THE DEVELOPMENT AND IMPLEMENTATION OF A DECISION SUPPORT SYSTEM FOR DESIGNING COMPUTER NETWORKS USING COMPROMISE PROGRAMMING

By

ANGELA EPPLER DIXON

Bachelor of Arts
The University of Tulsa
Tulsa, Oklahoma
1965

Master of Business Administration
The University of Tulsa
Tulsa, Oklahoma
1980

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 1989

THE DEVELOPMENT AND IMPLEMENTATION OF A

DECISION SUPPORT SYSTEM FOR DESIGNING

COMPUTER NETWORKS USING

COMPROMISE PROGRAMMING

Thesis Approved:

Ramesh Sharda
Thesis Advisor

J. Scott Turner

Marilyn G. Kletke

G. E. Hedrick

Norman N. Durham
Dean of the Graduate College

ACKNOWLEDGMENTS

This writer wishes to express her grateful appreciation to Dr. Ramesh Sharda for his interest, direction, wisdom and counsel during the conduct of this study and the writing of this thesis. The writer also wishes to thank Dr. J. Scott Turner, Dr. George E. Hedrick, and Dr. Marilyn G. Kletke, all of whom aided and helped as members of the Thesis Committee.

Additionally, the writer wishes to express her gratitude to Dr. Robert J. Monroe, Dean of the College of Business Administration at The University of Tulsa for his willingness to adjust her teaching responsibilities during the course of this time of additional learning and research. Further, the writer expresses her appreciation to Mr. James M. Payne, Dr. Robert A. Russell, and her other faculty colleagues at The University of Tulsa for their understanding and advice during these years of additional study.

The writer dedicates this thesis to her father, Mr. Roy Thomas Eppler, and to the loving memory of her mother, Mrs. Delphine Summers Eppler.

Finally, this writer expresses her grateful love to her family; her son James, her daughter Jennifer and her husband Jim, without whose undeviating devotion, constant encouragement and faithful cooperation, this study would not have been completed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure | Page

CHAPTER I

INTRODUCTION

General Statement of the Problem

The design of computer networks has been an active area of research for several years. The approaches have been varied and include both optimization and heuristic solution techniques for centralized and distributed computer networks. The recent explosion in the use of microcomputers and the networking of microcomputers to other micros, minicomputers and mainframe computers, plus, the integration of voice and data communications technology, provides further impetus for development of effective network design techniques.

Regardless of the algorithm(s) used in the design process, assumptions have to be made concerning network components and there will be trade-off considerations in meeting the multiple and often conflicting network objectives, such as maximizing service level and minimizing setup and operating costs. The missing ingredient in the algorithms presently available is a method of systematically including the decision maker(s) preferences in meeting these objectives. Compromise Programming (Zeleny, 1982) provides a tool for studying and understanding these trade-offs.

The purpose of this study is to develop a decision support system to allow a decision maker to interactively and systematically identify the trade-offs and the nature of the limitations imposed upon the network design due to these trade-offs. The objective of the system is to support the decision maker in the search for the best design solution.

The general flow of the system is (1) accepting data from the decision maker for generating the original objective functions and constraints for the decision problem and storing this data in a relational data base, (2) solving the problem as a single objective function problem by repeatedly weighting and summing the individual objective functions and using both a minimal spanning tree algorithm and an all pairs shortest path algorithm (3) determining for each solution generated from step 2 if the solution is nondominated (the same or better performance achieved with respect to all of the objectives, with at least one being strictly better) and if it is nondominated, adding it to the candidate solution list (4) calculating and graphically displaying the ideal (best possible) and the anti-ideal (worst possible) solution for each objective and the achievement level of the nondominated solution in terms of distance from the ideal 's level of meeting each of the objectives, and (5) accepting from the decision maker the preferred alternative to the current solution, identifying the nondominated solution that is closest to the decision maker's preference, and displaying this solution.

The decision support system (DSS) is based on a compromise programming model suggested by Milan Zeleny (1982) named Interactive Decision Evolution Aid (IDEA). The model assumes that (1) the decision maker's preference function is unknown and evolving throughout the decision process, (2) the set of alternatives can be specified through constraints or through a listing of specific requirements and that the ideal solution can be identified, (3) the decision maker prefers a nondominated solution to a dominated one and would accept the ideal if possible, (4) no weights of criteria importance are to be specified a priori, and (5) the decision maker is capable of determining acceptable and unacceptable solutions.

Objective of the Study

The objective of the study is to design, implement, and test a decision support system that applies the concepts of multiple criteria decision making (MCDM) to the design of computer networks. The decision support system (DSS) allows a decision maker(s) to interactively identify the appropriate communications components that would allow existing computers to communicate with each other and to share scarce data, software, and hardware resources in an optimal, cost-effective, and service-effective fashion.

It assumed that the decision-maker (DM) initially has a vague concept of the nature of the limits and trade-offs between a cost-effective and service-effective communicating network, but as solutions are presented, the decision-maker learns and is guided to what he/she considers to be the best compromise between the trade-offs.

The system incorporates a network topological design in terms of backbone analysis, local access network design, connectivity, and delay analysis as constraints in the minimal spanning tree and all pairs shortest path algorithms used in developing nondominated solution sets for application of the compromise programming model.

The interactive compromise programming approach is well suited for situations where the decision maker is actively involved in the decision analysis, but initially has a vague concept of the limits and trade-offs between conflicting multiple objectives. (Zeleny, 1982) This is frequently the situation in computer network design. Plus this method overcomes the inadequacies of hierarchical or utility ranking when the decision-maker is unable to initially rank the relative importance of the objective functions. (Benayoun, et al, 1971) The approach also does not require an initial feasible solution that is required in the Exchange Search Heuristic.

Two primary advantages of using the approach are the use of an IBM-PC-AT microcomputer-based system, rather than a mainframe computer, and the use of existing microcomputer software technology which has been interfaced to solve this particular problem. The software products include a relational database management system and a graphics generator subroutine toolkit. Both the hardware and the software are widely available and relatively inexpensive. The net result is a product that is easier and less time-consuming to use than manual methods. This may encourage the DM to investigate more alternatives for each network design and to better understand the ramifications of each alternative, and thus have more confidence in the final decision.

Scope and Limitations of the Study

This decision support system would be useful in the design of a particular type of computer network that includes several choices of accessible communications media and multiple design objectives.

The scope of the system was limited to identifying where to make the connections between hosts and what type of media to use in the connection links, while attempting to optimize the conflicting, multiple objectives and satisfice (Simon, 1960) these objectives and the designers preferences.

The system assumed (1) a store-and-forward, packet-switching communications subnet while assigning communications links, (2) one concentrator per host in the local access network, (3) the assignment of customer sites to concentrators and the terminal layout had been previously determined, (4) that reliability would be controlled through the use of packet-switching techniques that include error-detection and error-correction capabilities, (5) that throughput would be regulated through the packet-switching techniques, (6) that delay would be minimized through an acceptable upper bound for node to node response time, (7) that setup cost would be minimized through an upper bound for total setup cost, (8) that operating cost per time period be minimized through an upper bound for total operating costs per time period, (9) there will be one effective transfer rate between any two nodes for any given communications media.

CHAPTER II

RELATED STUDIES

Multiple Criteria Decision Making

This study applies the procedures of multiple criteria decision making to the design of computer networks. Since both areas are substantial, the review of the literature will treat the topics as separate and distinct, and then examine the research that connects them.

