Decision Support Systems 35 (2003) 335–351

Decision Support Systems

www.elsevier.com/locate/dsw

Tradeoff decisions in the design of a backbone computer network using visualization

Indranil Bose [a,*], Kemal Altinkemer [b], Alok Chaturvedi [b]

[a]Department of Decision and Information Sciences, Warrington College of Business Administration, University of Florida, 351 Stuzin Hall, P.O. Box 117169, Gainesville, FL 32611, USA

[b]Krannert School of Management, Purdue University, West Lafayette, IN 47907, USA

Accepted 1 April 2002

Abstract

Visualization provides a useful tool for analyzing large, complex data sets. In the design of backbone computer networks, rough-cut design decisions can gain from a visual analysis of the generated solution with respect to design parameters such as average message delay, delay cost, average message length and total network operating costs. In this paper, we show how two-dimensional and three-dimensional surface and glyph representations can be used for understanding the cost–delay tradeoffs involved in the network design problem, and an idea of the 'efficient frontier' where the user may choose to operate. It provides an opportunity to revisit relationships that exist between the different network design parameters as well as discover new ones.

Keywords: Computer network design; Mathematical programming; Visualization

1. Introduction

Most problems related to the design of computer networks are combinatorially explosive and involve decisions about a large number of problem parameters. Many of these problems have been shown to belong to the NP-complete class of problems. Since these problems cannot be solved to optimality, efficient heuristics are often suggested to solve these computationally intractable problems using the well-established methodology of 'divide and conquer'. The solutions produced by these heuristics are not the 'best' in general and sensitivity analysis has to be performed to test the robustness of the heuristics to system parameters. Traditional sensitivity analysis may not recognize the 'hidden' relationships that exist between the generated solution and the problem parameters. Jones [14] illustrated the usefulness of visualization for sensitivity analysis due to movement of the location of a city in case of the traveling salesman problem and showed that the solution generated by an approximation algorithm is less stable than that obtained using an optimization algorithm. This conclusion could only be reached by using visualization.

Visualization is a useful technique in the solution process for optimization problems. However, though

* Corresponding author. Tel.: +1-352-392-0648; fax: +1-352-392-5438.

E-mail addresses: bosei@ufl.edu (I. Bose), kemal@mgmt.purdue.edu (K. Altinkemer), alok@mgmt.purdue.edu (A. Chaturvedi).

0167-9236/02/$ - see front matter © 2002 Published by Elsevier Science B.V.
doi:10.1016/S0167-9236(02)00112-4

network design problems are conventionally large-scale optimization problems, researchers have not generally sought to use visualization in the solution procedure for these problems. In this paper, we argue that since visualization is a useful and valid technique that has been used in optimization problems, they can be of use in network design problems that are also combinatorial in nature. Specifically, we introduce the problem of tradeoff analysis involved in the design of backbone computer network and go on to show how visualization can prove to be a useful technique for this particular network design problem.

The organization of the paper is as follows. Section 2 provides a review of the use of visualization techniques for solving optimization problems. Section 3 provides a review of prior work in network visualization. The specific network design problem addressed in this research is described in Section 4. Section 5 describes the steps involved in data generation and Section 6 details the various techniques used for generation of visuals. The visualizations obtained for tradeoff analysis and interpretation of these visualizations appear in Section 7. Section 8 is the concluding section.

2. Use of visualization in optimization problems

The motivation for using visualization techniques comes from the literature that already exists on visual interactive optimization. In one of the seminal papers on interactive optimization, Fisher [9] explained the importance of bringing the human in the problem-solving loop to enhance the solution procedure. The human in the loop could contribute to the solution procedure in three ways—by facilitating model specification and revision, coping with aspects of the problem that are difficult to quantify and finally, assisting in the solution process. Optimization problems in general are visually oriented and in certain situations a visual representation of the problem enables a human expert to identify 'patterns' and come closer to the best possible solution [14].

Brady et al. [4] described an interactive optimization system that solved a facilities location problem. The algorithm drew circular regions on the screen, indicating prospective facility location sites, with the user having to identify a point in the intersection of the regions. The algorithm terminated when the region contained only a single point—the optimal location of the facility. Cullen et al. [5] described another interactive optimization system for solving the vehicle routing problem. In this system the users could specify new routes as the starting points. Pirkul et al. [18] described a human–machine cooperative approach for solving the capacitated P-median problem. Eick and Wells [8] gave a comprehensive survey of the core features of interactive graphics and investigated how familiar plots like histograms, quantile diagrams and scatterplots could be made interactive.

Another area of optimization that very naturally lends itself to visualization is animation of algorithms. The main idea behind algorithm animation is, as the algorithm executes, a picture of the current state of the algorithm is updated when 'interesting' events occur. Lustig [16] used a three-dimensional polytope of either a three variable linear programming problem or a projection of a large problem in three dimensions. Each pivot of the algorithm was illustrated as a movement from one vertex to another along the polytope. Gay [11] provided an algorithm animation of an interior point method and showed how the algorithm iteratively distorted the polytope.

Visualization can also be used in interactive animation—where the user interactively modifies the path of execution of the algorithm using his/her domain knowledge. This can be done by modification of the data or by modification of the underlying model. Visualization of algorithms and complex solution procedures can also lead to new problems and new theory. Gleick [12] visualized the behavior of Newton's method for finding the roots of nonlinear functions. When each starting point was color-coded, based on the final solution that was produced, the image that resulted was a fractal. This example illustrated the power of visualization in providing insights into the behavior of even well-established algorithms.

Although interactive optimization has been widely used, the usefulness of such a procedure is still debatable. Stasko [20], to complicate the matter, showed that algorithm animation did not provide any significant aid to understanding the behavior of algorithms. Nonetheless, the wide use of the technique suggests that it does have some useful applications. As stated by Jones [14], "the challenge is to discover exactly how it is best delivered, to which audiences, and for what tasks". However, there is no denying the fact that visualization has proved to be an important technique in the solution of large-scale optimization problems.

3. Prior work in network visualization

Shaw and Yadav [19] provided a detailed review of network design methods and proposed a new network design methodology. However, it is interesting to note from that paper, though network design problems are often solved using optimization techniques and though visualization has proved to be a useful tool for solving optimization problems, yet the use of visualization for solving network design problems have not attracted the attention of researchers. We now provide a brief review of research that has used visualization in network design. In one of the earliest works in this area, Jack et al. [13] described a PC-based interactive computer-aided network planning model called NETCAP, which helped the user make a decision about how to expand the capacity of the existing network to meet the future requirements for voice service. This system is currently in use in all network operating departments at the GTE telephone operating companies.

Liu and Hockney [15] provided an example of visualization of network topology optimization. This paper described XNET as an X Windows-based graphical toolkit that took full advantage of human expertise in the context of an automated design process of backbone and local access computer networks. In this toolkit, visualization was used to satisfy three different goals—to convey information about relationships, to understand the global nature of the numerical solution and to allow the user dynamic steering of the computation according to her requirements. To avoid clutter, XNET used the idea of layered and pop-up canvas. Another example of an interactive network visualization toolkit is the NETPAD system developed by Dean et al. [6]. This system is specially suited for network design scenarios where the algorithmic needs change over time to meet users' needs for additional performance or functionality.

Eick and Wills [7] investigated the role of visualization in the navigation of networks with hierarchies. The paper described HIERNET—a toolkit developed for studying the nature of e-mail communication at AT&T. This study was important in a corporate environment as it helped in designing interfaces and conference tools, allocating people to tasks and investigating how people interacted within a given domain.

