003.

[6] Robert S. Cahn, *The Design Tool: Delite (software)*, 1998, http://www.kt.agh.edu.pl/~mwagrow/projektowanie siecitelekomunikacyjnych/DeLite/setupex.exe

[7] Andrew Makhorin. *GLPK (GNU Linear Programming Kit)*, 2000, http://www.gnu.org/software/glpk/