The field of MCDM evolved primarily from the fields of operations research, management science, decision sciences and systems analysis in response to the need for solution methodologies that allowed for multiple, and usually conflicting, criteria. Historically, these fields have been concerned with solving single aggregate criterion or unidimensional decision problems. Many researchers, however, perceive decision making as a multidimensional problem which involves more than one criterion and sought methods of solving this category of problems.

Multiple criteria decision making (MCDM) became an organized topic of inquiry as an outgrowth of a conference held in the early 1970s at the University of South Carolina. A volume of conference proceedings was published in 1973, "Multiple Criteria Decision Making" (Cochrane & Zeleny) that became the seminal source book for a decade.

When the second Multiple Criteria Decision Making conference was held in Washington D.C. in 1982, more than 3,000 references were included in the bibliographies. The topic had grown considerably in this decade as an area of study and research. Vincke (1986) reported a growth in the percentage of MCDM articles appearing in operations journals to have increased from approximately 3% in the early 1970's to almost 15% in the mid 1980s.

The fundamental intent of all of the multiobjective decision making is not to identify the optimal solution, which is generally infeasible, but to identify a complete or representative set of nondominated solutions. This approach is justified by the Kuhn-Tucker extension of their one-objective function necessary and sufficient conditions for optimality to a vector minimization problem. Their extension introduced the necessary and sufficient conditions for what they referred to as a "proper" solution. (Kuhn and Tucker, 1951)

The typical MCDM areas are multiobjective linear programming, multiattribute utility theory, goal programming, and compromise programming. The common MCDM applications are in economics, governmental studies, engineering, business, and management. Representative studies in the typical MCDM areas are summarized below.

Multiobjective Linear Programming

The intent of multiple objective linear programming is to locate all nondominated corner point solutions and from these to identify nondominated segments or faces of a feasible set.

Three of the most frequently applied MOLP procedures are weighted-sums, ϵ-constrained, and goal programming. The basis for the weighted-sums and ϵ-constrained methods is provided by Kuhn and Tucker's (1951) statement of the necessary conditions for a nondominated point.

Zadeh (1963) proposed the weighted-sum approach to generating the nondominated set. A scalar weight is applied to each of the objectives in the weighted-sums approach. The weighted objectives are then summed and the multiple objective problem is transformed into a single objective problem with a composite criterion. The decision maker (DM) can select the point estimate weights that will eventually lead to the efficient extreme point of his/her highest utility or the weights can be parametrically varied and applied to the objectives.

The acknowledged difficulty with the DM attempting to quantify the relative importance of the different objectives is that the contents of the feasible region are usually unknown to the DM. Zeleny (1982) pointed out that the weighting vectors also is a function of the geometry of the feasible region in the vicinity of optimality and this is unlikely to be known before analysis is begun. Systematically varying the weights and applying them to the objective functions removes the burden of predefining the weights from the DM.

The ϵ-constraint method allows the DM to specify upper and lower bounds for each of the objective functions in an interactive serial manner. The problem is initially solved separately for each objective function to establish some concept of what the range of the bounds should be. The DM selects a range of acceptable values for one or more objectives and the corresponding nondominated points are generated. This process continues until a solution is accepted by the DM. This method is the foundation of the Surrogate Worth Trade-Off method discussed later. (Goicoechea, Hansen, and Duckstein, 1982)

Multiparametric decomposition methodology alleviates the problem of a priori weights determination and also has important bearings on interactive procedures. Multiparametric decomposition attempts to maximize the weights associated with each objective function instead of maximizing the objective functions as separate parallel entities. Further, the set of all parameters can be decomposed into subsets associated with individual nondominated solutions. This allows the various combinations of weights to be projected in terms of corresponding nondominated solutions and forms a base for good decision maker-model interaction.

The number of computed nondominated solutions in multiobjective linear programming is often too large for a decision maker to make an intelligent identification of the most preferred one. Therefore, several approaches have been developed which would allow one to "prune," "filter," or simply "reduce" the size of the nondominated set to a manageable size.

Steuer (1977, 1979) and Steuer and Schuler (1978) have developed several reduction approaches based on the multiparametric decomposition methodology. Their approaches require the DM to state upper and lower bounds for each objective function weight. The weights are gradually contracted through interaction with the DM and his/her selection of the most preferred solution until the nondominated set is located which contains the most preferred solution for the overall problem.

Morse (1980) used the statistical technique of cluster analysis to "prune" the nondominated set. Each cluster can then be represented to the DM by an arbitrarily selected nondominated point. If the DM is interested in one solution set, then that cluster can be more fully explored.

Bitran (1977, 1979) used intermediate, feasible solutions of the zero-one multiobjective model solved by the simplex or dual-simplex algorithm for identifying nondominated points and the directions of preference along which all objectives can be improved. Kiziltan and Yucaoglu (1983) developed a branch and bound algorithm which examines the nodes of a combinatorial tree to decide whether feasible and nondominated solutions can be found on this branch or not. Both algorithms are limited to small problems.

Multiattribute Utility Theory

Several researchers have approached the multiple conflicting objectives issue from the decision maker's preferences point of view by using multiattribute utility theory (MAUT). MAUT assumptions include: perfect rationality, utility maximization, and predictability of aggregate phenomena.

Keeney (1972) developed the MAUT method which uses two assumptions of 1) preference independence and 2) utility independence to limit the utility function to specialized forms. Preferential independence concerns ordinal preferences among attributes, while utility independence is concerned with the cardinal preferences of the decision maker. The presumption behind the model was that if an appropriate utility could be assigned to each possible outcome and the expected utility of each alternative could be calculated, then the best course of action for any DM is the alternative with the highest expected utility. It utilizes an additive utility decomposition approach.

The method requires very high demands on a DM's judgments both in terms of complexity and numbers. The method verifies the independence of attributes by confronting the DM with a battery of lottery questions. MAUT then constructs each individual single-attribute utility function, again with a lottery question approach to the DM. Then it requests scaling factors (weights) for each attribute. Once all the preference information has been obtained from the DM, the alternative with the highest expected utility could be derived.

Also in 1972, Geoffrion, Dyer, and Feinberg developed an interactive procedure, known as the "GDF method." The overall preference (or utility) function is assumed to be unknown, but differentiable and with positive marginal utility, and its arguments are assumed to be well defined and the feasible set convex. As each solution set is displayed to the DM, the DM determines the new weighting for each variable in each objective function. Thus, this technique also is very burdensome to the information-processing capability of the DM.

Haimes, Hall, and Freedman (1975) created the surrogate worth trade-off (SWT) method which estimates the utility function by constructing a sequence of local preference approximations. In this method, a representative set of nondominated solutions is computed with a corresponding trade-off ratio between any two objective functions. The DM is then asked to assess the trade-offs, for one objective function at a time, while holding the other objectives at their current value. The procedure ceases when all further surrogate worth trade-offs are equal to zero.

The SWT method was extended for use with large-scale systems by Sakawa and Seo (1980). Nakayama, Tanino, and Sawaragi (1980) used Keeney's MAUT method, with trade-offs, for assessing the preference structure of decision makers as the basis for their Interactive Relaxation Method (IRM). Their method requires the DM to judge whether his/her marginal rate of substitution is more or less than the trade-off ratio for each of the objectives. Based upon these judgments, the method can identify noninferior curves of any pair among the objective to increase optimization in the direction of each axis. Their method applies to nonlinear as well as linear solutions.

The Zionts and Wallenius (ZW) method (1976) assumes that the DM's implicit utility function is linear, and it attempts to identify the set of weights at which this function is maximized. The trade-offs are shown to the DM as non-basic variables, which, when introduced into the basis, would increase one objective, while decreasing at least one other objective. This method requires the DM to make full pairwise comparisons among multidimensional solutions, which can be difficult and unreliable.