In one of the recent works on network visualization, Becker et al. [3] described a systematic procedure for visualizing the plethora of data generated by large networks. The paper described three static displays (link maps, matrix layout and spatial layout) for overcoming the problem of clutter in representing data. The paper concluded that different representations might be useful for visualization of large hierarchical networks, depending on the user and the purpose of use. Martin [17] provided visualization of network data that reside in large relational databases. The paper described tools that allowed the users to create static and animated visualizations of network data instead of writing and executing database queries. These tools were able to highlight the temporal content, geographical content, meaningfulness and hidden relationships of the voluminous data.

4. Problem statement

Among the many parameters involved in the design of backbone networks, message length (bits), average message delay (ms) and unit delay cost ($) are some of the more significant ones. The unit delay cost represents the opportunity cost of the user. The delay cost arises due to the waiting time of the computer users in front of their terminals. Unit delay cost represents the amount of money that the users of the network are willing to pay for delivery of their messages. Of course, it follows that a user with higher delay cost is more sensitive to network delay than a user with low delay costs.

The total cost of network design consists of three components—the fixed cost, the variable cost and the queuing cost. The fixed cost can be broken into two components—a fixed setup cost and a term proportional to the length of the link. The variable cost is proportional to the message rate and represents the cost per message unit charged by a common carrier. The queuing cost is the cost associated with the average delay of the messages being transmitted over the network. This cost is insignificant in the case of a single user. However, when thousands of users make use of the network, the message transmission rate is slow and delays are incurred. In some sense, the queuing cost thus represents the opportunity cost of the users in using the network. The queuing cost (Q) experienced by the user is related to the unit delay cost (D) such that if d is the delay experienced by a message on a link of the network and if L is the index set of links for the network, then the queuing cost experienced by the user is given by $Q = \sum_{l \in L} Dd$.

Gavish and Altinkemer [10] studied the problem of assigning capacities to the links and deciding on routes for messages for each origin–destination communication pair in a backbone computer network. The topology of the network was given and the end-to-end traffic requirements were known. The optimization procedure sought to find a least cost design for the network. An interesting aspect of this research was the study of the tradeoff between the different costs, namely the fixed cost, the variable cost and the queuing cost.

Another important tradeoff in network design is that between the total cost of network design and the average network delay. Fox example, a communication link with a message rate of 300 bits/s will have a low connection cost but high average message delay, whereas a T3 link with a message rate of 45 Mbits/s will have a high connection cost but low average message delay. To elucidate the cost–delay tradeoff, Balakrishnan and Altinkemer [2] developed an 'efficient frontier' between the number of hops, used as a surrogate for delay, and the network cost.

The focus of this paper is to show how data visualization can be used to study the tradeoff between the different cost components, average message delay, total network operating costs and various combinations of message length, link capacities and unit delay cost in case of the ARPA network. In some sense, this paper is an extension of the two-dimensional relationship between number of hops and cost of network design that was introduced in Balakrishnan and Altinkemer [2].

5. Data generation

Gavish and Altinkemer [10] used a Lagrangean relaxation-based method for determining the least cost network design in terms of messages and capacities of links. The mixed integer programming formulation is given in Appendix A. They tested their algorithm on four different networks—namely ARPA, OCT, USA and RING. In this paper ARPA is taken as the example network for the study of the cost–delay tradeoff analysis. Fig. 1 shows the ARPA network.

The Lagrangean relaxation-based solution technique, as detailed in Ref. [10], is used for generating the data points. For the link capacities seven choices are allowed. The choice of capacities and the corresponding setup cost, distance cost and variable cost are the same as in Ref. [10]. A route generation procedure is used as an integral part of the Lagrangean procedure that generates routes as needed during the solution process. The experiments are run for five different message lengths—200, 300, 400, 500 and 600 bits, respectively. The unit delay cost is varied as 100, 500, 1000, 2000 and 3000 (in dollars). These lead to 25 possible combinations. However, these are for a particular value of the fixed cost multiplier. The fixed cost multiplier is then varied in three steps—0.5, 1.5 and 3.0. For the above experiments the variable cost multiplier is kept fixed at 1.0. For each choice of fixed cost multiplier, the solution procedure generates the total cost of network design, components of the total

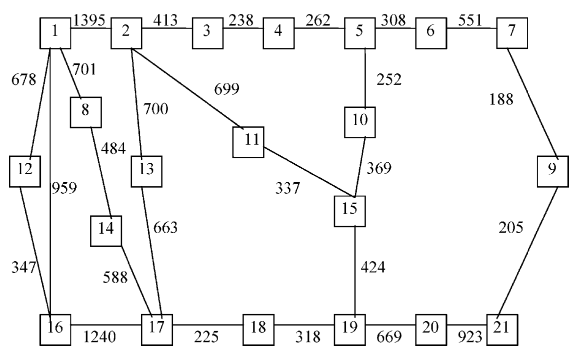

Fig. 1. ARPA network topology and the link distances.

cost (namely fixed cost, variable cost and queuing cost) and the average message delay. The solution procedure also gives the gap between the Lagrangean value and the feasible solution. As a second part to the experimental procedure, the same set of experiments is repeated, keeping the fixed cost multiplier fixed at 1.0 and varying the variable cost multiplier in three steps—0.5, 1.5, 3.0. The main idea behind changing the fixed and the variable cost multipliers is to find out how the total cost and the average message delay changes with change in the fixed cost or the variable cost. The visuals, based on the results of these experiments, are obtained using IBM's visualization software—Data Explorer.

6. Visual techniques

Several visual techniques are available for data representation. Some of the techniques that are available in IBM's Data Explorer are color map, surface plot, color glyph, isosurface, contour line, volume rendering, sequencer slice, streamline and streakline. Bajaj et al. [1] provided a detailed discussion on the relative advantages and disadvantages of the various visual techniques. Not all visual techniques are suitable for representation of every type of data and the visual analyst has to decide what visual representation would be the most appropriate for the problem under consideration.

For the research in question, three visual techniques are chosen for creating the visuals. The first technique used is a color map. A color map represents a relationship between a range of data values and a set of colors. In this case, spectral colors are used in the color map where 'red' signifies a 'high' value of the variable and 'blue' signifies a 'low' value of the variable. Color maps are useful if the user is interested in the values of the fourth variable on the hull of the surface formed by three variables. The second technique used is a surface plot. A surface plot is a surface that connects all points in a three-dimensional space. Data Explorer uses the technique of triangulation for interpolation in between the given data points for construction of the surface plot. Surface plot is useful when it is possible to conceive of a surface that joins all data points. Surface plots do not convey much information in case of clustered data. The third technique used is a color glyph. A glyph is a visual object. Glyphs are made by copying a generic object, positioning them appropriately and coloring them according to the data associated with that sample point. Glyphs are useful for discrete data and can represent a number of attributes through their shape, size and color. However, glyphs can result in information clutter if the data points are not spaced out.

The visuals generated using the results of the numerical experiments are displayed in either 'off-diagonal' or the 'front' view. For any visual that particular view is chosen which revealed the most information. Data Explorer provides 14 different views and also provides capabilities for rotation, zooming, panning, roaming, navigation and animation of the generated visuals.

7. Visualization of design parameters

7.1. Relationship between total cost, message length, message delay and delay cost in 3D

In the first set of visuals the relationships between total cost and the different problem parameters are studied. One of the interesting design questions is to see how the total cost changes when the fixed cost and the variable cost are changed. To answer this question, the fixed cost is varied by using fixed cost multiplier values 0.5, 1.5 and 3.0. An increase in the fixed cost may be interpreted as investment in new technology that helps reduce the operating costs. Fig. 2(a)–(c) represents the total cost as a colored surface (colored according to its magnitude, with blue representing low total cost and red representing high total cost) against three problem parameters—message length, average message delay and delay cost.