Interactive MCDM Techniques

There are three general approaches to multicriteria decision processes: 1) a priori articulation of preferences, and the generation of a solution based on these preferences 2) progressive articulation of preferences and the arriving at a desireable solution in an interactive manner and 3) a posterior articulation of preferences, or generating all relevant solutions and then choosing from among them.

The a priori articulation of preferences places a burden on the decision maker who is forced to make a decision in a situation where there is frequently an information void pertaining to both the nature of the feasible region and the trade-offs involved in various weighting schemes for individual objectives.

The a posterior articulation of preferences alleviates this problem, but may also be difficult for the DM. This method requires that a large number of various weighting schemes be calculated and a large number of decision alternatives be presented to and considered by the DM. This task can be burdensome.

The interactive process surmounts both of these problems and may even reduce the requirements of the formal modeling phase (Kavrakoglu, 1984). It requires a calculation phase followed by a decision phase and continues with this progressive articulation of preferences until a satisfactory solution is selected by the DM. It is the most intuitively appealing of the three approaches, but is considered by some to be the least workable.

The basis of the criticisms of this method is the perceived dichotomy between the ability of the system analyst to properly show the alternatives for each proposed solution and the ability of the DM to identify and understand both what the analyst is demonstrating and a better solution direction.

Nakayama, Tanino, and Sawaragi (1980) developed an interactive multiobjective optimization technique called interactive relaxation method (IRM). The model graphically displays the nondominated geometric surface for three or fewer objective functions and calculates a trade-off ratio and a marginal rate of substitution. The DM uses this information to direct the search for the best solution.

Sakawa and Seo (1980) presented an interactive system for solving large-scale problems that combines the surrogate worth trade-off method and the multiattribute utility method by using a dual decomposition method to identify nondominated solutions. It also uses marginal rates of substitution to determine the direction in which the utility function of the DM increases the most rapidly. This is used to determine a step size for generating a new nondominated solution.

Villarreal, Karwan and Zionts (1979) developed an interactive branch and bound algorithm for solving the zero-one problem that uses a heuristic to solve the problem, whose solutions may or may not be nondominated, and presents these to the decision maker. By not solving the integer problem to get a set of efficient solutions first, they are able to solve problems faster and therefore can solve larger problems.

Ozernoj (1979) used a multistep decision rule approach for determining a DM's preferences. The decision rule established a preference-indifference relationship after each solution set was shown to the DM and the DM identified which attribute to change and by what amount. The decision rule was used to order the set of feasible alternatives.

Goal Programming

The term, Goal Programming, was first coined by Charnes and Cooper in 1961, and has gained popularity in the 1960s and 1970s from the works of Ijiri (1965), Lee (1972), and Ignizio (1976). It is now considered to be an important area of multiple criteria optimization.

Steuer (1986) describes two basic GP models: the Archimedean model and the preemptive model. The Archimedean model generates candidate solutions by computing points in the feasible region whose criterion vectors are closest, in a weighted L_1-metric (distance) sense, to the utopian set (where every objective attains its optimal value) in criterion space. The preemptive model generates solutions whose criterion vectors are most closely related, in a lexicographic sense, to points in the utopian set. Thus, the preemptive model solves each objective function or class of objective functions separately, in a priority order prescribed by the decision maker, and then sets this goal equal to the level achieved and adds the goal as a constraint. This process is repeated until there are no more priority levels. The preemptive model can be solved by an efficient partitioning algorithm such as the algorithm developed by Arthur and Ravindran (1978).

A Multigoal Programming GP model has been proposed by Zeleny (1982). This theoretical model identifies all nondominated solutions with respect to objective functions with no need to specify criterion weights (either preemptive or archimedean) and further, does not utilize an aggregate preference or distance function. Thus far, this model has not appeared in the goal programming literature.

The extensions of goal programming into interactive goal programming attempt to alleviate the weakness in goal programming with respect to the a priori setting of goals. Monarchi et al. (1973, 1976) developed an interactive GP technique that defined goals as required values of objectives which are difficult to change because they are imposed on the decision maker by external circumstances and aspiration levels as desired values of objective functions, which may change due to learning, improved understanding, or shifts in a preference pattern. Their technique allowed the DM to state a beginning weighting on aspiration levels, and then the ability to change the weights on aspiration levels, or to change the aspiration levels themselves, in order to bring unsatisfied aspiration levels into the solution. These weights can be interpreted, however, as preemptive priorities. For each change in weights or levels, the problem was resolved.

Nijkamp and Spronk (1978) developed another interactive multiple goal programming (IMGP) technique that also attempts to avoid setting aspiration levels and priority weights a priori. Their technique calculates the minimum and maximum achievable values for each objective function, with respect to the feasible set. A trial solution is initially presented to the DM at the minimum achievable values for each objective. The DM can then indicate which objective should be raised. The minimum values of the current solution then become constraints (which reduces the feasible region), the selected objective is raised to the midpoint of its potential and a new solution is returned. The process continues until there is no potential for improvement of any of the objectives, or until the DM identifies a satisfactory solution.

The model assumes a high degree of consistency in the DM's expression of preferences, and it assumes that the DM will always recognize a satisfactory solution, which may be an erroneous assumption. It further uses an implied preemptive prioritization of objectives by having the DM raise each of the objectives to its acceptable value in the order of their importance. It also does not allow the DM to shift priorities of previously 'raised' objectives.

More recent interactive techniques include Hwang and Masud's (1979) interactive sequential goal programming (ISGP) which attains a "best compromise solution." It makes use of upper and lower bounds for the DM to use in setting goals. The method is advantageous because nondominated solutions are guaranteed, nonlinear problems can be solved, and a variety of solutions are presented to the DM at each iteration in order to guide him/her in future refinements of options. The disadvantages are the difficulty of finding an initial feasible solution and of finding a feasible solution after the DM has modified the goals. This conceivably could necessitate a considerable amount of interaction with the DM. The number of problems solved could also be high relative to other approaches. (Zeleny, 1984)

Another interactive goal programming algorithm was introduced by Nijkamp and Spronk (1978) and Spronk (1981). Their method allows the DM to improve goals in small increments without severely penalizing the achievement of other goals and for changing more than one goal simultaneously. Disadvantages in the method include the need to specify distinctive gradations of aspiration levels for each goal and the difficulty for the DM in determining if the proposed solution is satisfactory.

Compromise Programming

The concept behind compromise programming was first introduced by Zeleny in 1972. It is an interactive technique that does not require the pre-setting of weights on objective functions. It evolved from the notion of the displaced ideal, or the concept that an "ideal" solution for all objectives is usually infeasible, and therefore must be replaced by an acceptable, but displaced ideal, alternative. The displaced ideal concept was briefly introduced by Geoffrion (1965) under the term "perfect solution." The concept also appears in works of Radzikowsky (1967) and Juttler (1967). Saska (1968) provided the first fully operational use of the displaced ideal in a linear multiprogramming methodology. Dinkelbach (1971) also reviewed the concept. More recently Evren (1987) proposed a solution method for MOLP problems called Interactive Compromise Programming, which combines the method of compromise programming and a two-person zero-sum game in an interactive manner.

Under the term "movable target," a progressive orientation procedure for exploring non-ideal solutions was devised by Benayoun and Tergny (1969). The more developed model, based upon Benayoun and Tergny's procedure, called

STEM (STEP Method) was published by Benayoung, de Montgolfier, Tergy, and Larichev (1971). Their model included an iterative component in which the computations were performed and the "nearest" solution (in the MINIMAX sense,) and a payoff matrix showing the maximum achievable level for each objective were shown to the decision maker. The decision maker could then examine the results and give new weighting information for each objective. The procedure ceases when a solution is chosen.

Aubin and Naslund (1972) used the same concept under the term "shadow minimum" in an exterior branching algorithm devised in the game-theoretical framework. Zeleny (1973, 1974) introduced the concept of the compromise set and developed the displaced ideal method. Roy (1977) introduced the evolutive target procedure which is based upon sequential displacements of the ideal solution.

After Zeleny's introduction of the technique, he authored several follow-up publications (1974, 1975, 1976, 1977, 1981, 1982) and Yu (1973) extended the concept to group decision making.

The mathematics of compromise programming, which are based on the idea of minimizing distance from feasible, nondominated solutions to an ideal point, were in the early work of Benayoun et al (1971). Ecker and Shoemaker (1980) developed an algorithm for finding all the efficient points for linear multiple objective optimization. They also introduced the "trade-off compromise set," which is a special subset of efficient solutions and showed how this set provides information about possible trade-offs among objectives.

A general notion of a compromise solution appears in several works of Salukvadze (1971, 1972, 1974, 1982), but particularly in his study of solutions to dynamic optimal control problems with multiple objectives. Yu and Leitmann (1976) pointed out the relationship of Salukvadze's work to compromise programming. Bowman (1976) described how the efficient frontier can be generated from compromise solutions by varying the ideal point. Gearhart also generated the efficient frontier by varying the weights with a fixed ideal point. Gershon (1984) studied the role of weights and scales in multiobjective decision making. Gearhart (1985) also developed an abstract framework for the analysis of compromise programming.

Compromise programming has been applied to interactive water resource planning using the STEM method of Benayoun et al (1971), and by Loucks (1977). Nijkamp and van Delft (1977) used interactive compromise programming for regional planning. Bardossy and Bogardi (1983) utilized a composite form of compromise programming in observation networks of several spatially correlated and anisotropic environmental variables. Madey and Burton (1985) utilized both goal programming and compromise programming in a study of project selection and budgeting strategic planning.

Summary of MCDM Methods

Multiple criteria methods are viewed by many as being less rigorous than those of the axiomatic, classical decision analysis field and therefore, not acceptable by the standards of that discipline. On the other hand, the multiple criteria researchers perceive themselves as extending the classical decision analysis to encompass the characteristics of an important set of decision problems. (Starr, 1982)

There is no simplistic response to this criticism, but Starr suggests that the MCDM researchers perceive themselves as a separate field that defines a bridge to the behavioral sciences. In this context, the definition of optimality would be based upon what is feasible and desirable for decision makers.

There have been several extensions of the Linear Programming model to accommodate multiple criteria. Linear Multiple Objective Programming is a logical extension, but it applies only to linear objectives, which limits its applicability.

Goal programming has been and continues to be one of the most active areas of research, but it has three main areas of criticism: the use of preemptive priorities; the difficulty in choosing a priori weights; and the difficulty in choosing goals.

These criticisms assume higher or lower priorities based upon the DM's depth of knowledge of the problem. If the DM has a good idea of what goals can be achieved in view of the constraints of the problem, i.e. a plausible concept of the feasible region, then he/she can set reasonable preemptive priorities, a priori weights and specify sound goals. However, if the DM does not have a good idea of what goals can be achieved in view of the constraints, then the criticisms hold.

There are several differences between goal programming and compromise programming: compromise programming determines its goals internally through computations whereas goal programming requires the goals to be initially established; compromise programming does not use preemptive weighting whereas some goal programming algorithms do; and compromise programming considers a large variety of distance functions for its objectives whereas goal programming considers one.

Gershon (1984) noted that all multiobjective decision making requires, at some point in the analysis, the specification of the preference structure of the decision maker over the set of objectives. He further notes that in goal programming, which some consider to be a subset of compromise programming, the choice of a goal point is somewhat arbitrary, while compromise programming implies the use of a specific ideal point, the ideal point being that vector made up of the best values attainable for each of the objectives.

In summary, the essential feature of compromise programming is the determination of solutions whose criteria values are close, according to some measure of distance, to a given ideal criteria value. It combines the best and most useful features of Linear Multiobjective Programming and Goal Programming. It does not require the a priori determination of criteria weights that Goal Programming requires, nor is it limited to linear problems. Plus, it can identify nondominated solutions under the most general conditions; it does allow prespecified goals; and it provides a base for interactive programming (Zeleny, 1982).

Computer Network Design

The design of computer networks historically has revolved around the availability of communications connections to provide the links between the network nodes. The analyst designing the network attempts to combine these available links in such a manner as to meet several design criteria such as high speed of transmission between connected computers; a wide area of transmission capability; high reliability; and low cost.

Typically, multiple criteria decision making methods are not used. Instead, one of these objectives is selected as having a top priority and the network is designed to maximize/minimize this objective, through either optimization or heuristic techniques, while meeting minimal/maximal requirement levels for the other objectives.

The requirement levels for the non-priority objectives are generally verified using a variety of heuristics. If the requirement levels are not met, then the original design is regenerated and checked again for meeting requirement levels. This process continues until a satisfactory design is produced.

One of the earliest implemented computer networks was the (Defense) Advanced Research Projects Agency of the U.S. Department of Defense (ARPANET), which started in the late 1960s by providing grants to computer science departments at several United States universities and a few private corporations. Their research led to an experimental four-node network that was launched in December of 1969. Today over one hundred computers spanning half the globe, from Hawaii to Norway, are serviced by the network.

The terminology used in the following discussion was adopted from the ARPANET project. The general computer network design model was suggested by Tannenbaum (1981). The discussion is intended as an overview of historical network design tools and contains a representative sample of the optimization and heuristic approaches used in the design of computer networks.

The necessary components in a communications network include hosts (processors) connected by the communication subnet (transport or transmission system composed of circuits or channels) through Interface Message Processors (communication computer, packet switch, front-end processor or data switching exchange). The host is typically linked to one Interface Message Processor (IMP). Terminals may be connected to concentrators or multiplexors that are in turn connected to an IMP. In addition, one or more terminals are generally connected to a terminal controller.

There are two general designs for the communication subnet: point-to-point (store-and-forward) where pairs of IMP's are connected; and broadcast channels where a single communication channel is shared by all IMPs. In a point-to-point subnet topologies could be a star, loop, tree, complete, intersecting loops or irregular. In a broadcast channel subnet, topologies could be a bus, a satellite, a radio or a ring. Several taxonomies for computer networks have been proposed, but no consensus has been achieved. A survey of the proposals is given by Jensen et al. (1976).

The classifications of solution methodologies that seem best suited for overall network design analysis include: overall network topological design; backbone analysis; local access network design; connectivity analysis; and delay analysis.

Overall Network Topological Design

The topological design problem has been typically characterized by having as givens the location of the host computers and terminals, the traffic matrix, and the cost matrix (setup and/or operating costs); as performance constraints reliability and delay and/or throughput; as decision variables the network topology, line capacities, and flow assignment; and a goal of minimizing cost (Tannenbaum, 1981). The locations of the hosts and terminals typically are considered as equivalent and could also be referred to as a location or site.

The traffic matrix identifies the number of packets sent to site i from site j. In a new network, this is usually unknown, but could be estimated as either 1) the product of the populations of the two sites, divided by the distance between them or 2) by some other reasonable distribution function, such as an actual distribution function from a known and comparable site. The probability density function usually is assumed to be known and the same for all sites.

The operating cost matrix gives the cost per month for a leased line from site i to site j. These costs include costs for IMPs and concentrator locations as well as hosts. Usually the cost of a line depends on distance and speed in a highly nonlinear fashion. There also is usually a fixed charge that is dependent only upon the speed, such as modem depreciation.