The following observations can be made about the visuals:

- The total cost increases with increase in the message length irrespective of the multiplier for fixed cost;
- The range of average message delay (for the same message length combinations) increases as the multiplier for the fixed cost increases;

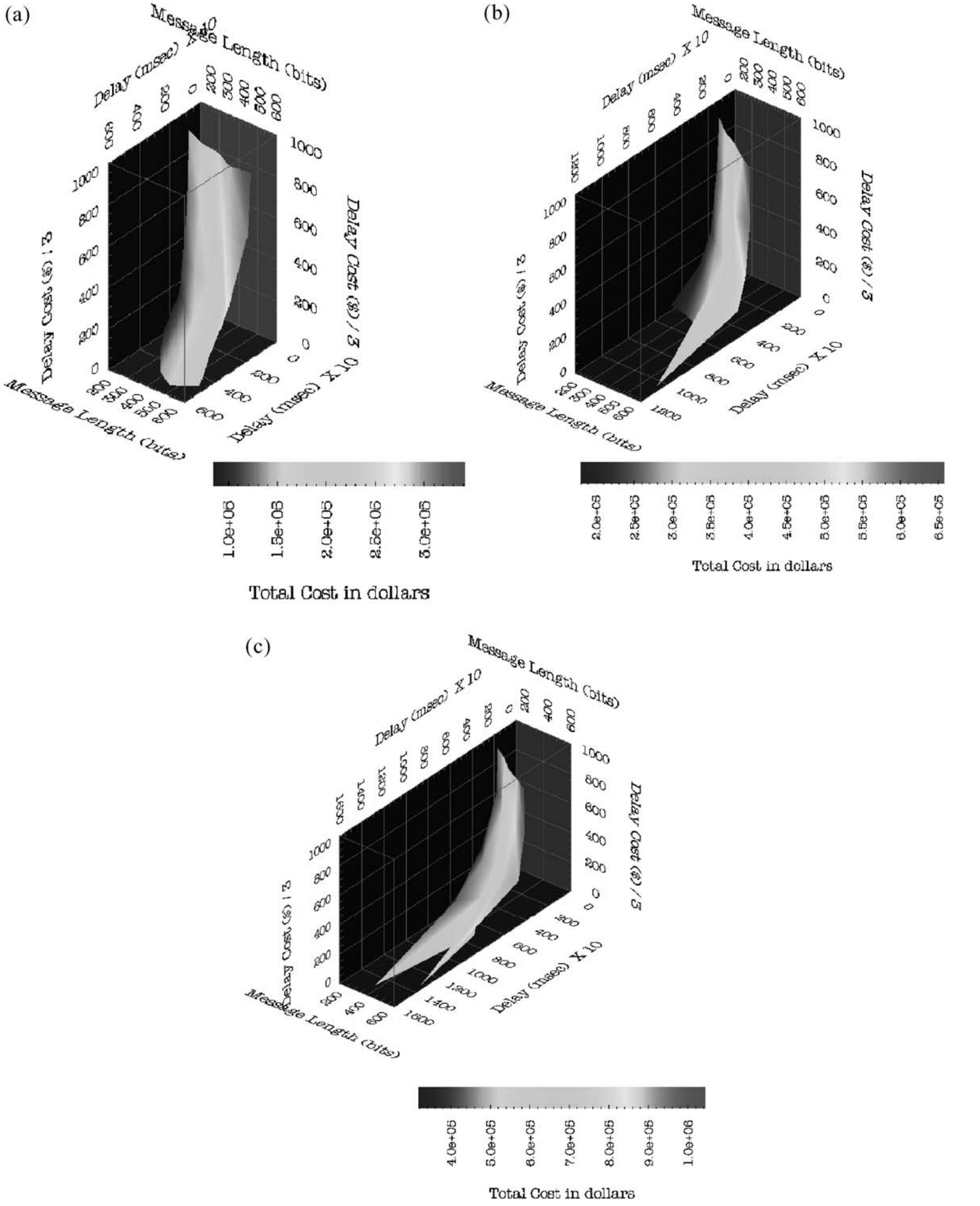

Fig. 2. Variation of total cost with average message length, average delay and delay cost when fixed cost multiplier is: (a) 0.5, (b) 1.5 and (c) 3.0.

- Higher delay cost leads to higher total cost for all possible fixed cost multipliers;
- Higher total cost does not imply lower average message delay. The effect of the message length seems to be more dominant.

The next set of visuals shows the variation of total cost of network design (fixed + variable + queuing) as a function of average message delay, delay cost and message length. In these 3-D representations (Fig. 3(a)–(c)), the variable cost multiplier is 0.5, 1.0 and 3.0, respectively. The surface map represents the total cost.

Observations about the visuals:

- The total cost increases with increase in the message length for all possible multipliers of the variable cost;
- When the variable cost multiplier is very high (i.e. 3) the range of average delay for different message lengths becomes small;
- The total cost increases with increase in the delay cost.

7.2. Relationship between total cost, message delay and delay cost in 2D

The next set of visuals shows the relationship between average message delay, delay cost and total cost. The total cost is represented as a surface plot. Figure sets 4(a)–(c) are for three different fixed cost multipliers—0.5, 1.5 and 3.0, respectively.

The visuals indicate:

- Higher delay cost leads to lower average message delay and vice versa. This trend is satisfied for all three values of the fixed cost multiplier;
- The range of average delay for the same message length and unit delay cost combinations increases with increase in the fixed cost multiplier;
- In general, the total cost of design is maximum for the medium delay range and minimum for the lower delay range.
- For lower delay cost and a large change in delay, the change in total cost is comparable to the change in the total cost for higher delay cost and a small change in delay. This is evident from the shape of the visual obtained in all the three cases.

The next set of visuals (Fig. 5(a)–(c)) shows the relationship between delay, delay cost and total cost. The three images are for three different variable cost multipliers (i.e. 0.5, 1.5 and 3.0, respectively). The colored surface plot represents the total network operating cost.

Observations about the visuals:

- Higher delay cost leads to lower delay and vice versa;
- The total cost is highest for medium delay range;
- The change in the total cost is high for a small change in average message delay at high delay cost.

7.3. Relationship between fixed, variable and queuing costs and message length in 3D

The next three visuals (Fig. 6(a)–(c)) show the variation of total cost of network design and average message delay as a function of the three different cost components. In these 3-D surface representations, the fixed cost multiplier is 0.5, 1.0 and 3.0, respectively. The total cost is represented by the color-coded surface map, whereas the average message delay is represented by the colored glyphs.

Observations about the visuals:

- The total cost is higher for higher fixed cost;
- Higher total cost does not imply lower average message delay and vice versa. However, for high total cost the average message delay is usually low;
- Lower queuing cost leads to higher delay and vice versa. This is so because the queuing cost represents the opportunity cost that the user incurs due to time lost waiting for delivery of messages over the computer network. Lower queuing/opportunity cost implies that the user will possibly experience higher delays as the user is not extremely sensitive to the delay.