The performance reliability constraint is concerned with the issue of the network not collapsing if one IMP or line goes down. The delay/throughput constraint deals with the requirement of minimal acceptable delay time and/or minimal acceptable delivery time.

The topology decision variable usually is concerned with the placement of IMPs, concentrators and lines. It is assumed that the hosts are givens. The other decision variables are typically line capacity and flow assignment (routing algorithm). The problem, however, can quickly become huge. With n locations there are n(n-1)/2 potential lines that may or may not be present yielding $2^{n(n-1)/2}$ possible topologies. Exhaustive methods would not work. Therefore, to simplify analysis, usually the only objective is to minimize overall cost. Conceivably though, any of the resource constraints could also become objectives. (Dutta, 1986)

Past experience has provided a model that can be used as a starting design strategy. The scheme is to divide the design problem into a hierarchy and solve each level of the hierarchy separately (Tannenbaum, 1981, Boorstyn & Frank, 1977, Gerla & Kleinrock, 1977). One level of the hierarchy could be the highly redundant network that connects the IMPs known as the backbone design. Another level of the hierarchy could be the local access design that ties hosts to the backbone. This allows the local access topology to be a tree, which eliminates the routing problem at the local access level.

Backbone Design

The design for the backbone historically has been iterative. A potential design is generated and then checked to see if it complies with the connectivity and delay/throughput constraints. If it does not, another design is generated. If it does, the cost is computed. This topology is then used as a starting point. Small perturbations are made to the model to check for better solutions. Then a new, feasible design is generated and it is perturbed. Considerable work, much of which is unpublished and proprietary, has been done on choosing starting topologies, assigning the flow and capacities, and generating perturbations of the starting topology (Boorstyn & Frank, 1977, Frank & Chou, 1972, Frank et al., 1970).

One heuristic for generating a starting topology is based on Whitney's theorem (Steiglitz et al., 1969), which assures that in a k-connected network every IMP has at least k links. Shortest-path routing is a recommended algorithm for flow and capacity assignment. (Chou & Gerla (1976,) Kleinrock & Gerla (1980,) McQuillan & Walden (1977,) Pouzin (1976,) Schwartz & Stern (1980))

One method for perturbing the network is known as branch exchange. This method selects two links to remove from the network and adds two new links using another combination of the four nodes in the two omitted links. Gerla used a variation of the branch exchange method in a saturated cut heuristic which both adds and deletes links based upon some criteria, such as cost or utilization, as a method for perturbing the network. (Gerla et al., 1974)

A problem with the perturbation heuristics is that they might reduce the connectivity of the network. Each network perturbation requires a subsequent search for proper connectivity. A transformation method by Lavia and Manning (1975) produces networks whose connectivity is at least as high as the original network. This can save running the time-consuming connectivity algorithm for each new perturbed network.

Other methodologies which could be used for the backbone design problem are a shortest path algorithm (Dijkstra, 1959, Lawler, 1976, Gerla and Kleinrock, 1977), an all-pairs shortest path algorithm (Floyd, 1962, Danzig, 1966, Tabourier, 1973), and a minimal spanning tree algorithm (Boorstyn and Frank, 1977, Ferguson and Mason, 1984). A shortest path algorithm has a starting or source node and a terminal or ending node and finds the shortest path between the two. Dial, Glover, Karney, and Klingman (1979) give a comprehensive review and comparison of alternative algorithms for solving shortest path problems. The all-pairs shortest path algorithm finds the shortest path between all pairs of nodes in the network and considers intermediate nodes on the path. The minimal spanning tree, or minimum weight spanning tree designs a spanning tree with the minimum sum of arc weights (Hillier and Lieberman, 1980).

The all-pairs shortest path problem is a variation of the shortest path algorithm. This algorithm finds the shortest path between all pair of nodes in a network in an iterative fashion while considering intermediate nodes on the path. The shortest path between any pair of nodes can also be determined by solving the shortest path algorithm repetitively using each node in the network as a source node and every other node in the network as a terminal node. Kelton and Law (1978) found in comparing the all-pairs algorithms developed by Floyd (1962), Dantzig (1966), and Tabourier (1973) (modification to Dantzig's algorithm) with solving the single source shortest path algorithm and sequentially varying the source through all of the nodes that Tabourier's algorithm generally out performed the others. The exception was for very sparse networks and multiple application of the single source algorithm performed most efficiently.

<u>Local Access Network Design</u>

The three basic subproblems usually solved in local access network design are the concentrator location, assignment of customer sites to concentrators, and terminal layout within a site. Zero-one programming has been used to determine the concentrator location. (Kershenbaum & Boorstyn, 1975) This approach is intractable, however, in large networks.

A simple heuristic suggests forming a matrix where rows are sites and columns are concentrators, with the matrix values being costs. The heuristic then scans the rows in several row orders, selecting the minimum cost in each row. An assumption of this heuristic is that each site can only be assigned to one concentrator and that the concentrator has a given number of possible assignees. An extension to the algorithm would be to keep track of concentrators that were in demand after being filled and possibly reassigning some of the sites to other concentrators at a lower overall cost.

Once the assignments of sites to concentrators is determined, then a location for the concentrator must be made. (McGregor and Shen, 1977) Two heuristics are ADD (Kuehn and Hamburger, 1963) and DROP (Feldman et al., 1966). The ADD algorithm starts with all sites attached to the central site and then adds concentrators either leaving a site attached to the central site or attaching it to the added concentrator. The DROP algorithm is the exact opposite of the ADD algorithm. It begins with all the concentrators in use and then begins to drop a concentrator, assigns its customers to other concentrators, and then computes total cost. It then puts the dropped concentrator back in, drops a different concentrator and repeats the process for each concentrator. Its intention is to eliminate uneconomical concentrators.

The terminal layout problem can be solved by use of a minimal spanning tree algorithm. (Chandy & Russell, 1972; Dutta, 1986) This technique solves the uncapacitated problem and, with partitioning heuristics, solves the concentrator constrained problem.

Connectivity Analysis

The reliability constraint can be met with a high degree of success, even with unreliable components, if the network is redundant. Or, high reliability can also be achieved through the use of store-and-forward and packet-switching techniques (Newman, 1987; Leigh and Burgess, 1987) These techniques have become a standard method of switching data on both wide area and local area networks.

The techniques are particularly effective for local access networks, where distance and transfer speed requirements are minimal. They also are appropriate for wide area (long haul) networks, but are generally accompanied by error-detecting and error-correction schemes. This is largely due to the wide use of telephone (voice-grade) lines as a transmission media. Voice-grade lines tend to be slow and to have a high error rate and therefore require schemes to recognize and correct for transmission errors.

There are several methods that could be used to study the redundancy of a network, including spanning tree and minimal spanning tree, (Dijkstra, 1959; Chandy & Russell, 1972) cuts and network flow and the max-flow algorithm, (Stone, 1977; Malhora et al.,1978) disjoint paths, (Kleitman, 1969; Even, 1975; Locks, 1982) Markov modeling, (Lazaroui & Staicut, 1983) and Monte Carlo connectivity analysis.

The minimal spanning tree has been used to generate a design for the local access network. Minimum cut and maximum flow algorithms are techniques that have been used for analyzing topological network reliability. Disjoint path analysis can be used when network designers are confronted with the problem of whether or not a proposed network is k arc-connected or k node-connected. The objective of the analysis is to answer the question of whether a graph can lose k nodes and still remain connected.