The following visuals illustrate the tradeoff between total cost and delay with respect to the three cost

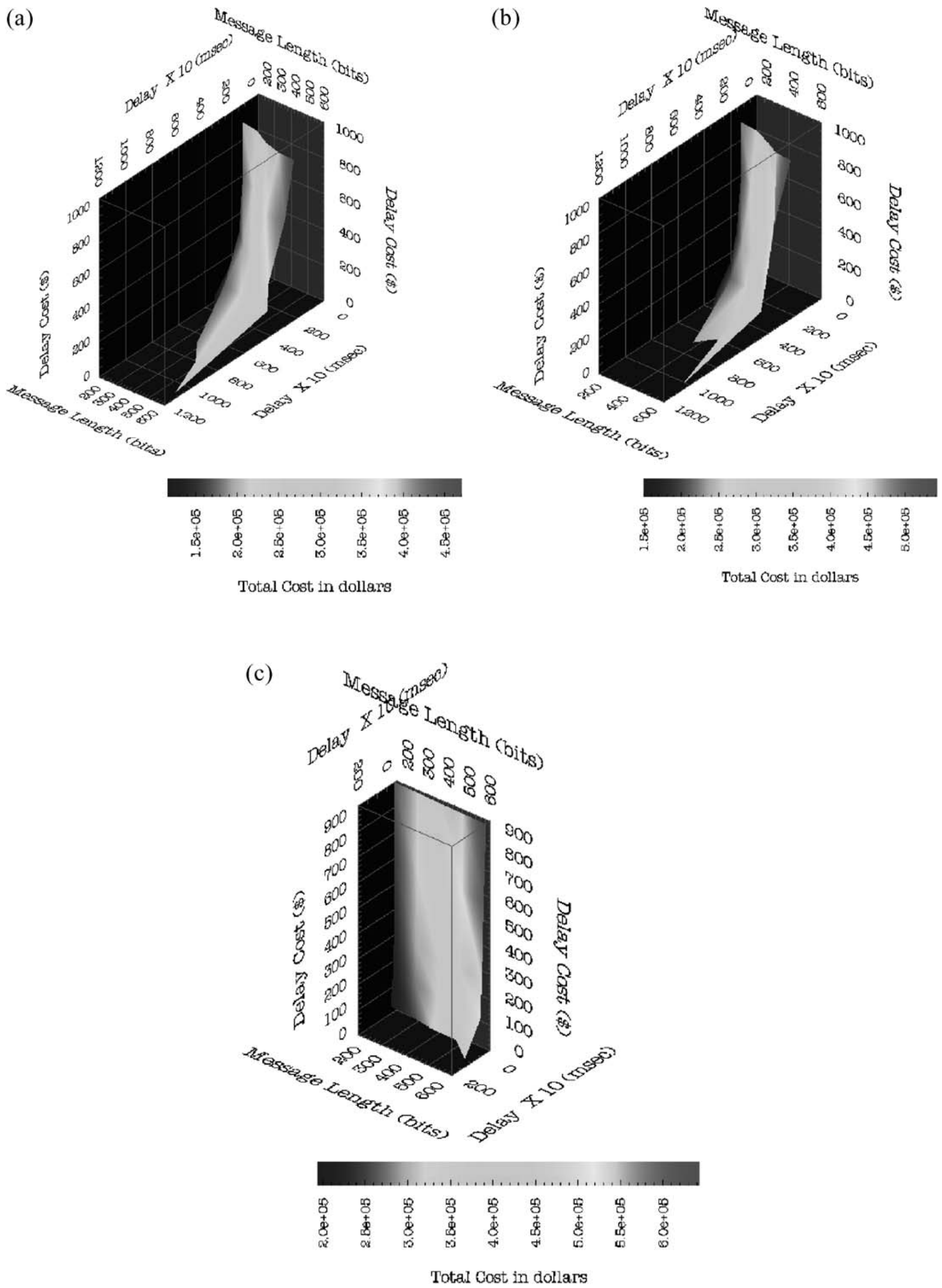

Fig. 3. Variation of total cost with average message length, average delay and delay cost when variable cost multiplier is: (a) 0.5, (b) 1.5 and (c) 3.0.

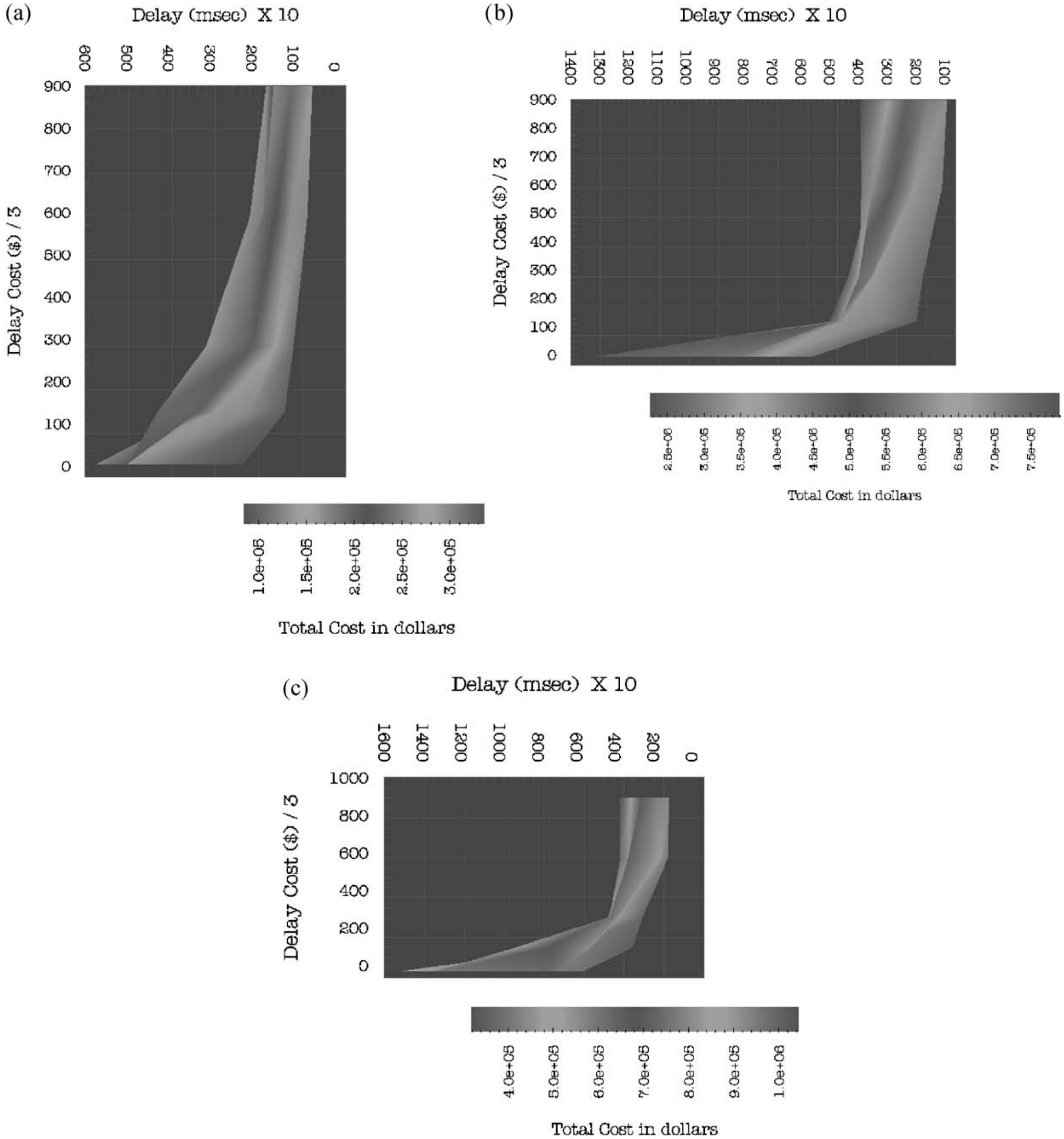

Fig. 4. Variation of total cost with average delay and delay cost in 2D when fixed cost multiplier is: (a) 0.5, (b) 1.5 and (c) 3.0.

components (namely fixed cost, variable cost and queuing cost). In Fig. 7(a)–(c) the variable cost multiplier is 0.5, 1.0 and 3.0, respectively. The color-coded surface represents the total cost and the color-coded glyphs represent the average message delay.

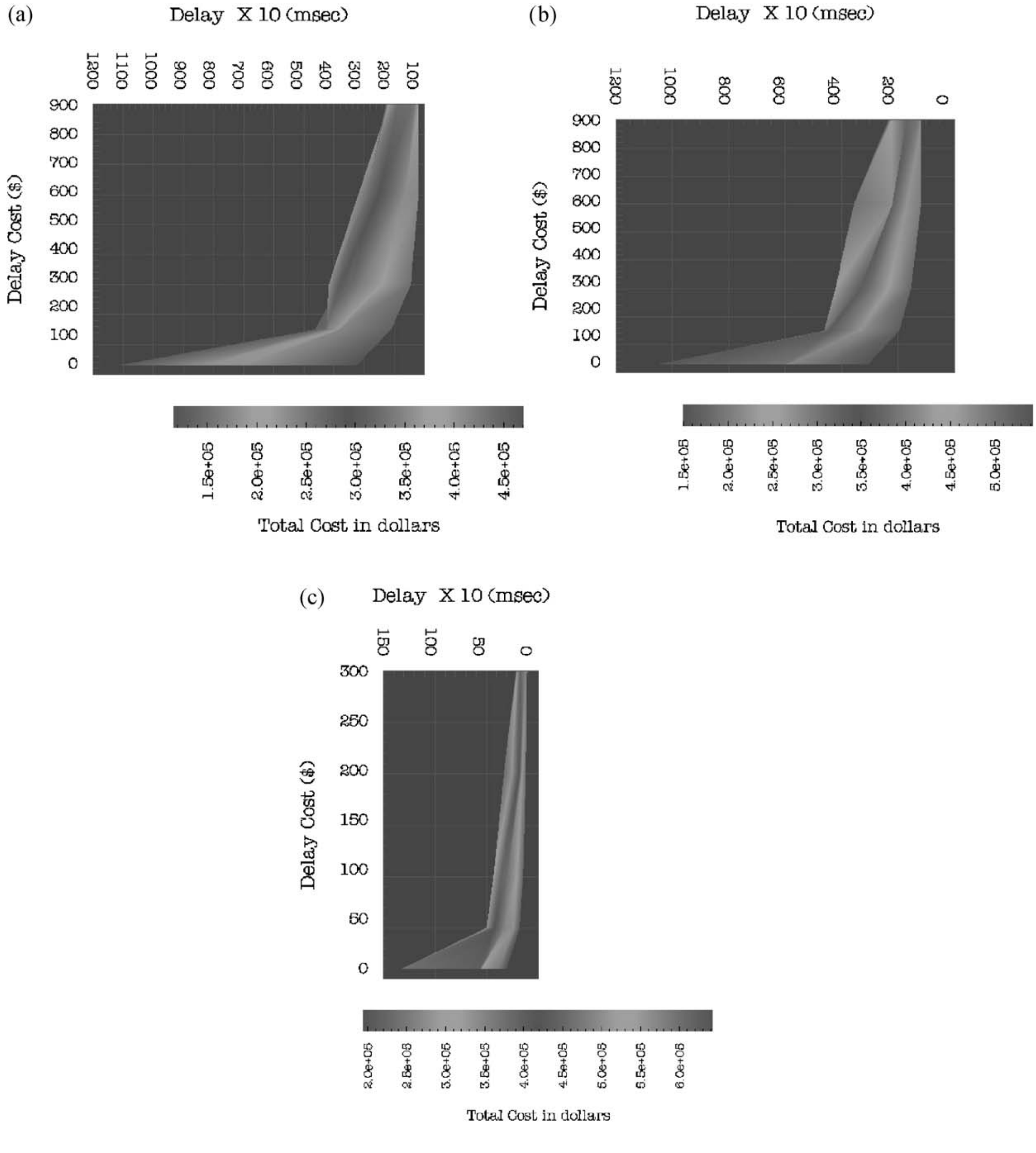

Fig. 5. Variation of total cost with average delay and delay cost in 2D when variable cost multiplier is: (a) 0.5, (b) 1.5 and (c) 3.0.

The visuals convey the following information:

- The total cost is higher for higher fixed cost irrespective of the increase in the variable cost;
- Higher total cost does not imply lower average message delay and vice versa;
- Low average message delay seems to be associated with high queuing cost.

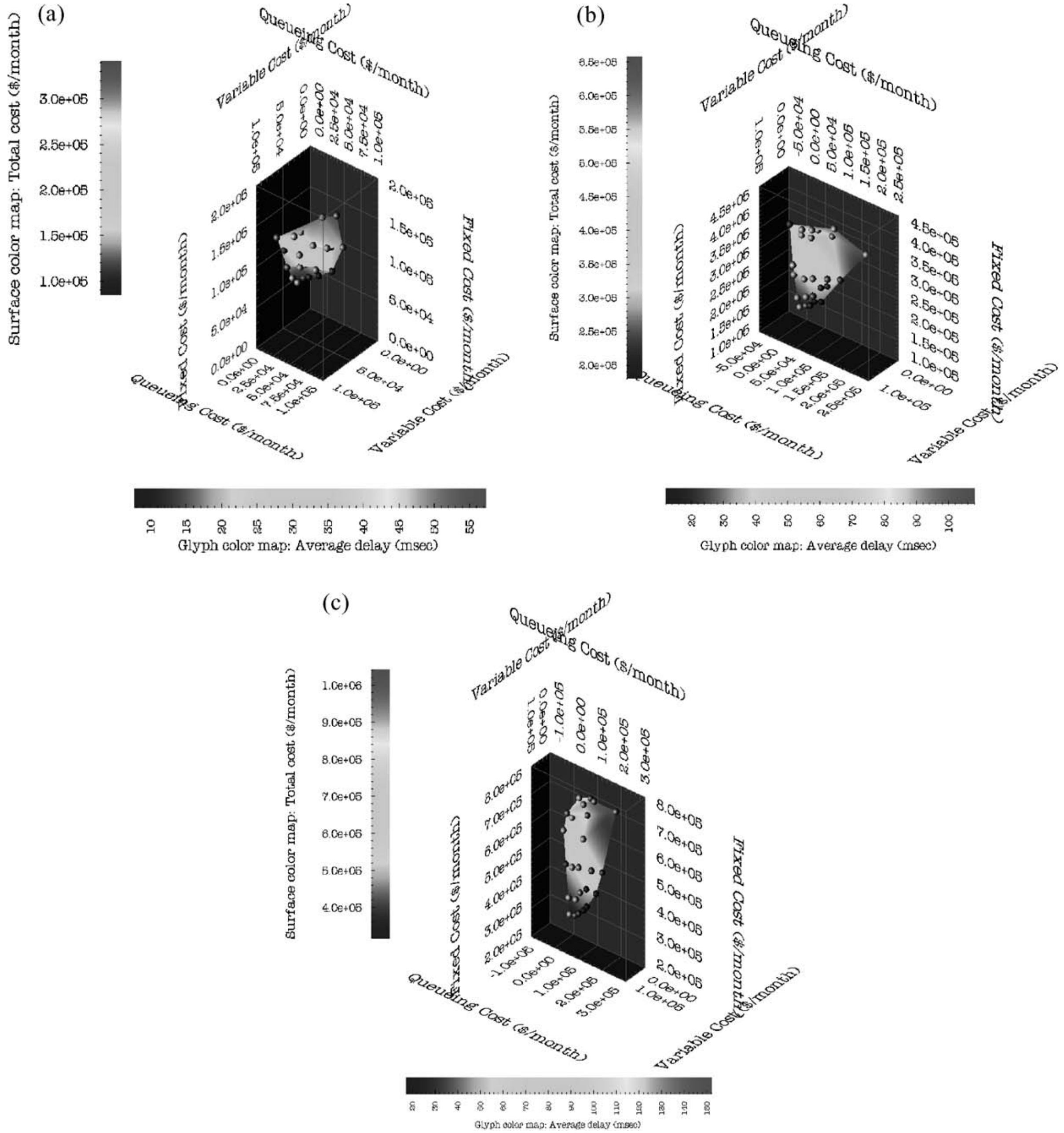

Fig. 6. Variation of total cost and its component costs (fixed, variable and queuing) with average delay when fixed cost multiplier is: (a) 0.5, (b) 1.5 and (c) 3.0.