Delay Analysis

The delay in a communication network can be due primarily to three reasons: if network traffic is light, the delay is primarily due to the time that the IMP spends in storing and forwarding a packet; if there is a long distance between the IMPs, propagation delay account for a part of the delay; or as traffic increases, the waiting time in a queue becomes the principal delay. (Kleinrock, 1976)

Queueing theory is generally the foundation for delay analysis. Little's result has been used to find the total waiting time, including the service time. (Little, 1961) The Pollaczek-Khinchine equation has been shown to be valid for any service-time distribution. The network becomes an open network with M/M/1 queues which can be independently solved. (Jackson, 1957) The wait times can then be summed to obtain total wait time. Variations of this model can be used to accommodate arrival patterns that are not Poissonly distributed.

However, to achieve maximum flow, an infinite delay must be tolerated. The delay and throughput constraints are inherently in conflict. The network designer must take this conflict into account during the design process.

MCDM Applied to Computer Networks

There have been many attempts to optimize individual components of network design, such as the network topography, or the network reliability, but the optimization of the overall design, in particular regarding multiple objectives, has had little reported research.

Yet, the concept of MCDM is attractive to computer network design, where designers historically have attempted to satisfy many different objectives with heuristics and optimization techniques, in sequence. These isolated network components could conceivably be incorporated into one all-inclusive representation of a network and solved with MCDM techniques.

Dutta and Jain (1984) used the Exchange Search Heuristic of Spath (1977) to develop a command driven DSS for designing the optimal use of available computer resources with multiple conflicting objectives. The objectives were to place databases on the network in such a manner that would minimize system cost, which consists of the sum of processor costs (based upon a reliability probability with less reliable processors being assigned a higher cost), communication network (operating) costs, and file storage cost (based upon the amount of storage space used); minimize weighted average response time, which is a routing decision; and maximize file availability. Their model includes decision variables for selection of processing power and location thereof (assumes multiple processors at each site); selection of channel capacity (baud rate) and network topology; and the location of database files. The exchange search heuristic requires an initial feasible solution from either the user or from the system. A minimal spanning tree algorithm was used to generate the system's initial feasible solution if the user chose not to supply one. The nondominated solution set was generated using an ϵ-constrained approach.

Several researchers have concentrated on the bicriterion or multiobjective shortest path problem. White (1982) used a multiple objective weighting factor method for generating the efficient solutions for shortest path problems. Shetty, Olson, Venkataramanan (1988) concentrated on an upper and lower bound technique for generating efficient solutions. Hansen (1982) developed a labeling algorithm to aid in the identification of nondominated solutions. Shier (1988) implemented a general label correcting algorithm for bicriterion networks and found the algorithm to be efficient at solving reasonably large problems. Henig (1986) developed two procedures, based on maximizing a decision maker's utility function, to generate the nondominated set. Climaco and Martins (1982) solved the bicriterion shortest path problem for total time and total cost by generating the k-shortest paths between the solutions optimizing the first and the second objectives. Warburton (1987) solved the problem by introducing a knapsack-type of constraint. Mote, Murthy, and Olson (1988) in solving bicriterion shortest path problem found that the number of nondominated solutions was not as prohibitively large as was previously assumed.

The advantages of using multicriteria optimization are that it allows the DM to achieve some compromise among conflicting objectives by appropriately manipulating the decision variables and during any one iteration it allows the DM to concentrate on only the "best" alternatives for that solution. This enables the designer to experiment with perturbations and see the resulting solutions.

The tools for solving MCDM problems have evolved and abounded within the last decade and are becoming more viable as solution tools for unstructured decision problems such as computer network design. The progressive articulation of preferences, as the designer learns about the feasible region and the impact of the trade-offs between the objective functions, make MCDM a particularly attractive decision support option.

Conclusions From Literature Review and Justifications of the Study

The literature review points out that little has been done to integrate the areas of multiobjective decision making and the topological design of computer networks. Dutta and Jain's (1986) DSS for a distributed computer system design is the only paper that appears to have applied MCDM to the design of a computer network. Dutta and

Jain's DSS for network design is a very similar problem to this study, but their approach to the problem and its solution is different: their focus is on the optimal location of databases and paths to the databases in order to minimize cost and time, rather than the initial design of a computer network; their prototype implementation uses English-like commands for interacting with the user and tabular displays for showing nondominated solutions, rather than graphics; they use the ϵ-Constraint method for generating nondominated solutions, rather than the weighting method; their problem is solved using the branch-exchange algorithm, rather than the minimal spanning tree and all pairs shortest path algorithms; their DSS is implemented on a mini-computer, rather than a personal microcomputer.

Several references to bicriterion shortest path algorithms and their implementations exist, but they have not been tested in a decision making context. Warburton (1987) and White (1982) have developed the theories of the tricriterion problem, but have not implemented their theories. A test of the effectiveness of these algorithms in helping a decision maker make a decision on networks could be useful.

There has been considerable research using either the minimal spanning tree algorithm or the shortest path algorithm for designing computer networks, but the use of both algorithms for generating the nondominated solution set does not appear in the literature. The minimal spanning tree algorithm is particularly useful for generating a solution requiring a design that minimizes overall setup and operating costs and the all pairs shortest path algorithms are appropriate for minimizing service time between all pairs of nodes. The use of a combination of the two algorithms could exploit the strengths of both algorithms.

It should also be noted that the majority of the interactive multiobjective systems that have been implemented do not make use of readily available user-interface tools such as databases and graphics. The typical user interface is in a question and answer form with a tabular display of solution results and choices to make. There is considerable evidence that the use of graphical tools would facilitate the use of a DSS by a decision maker and aid in his or her understanding of the problem and its solution. Hurrion (1986) and Billington (1987) both state a significant improvement in a DM's understanding of a semi-structured to unstructured operations research problem and the DM's subsequent confidence in a solution when visual interactive modelling tools are used. The 1982 Wharton Business School's graphics study (Meilach, 1986) concluded that graphics positively influence how decisions are reached and the time required to reach a decision.

This study was an attempt to apply the concepts of multiple criteria decision making to a telecommunications network design problem as an interactive decision support system, to implement the system on a microcomputer, and to test the system's effectiveness in the decision making process.

The problem was stated, however not solved, as a linear programming formulation with three objective functions and setup cost, operating cost, transfer time, and connectivity constraints. The system incorporated the following algorithms:

.Compromise Programming was selected as the MCDM tool because it did not require the a priori specification of weights by the DM and because it allowed the user to be inconsistent while learning the trade-off nature of solutions in the solution set.

.The weighted method of generating nondominated solutions was chosen for the same general reason - it did not require the DM to set upper and lower bounds for each of the objective functions.

.The Minimal Spanning Tree Algorithm and the All Pairs Shortest Path Algorithm were selected for the strength that each would bring in the generation of solutions.

.A G/G/1 queueing model was used for calculating transfer times between pairs of computer nodes.

The implementation of the system utilized the following tools:

.A relational database for storing the problem data and solutions.

.A graphical toolkit for creating forms and solution displays.

CHAPTER III

THE MODEL AND ALGORITHM

Decision Problem

The general application problem consists of designing a network that connects several remote facilities. The design variables identify (1) the pattern for connecting the nodes and (2) the communications media to be used in the connection. The model assumes that there will be zero, one or two communications media connecting any two nodes and that there will be one effective transfer rate for each possible communications media between these nodes. The DSS originally was conceived to solve a specific problem which involves designing a computer network that connects University Park (a multi-use industrial-business park adjacent to University Center at Tulsa) with six remote facilities. (See Figure 1).