7.4. Special cases

In this section two hypothetical scenarios are described. In the design of backbone networks the network provider has two options. The first option is to set up the lines by themselves and thereby incur fixed cost that includes a setup cost and a cost term proportional to the length of the links. The second

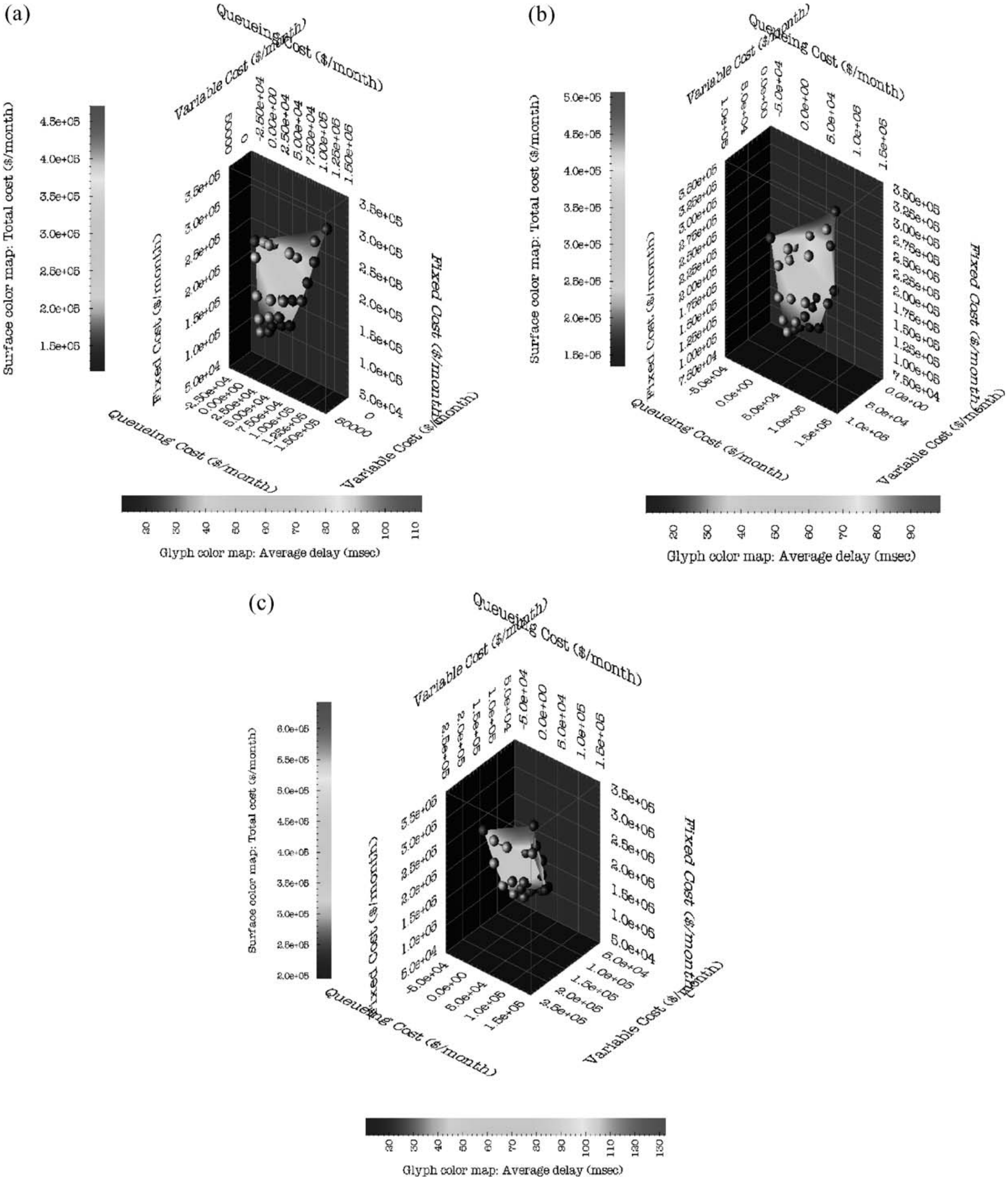

Fig. 7. Variation of total cost and its component costs (fixed, variable and queuing) with average delay when variable cost multiplier is: (a) 0.5, (b) 1.5 and (c) 3.0.

option is to lease the lines from a communication service provider. In this case no fixed cost is incurred but this results in variable cost.

Actual design situations are a combination of network service providers setting up new communication lines and leasing lines from common carriers and as such in all cases considered in the previous section, both fixed and variable costs are incurred. However, it may be interesting to see how the results change when these two special scenarios are considered. The variation of total cost of network design with message length, average message delay and delay cost when the fixed cost multiplier is 0.0 and the variable cost multiplier is 1.0 is shown in Fig. 8.

The visual shows:

- The total cost changes almost proportionally with the message length. The total cost remains low for small message lengths irrespective of the delay cost;

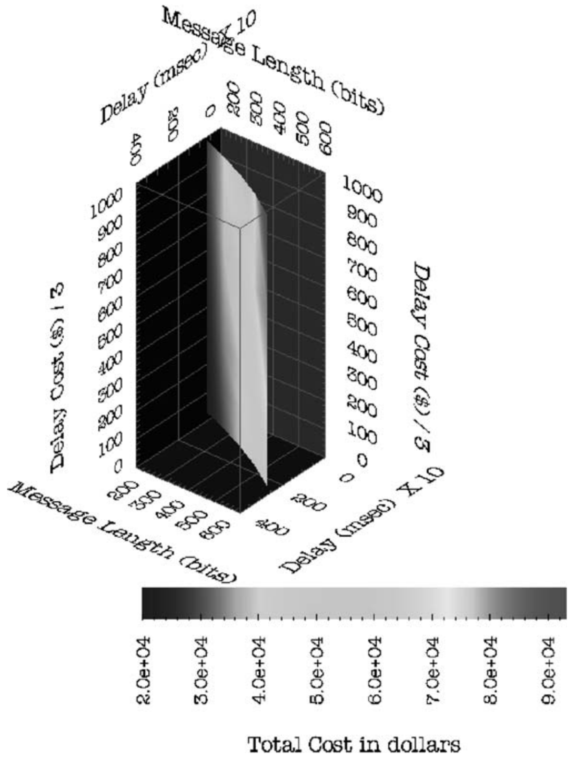

Fig. 8. Variation of total cost with average message length, average delay and delay cost when the fixed cost multiplier is 0.0.

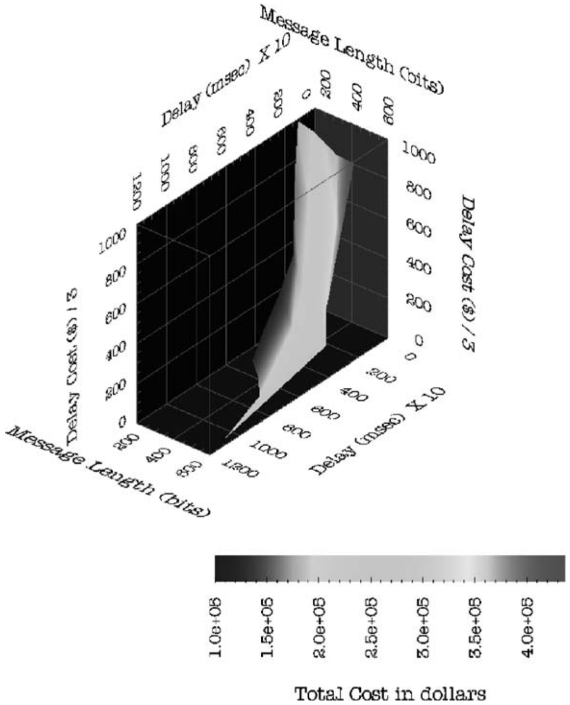

Fig. 9. Variation of total cost with average message length, average delay and delay cost when the variable cost multiplier is 0.0.

- The average message delay increases with increase in the message length;
- The total cost is higher for larger message lengths and higher delay cost than for larger message lengths and lower delay cost;
- Higher queuing cost does not imply lower average message delay, specially for larger message lengths.

The variation of total cost of network design with message length, average message delay and delay cost when the variable cost multiplier is 0.0 and the fixed cost multiplier is 1.0 is shown in Fig. 9.

The visual shows:

- The total cost is high for larger message lengths and vice versa;
- Higher delay costs lead to higher total cost for large and small message lengths whereas the total cost remains unchanged with the change in the delay cost for medium message lengths;

- For the same message length higher delay cost leads to lower average message delay.

A comparison of Figs. 8 and 9 shows that when the fixed cost multiplier is 0.0 (i.e. all lines are leased from common carriers), the average message delay is much lower and the total cost of network design is also much lower. This shows that fixed cost constitutes the majority of the total cost of network design and, as such, reduction in fixed cost can lead to tremendous reduction in the total cost of network design and the average delay of the messages.

Fig. 10 summarizes the main results obtained from the visualizations depicted in Figs. 2–9 in the case of the ARPA network. This diagram illustrates the influence of various factors on each other for the backbone network design problem. The main goal of this diagram is for decision support of the network designer. Visualization of the network design parameters helps in rough-cut analysis by indicating the positive and the negative influence of factors like average message length, average delay cost, total fixed cost, total queuing cost, multipliers for fixed cost and variable cost on total cost of design, average message delay and the range of average message delay. A network designer will be able to use this diagram to find out how to control certain factors when an increase/decrease in the total cost or the average message delay is desired. However, it is to be remembered that the diagram shown in Fig. 10 is for the ARPA network and should not be generalized for other example networks. For other networks, similar diagrams have to be constructed based on obtained visualizations for rough-cut influence analysis of the various factors involved in design of backbone networks.

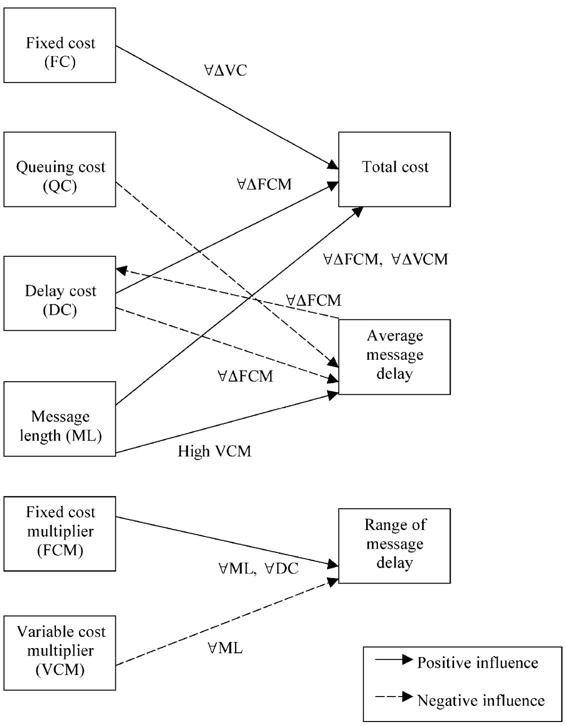

Fig. 10. Influence of network design parameters obtained from visualizations.

8. Conclusion

The main emphasis of this paper is to show how visualization can be used as a plausible technique to be integrated in the solution algorithm for backbone network design, to study the relationships that exist between the generated solutions and the problem parameters in a network design problem. The backbone network design problem is an NP-complete problem and hence 'good' solutions for such a problem are perhaps more important than 'optimal' solutions. Similarly, rough-cut ideas about relationships between parameters are perhaps more important than the exact relationships between parameters. We can look upon the results obtained for the ARPA network as a data point in the domain of research on decision support systems for design of backbone networks, which adds value to the designer of the ARPA network. Here, it is shown that several meaningful relationships exist between total cost, average message delay, message length and delay cost. It is interesting to note that in most cases, larger message length leads to higher total cost, higher delay cost leads to lower average message delay and higher total cost does not necessarily imply lower average message delay. These and other relationships, as shown in this paper, can be successfully used by a network designer to decide how to spend the allocated budget when a particular performance guarantee (in terms of delay of messages) is to be met. The designer has to keep in mind that the conclusions reached in this paper are a function of the input parameters and the specific algorithm used for solving this problem. However, it is to be remembered that the relationships obtained between parameters in this paper are valid for the ARPA network. Repeating the experiments will not guarantee that the same relationships will hold across all networks. But the main contribution of this study is to show that data visualization techniques can be useful in understanding these relationships, though the relationships may not be exactly same across all networks. In future, we would like to repeat the experiments for other networks as well. Of special interest will be the comparison of cost–delay tradeoffs between sparse and dense networks. Another area of future research can be the comparison of visualization with existing pattern recognition techniques (e.g. neural networks) to find out which technique leads to better discovery of underlying relationships between design parameters. It is to be remembered that when using visualization for network design problems, there is a fundamental assumption that the designer has sufficient domain knowledge about the problem under consideration. It is often difficult to determine the best visual technique to use for the study, understand the meaning of the 2-D and 3-D visuals and adjust several visualization-related parameters like degree of rotation, scale of axis, stand-off distance, among others, to uncover the most 'interesting' visualizations. A useful future research will be to study the impact of domain knowledge on the understandability and perceived usefulness of the visualizations for a given problem.

Appendix A

The backbone network design problem is to determine the link capacities and routes for transferring messages between origin–destination pairs while minimizing the total network operating costs. The mixed integer programming formulation of this problem is presented by Gavish and Altinkemer [10] and is reproduced here for the convenience of the reader in order to understand the various factors associated with the problem of backbone network design.