Decision Variables and Notation

The user interactively enters data for distances between nodes (b_{ij}), traffic patterns between nodes (peak traffic flow (t^{p}_{ij}), minimum traffic flow (t^{m}_{ij}), most

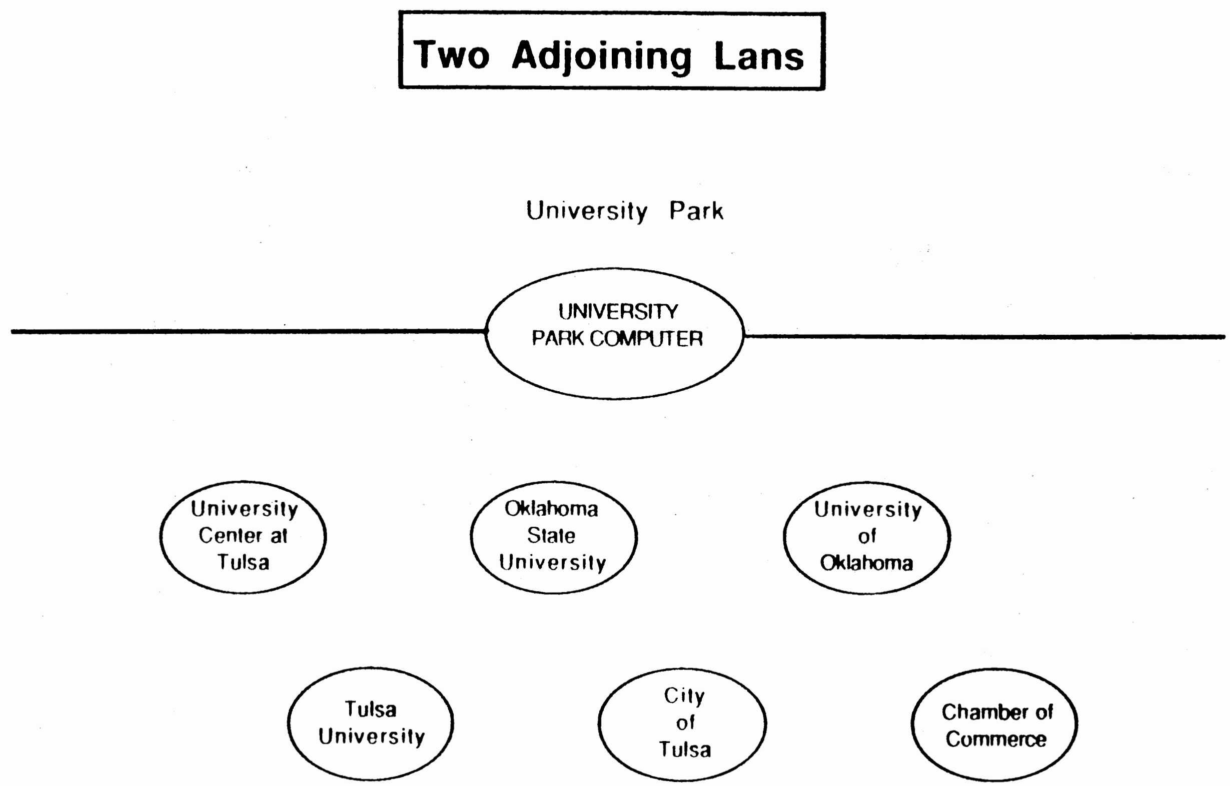


Figure 1. University Park Computer Lan

likely traffic flow (t^{l}_{ij}), estimates of file sizes to be transferred (large (f^{p}_{ij}), medium (f^{l}_{ij}) or small (f^{m}_{ij})), original setup costs (s_{ijk}) and operating costs (o_{ijk}) for communications media between nodes and store this data in a relational data base. The user also defines a minimum desired service level (S), a maximum setup cost (U), and maximum operating costs (O).

The model generates the effective service level for all possible communications connections, using a general arrival and general service time queueing formulation. From this data, the model will establish the objective functions to minimize service level, minimize setup costs and minimize operating costs and also generate the coupling constraints.

<u>Summary of Notation:</u>

n = number of nodes in the network

m = maximum possible connections between nodes

$$= \left[\frac{(n!)}{((n-2)!)(2!)} \right]$$

p = number of objective functions

q = maximum number of communications connections (i.e., fiber optics, satellite transmission, coaxial cable, telephone line, etc.)

F_l = objective functions where l=1,...,p

x_{ijk}=decision variable representing a connection (one) or no connection (zero) between node i and node j for a given communication media k between the two nodes

$$\begin{cases} 1, & \text{a media connection between two nodes} \\ 0, & \text{no media connection between two nodes} \end{cases}$$

b_{ij} = distance between node i and node j

where $i=1,\ldots,n$
$j=1,\ldots,n,\ i \neq j$

t_{ij} = traffic pattern between node i and node j

where $i=1,\ldots,n$
$j=1,\ldots,n\ i \neq j$

t^{l}_{ij} - most likely traffic pattern between two nodes

t^{p}_{ij} - pessimistic (maximum) traffic pattern between two nodes

t^{m}_{ij} -optimistic (minimum) traffic pattern between two nodes

f_{ij} = file size to be transferred between node i and node j where $i=1,\ldots,n$
$j=1,\ldots,n,\ i \neq j$

f^{l}_{ij} - most likely transferrable file size between two nodes

f^{p}_{ij} - pessimistic (maximum, largest possible) transferrable file size between two nodes

f^{m}_{ij} - optimistic (minimum, smallest possible) transferrable file size between two nodes

s_{ijk}=original setup cost between node i and node j for communications media k

where $i=1,\ldots,n$
$j=1,\ldots,n,\ i \neq j$
$k=1,\ldots,q$

r_{ijk} = effective transfer time between node i and node j for communications media k. This is approximation based on f_{ij} and t_{ij}.

where $i=1,\ldots,n$
$j=1,\ldots,n,\ i \neq j$
$k=1,\ldots,q$

o_{ijk} = operating costs between node i and node j for communications media k

where $i=1,\ldots,n$
$j=1,\ldots,n,\ i \neq j$
$k=1,\ldots,q$

l_k = maximum transfer rate for communications media k

where $k=1,\ldots,q$

S = maximum desired delay between any nodes

U = maximum total setup cost

O = maximum total operating cost, per time unit

Network LP Model

The model allows for three objective functions to: minimize transfer time between any pair of nodes; minimize setup cost; and minimize operating costs. There are n nodes in each network and (n!)/[((n-2!))(2!)], or m, possible links between nodes. There are q possible types of communications connections between nodes. Therefore, there are $2^{q \cdot n(n-1)/2}$ nonredundant paths through the network.

The effective transfer time between a pair of nodes is calculated using a single server, general input, and general service (G/G/1) queueing model approximation of the upper and lower bounds for the total time in the system, W_{ijk} (Gross and Harris, 1985). The arithmetic average of these bounds is then used as the coefficient for the transfer time objective function. This coefficient should not be construed to be the average transfer time, since the underlying probability distributions are general. However, it is the best surrogate measure available.