List of notations

L	index set of links in the network
I_l	index set of line types available for link l, $l\in L$
Q_{lk}	capacity of line type k, $k\in I_l$
C_{lk}	variable cost of line type k, $k\in I_l$ for one unit of traffic carried by it
D	unit delay cost
Π	set of communicating origin–destination pairs in the network
S_p	set of all possible routes for origin–destination pair p, $p\in\Pi$
R	set of all possible routes for all origin–destination pairs
λ_r	message arrival rate for the unique origin–destination pair associated with $r\in R$
λ_p	λ_r for all $r\in S_p$
δ_{rl}	$\begin{cases} 1 & \text{if link } l \text{ is included in route } r \\ 0 & \text{otherwise} \end{cases}$
$1/\mu$	average message length in the network
W_i	set of minimum number of links which disconnects the network into two sets of connected nodes A_i and B_i
C_i	set of communicating pairs with an origin node in A_i and a destination node in B_i or vice versa
Ω	index set of cuts which do not have any links in common
$\bar{L}$	index set of links which are not included in any cut

The decision variables used are:

$$x_r = \begin{cases} 1 & \text{if route } r \text{ is selected to carry the flow} \\ & \text{of its associated origin – destination pair} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{lk} = \begin{cases} 1 & \text{if link } l \text{ is selected to have a capacity} \\ & \text{of type } k \\ 0 & \text{otherwise} \end{cases}$$

Problem P

$$Z_p = \min\left\{\sum_{l\in L}\frac{Df_l}{1-f_l} + \sum_{l\in L,k\in I_l} S_{lk}y_{lk} + \sum_{l\in L,k\in I_l} C_{lk}Q_{lk}f_l y_{lk}\right\}$$

subject to:

$$\sum_{r \in R} \frac{\lambda_r \delta_{rl} x_r}{\mu} \leq f_l \sum_{k \in I_l} Q_{lk} y_{lk} \qquad \forall l \in L$$

$$0 \leq f_l \leq 1 \qquad \forall l \in L$$

$$\sum_{k \in I_l} y_{lk} = 1 \qquad \forall l \in L$$

$$\sum_{r \in S_p} x_r = 1 \qquad \forall p \in \Pi$$

$$\sum_{I \in W_l} \sum_{k \in I_l} Q_{lk} y_{lk} \geq \sum_{p \in C_i} \frac{\lambda_p}{\mu} \qquad i \in \Omega$$

$$x_r = 0, 1 \qquad \forall r \in R$$

$$y_{lk} = 0, 1 \qquad \forall k \in I_l \quad l \in L$$

Problem P can be shown to be NP-complete by fixing the values of y_{lk} and reducing it to the multi-constraint knapsack problem. To solve this problem, the technique of Lagrangean relaxation is used by dualizing a set of constraints to the objective function, breaking up the relaxed problem into solvable sub-problems, and finding the best nonpositive Lagrangean multipliers that yield the tightest lower bound for the problem. The technique of Lagrangean relaxation has the advantage that with the use of a simple heuristic the Lagrangean solution can be converted to a feasible solution and hence an upper bound on the solution to the problem. The quality of the solution is determined by the percentage gap between the upper and the lower bounds for the solution. The inputs to the problem include the origin–destination traffic matrix, the set of available line capacities, delay cost, the network topology and the Lagrangean procedure outputs the routes for each origin–destination pair, the link capacities and the total cost of backbone network design. For a more detailed description of the solution procedure the interested reader should refer to Gavish and Altinemer [10].

References

[1] C. Bajaj, A. Chaturvedi, S. Gupta, Business visualization: applications in marketing and economics. Working Paper, Krannert Graduate School of Management, Purdue University, West Lafayette, IN, 1998.

[2] A. Balakrishnan, K. Altinkemer, Using a hop-constrained model to generate alternative communication network design, ORSA Journal on Computing 4 (2) (1992) 192–205.

[3] R.A. Becker, S.G. Eick, A.R. Wilks, Visualizing network data, IEEE Transactions on Visualization and Computer Graphics 1 (1) (1994) 16–28.

[4] S.D. Brady, R.E. Rosenthal, D. Young, Interactive graphical minimax location of multiple facilities with general constraints, AIIE Transactions 15 (3) (1984) 242–254.

[5] F.H. Cullen, J.J. Jarvis, H.D. Ratliff, Set partitioning based heuristics for interactive routing, Networks 11 (1981) 125–143.

[6] N. Dean, M. Mevenkamp, C.L. Monma, NETPAD: an interactive graphics system for network modeling and optimization, Operations Research Letters 17 (1995) 89–101.

[7] S.G. Eick, G.J. Wills, Navigating large networks with hierarchies, Proceedings of the IEEE, 1993, pp. 204–210.

[8] S.G. Eick, G.J. Wills, High interaction graphics, European Journal of Operational Research 81 (1995) 445–459.

[9] M.L. Fisher, Interactive optimization, Annals of Operation Research 5 (1986) 541–546.

[10] B. Gavish, K. Altinkemer, Backbone network design tools with economic tradeoffs, ORSA Journal on Computing 2 (3) (1990) 236–252.

[11] D.M. Gay, Pictures of Karmarkar's Linear Programming Algorithm, Computer Science Technical Report No. 136, AT&T Bell Laboratories, Murray Hill, NJ, 1987.

[12] J. Gleick, Chaos: Making a New Science, Viking, New York, 1988.

[13] C. Jack, S. Kai, A. Shulman, Design and implementation of an interactive optimization system for telephone network planning, Operations Research 40 (1) (1992) 14–25.

[14] C.V. Jones, Visualization and optimization, ORSA Journal on Computing 6 (3) (1994) 221–250.

[15] H. Liu, D. Hockney, Visualization in network topology optimization, Proceedings of the 20th Annual ACM Computer Science Conference, Kansas City, MO, 1992, pp. 131–137.

[16] I. Lustig, Applications of interactive computer graphics to linear programming, Proceedings of the Conference on Impact of Recent Computer Advances in Operations Research, 1989, pp. 183–189.

[17] J.C. Martin, Visualization for telecommunications network planning, Proceedings of the IFIP TC5/WG5.2/WG5.10 CSI International Conference on Computer Graphics, Bombay, India, 1993, pp. 327–334.

[18] H. Pirkul, E. Rolland, R. Gupta, VisOpt: a visual interactive optimization tool for p-median problems, Decision Support Systems 26 (3) (1999) 209–233.

[19] N. Shaw, S. Yadav, DEACON: an integrated approach to the analysis and design of enterprise architecture-based computer networks, Communications of the AIS 7 (11).

[20] J. Stasko, A practical animation language for software devel-

opment, Proceedings of the IEEE International Conference on Computer Languages, IEEE Computer Society Press, Los Alamos, CA, 1990, pp. 1–10.

Indranil Bose is an Assistant Professor of Decision and Information Sciences at the Warrington College of Business Administration, University of Florida. His degrees include BTech (Electrical Engineering) from Indian Institute of Technology, MS (Electrical and Computer Engineering) from University of Iowa, MS (Industrial Engineering) and PhD (Management Information Systems) from Purdue University. He has research interests in telecommunications—design and policy issues, data mining and artificial intelligence, electronic commerce, applied operations research and supply chain management. His teaching interests are in telecommunications, database management, systems analysis and design, and data mining. His publications have appeared in Computers and Operations Research, Decision Support Systems, Ergonomics, European Journal of Operational Research, Information and Management and in the proceedings of numerous international and national conferences.

Kemal Altinkemer received his PhD in Computers and Information Systems from William E. Simon School of Business Administration, University of Rochester, Rochester, NY 14627 in 6/87. He is currently an Associate Professor and the area coordinator for MIS at the Krannert School of Management, Purdue University. His research interests are Infrastructure for E-commerce and pricing of information goods, bidding with intelligent software agents, strategy from Brickandmortar to Clickandmortar business model, design and analysis of local area networks, local access computer networks and backbone networks, infrastructure development such as ATM, LEOS systems such as TELEDESIC, distribution of priorities by using pricing as a tool, and time restricted priority routing. He has published numerous articles in journals such as Management Science, Operations Research, INFORMS Journal on Computing, EJOR and Transactions of the ACM.

Alok R. Chaturvedi is an Associate Professor of Management Information Systems and the Director of the SEAS Laboratory at the Krannert Graduate School of Management, Purdue University. He received his PhD in Management Information Systems and Computer Science from the University of Wisconsin-Milwaukee. He is an accomplished teacher and has published extensively in major journals and conference proceedings. He also has extensive consulting experience with companies such as AT&T, IBM, Ameritech, Bell Atlantic, Abbott Labs, Ernst and Young, the US Congress and the Department of Defense. He is widely cited as an expert on information technology strategies in the news media and magazines such as BusinessWeek, IndustryWeek and CIO Magazine.