The calculated time makes use of the three estimates for traffic patterns and file sizes that were entered by the DM, with the heaviest weighting on the most likely estimate, as follows:

Lower Bound (W^{LB}_{ijk}) Upper Bound (W^{UB}_{ijk})

$$\frac{\lambda^2 \sigma^2_B + \rho(\rho-2)}{2\lambda(1-\rho)} + \mu \le W_{ijk} \le \frac{\lambda(\sigma^2_A + \sigma^2_B)}{2(1-\rho)} + \mu$$

Where λ = mean arrival rate of files being sent between two nodes calculated from relative traffic patterns as

$$\lambda = 1/(t^{p}_{ij}+4t^{l}_{ij}+t^{m}_{ij})/6$$

σ^2_A = variance of the arrival rate

μ = mean service rate of files being sent between two nodes calculated from relative file sizes and maximum transfer rate for the communication media as

$$\mu = 1/\frac{[(f^{p}_{ij}+4f^{l}_{ij}+f^{m}_{ij})/6]}{l_k}$$

σ^2_B = variance of the service rate

$\rho = \lambda/\mu$

$$r_{ijk} = \frac{W^{LB}_{ijk} + W^{UB}_{ijk}}{2}$$

The service objective is to minimize the transfer time between all pairs of nodes for each possible type of communication media:

$$\min_x F_1(x) = \sum_k\sum_j\sum_i r_{ijk} * x_{ijk} \quad \text{where } l=1,\ldots,p \qquad (1)$$

The setup cost and the operating cost objective functions can be established directly from the input data. The objectives will be to:

(1) Minimize the setup cost between two nodes for each possible type of communication media:

$$\min_x F_2(x) = \sum_k\sum_j\sum_i s_{ijk} * x_{ijk} \quad \text{for } i=1,\ldots,n;\ j=1,\ldots,n\ i \neq j;\ k=1,\ldots,q \qquad (2)$$

(2) Minimize the operating cost between two nodes for each possible type of communication media.

$$\min_x F_3(x) = \sum_k \sum_j \sum_i o_{ijk} * x_{ijk} \qquad \text{for } i=1,\ldots,n \quad j=1,\ldots,n \; i \neq j \quad k=1,\ldots,q \tag{3}$$

Coupling constraints will:

(a) force there to be at most two direct communication connections between any two nodes

$$\sum_k x_{ijk} \leq 2, \text{ for every } i,j \tag{4}$$

(b) force there to be at least two connections for every node in the network to provide some redundancy

$$\sum_{i \neq j} \sum_k x_{ijk} \geq 2 \text{ for every } i \qquad j=1,\ldots,n \; i \neq j \quad k=1,\ldots,q \tag{5}$$

A path is a finite sequence of arcs $\ell=(a_1,a_2,\ldots,a_b)$ such that for each h=2,...,b arc b starts at the end of arc_{b-1}. ℓ is called a path from node i to node j if a_1 begins at node i and arc_b stops at node j. There are $2^{q\cdot[n(n-1)/2]}$ such paths possible in the network. Let £ be the set of all paths between node i and j. For the purpose of this constraint, we define

$$\ell = \{(1,2),(2,3),\ldots\}$$

$$x_{ijk} = \prod_{m,n \in l} x_{mnk}$$

The service constraints state that the maximum allowed delay between any pair of nodes be less than S. For each set of paths ($\pounds_{ij}$) between a pair of node i,j, the following constraint is needed:

$$\sum_k \sum_j \sum_i r_{ijk} x_{ijk} \leq S \qquad (6)$$

where $i = 1,\ldots,n$
$j = 1,\ldots,n \quad i \neq j$
$k = 1,\ldots,q$
S maximum allowed delay input by user

As one can see, constraint set (6) results in $2^{q \cdot [n(n-1)/2]}$ constraints.

The setup cost constraint restricts the total cost of all connections to an upper limit.

$$\sum_k \sum_j \sum_i s_{ijk} * x_{ijk} \leq U \qquad (7)$$

where $i = 1,\ldots,n$
$j = 1,\ldots,n \quad i \neq j$
$k = 1,\ldots,q$
U is maximum setup cost input by the user

The operating cost constraint restricts the total operating cost to an upper limit.

$$\sum_{k}\sum_{j}\sum_{i} o_{ijk} * x_{ijk} \leq O \qquad \text{where } i = 1,\ldots,n \quad j = 1,\ldots,n \quad i \neq j \quad k = 1,\ldots,q \qquad (8)$$

O is the maximum operating cost input by the user

Network Path Combinations

One approach to solving the model would be to use an optimization technique, such as integer linear programming, to obtain an optimum extreme point solution for each objective function. The extreme points could then be manipulated to generate other nondominated points within the feasible region and thus define the nondominated solution set. However, the problem as defined is considered to be a hard integer linear programming problem which could require running times and data storage requirements that would grow exponentially with the size of the problem (Boorstyn and Frank, 1977). For example, the transfer time restriction for the problem stated in the previous section requires that each possible combination of node to node transmission, with or without an intervening node, be defined as a constraint. The number of constraint paths thus defined is so large ($2^{q \cdot n(n-1)/2}$) that except for trivially sized networks, commercially available integer linear programming codes cannot be used. The data storage requirements for the integer linear programming codes are substantial, and the number of constraints must be kept to under a few thousand. This clearly precludes the use of an integer programming code.

Brown and McBride (1984) report that even though networks can be solved as linear programming problems, contemporary commercial linear programming systems consume more computer time and data storage requirements that do specialized codes. Golden, Ball, and Bodin (1981) and Glover and Klingman (1975) state that specialized network codes can outperform generalized linear programming codes with less memory requirements. As described later, we solved this problem using two network algorithms: minimal spanning tree, and all pairs shortest path problem.

Application of Compromise Programming

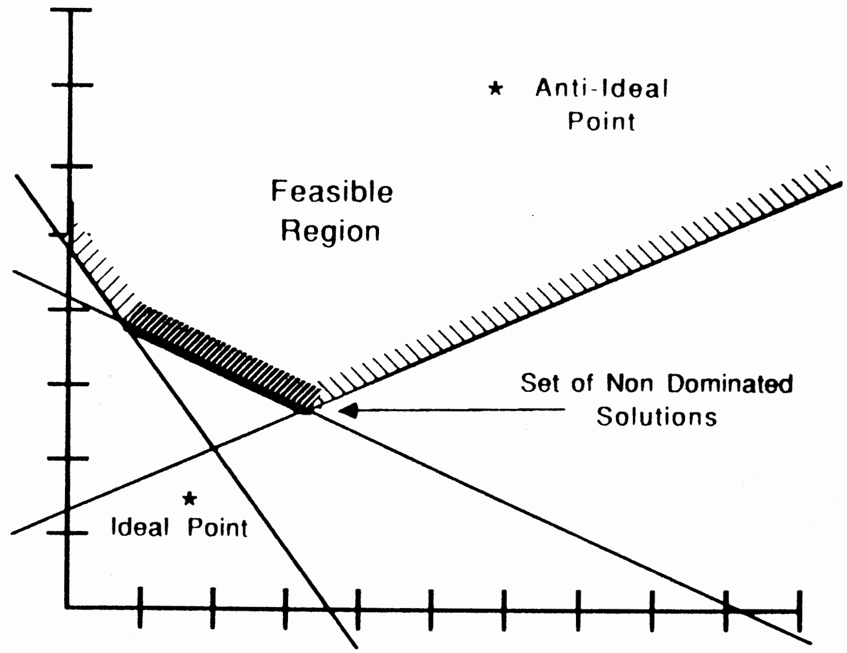
Since our problem is a multiobjective problem, we combine the use of network algorithms with a MCDM technique described earlier-- Compromise programming.

Nondominated Solutions

The nondominated concept, which is also known as a Pareto-optimal, noninferior, or efficient frontier solution is the basis for the compromise programming model. A nondominated solution is one where an improvement in any objective function can only occur at the expense of another objective function. Mathematically expressed, a nondominated solution is when a decision x* exists such that there does not exist another $x \epsilon S$ such that $F_k(x) \leq F_k(x^*)$, and $F_k(x)$ not = $F_k(x^*)$.

The compromise programming model identifies the set of all nondominated solutions or nondominated extreme-point solutions. The model computes both the ideal and the anti-ideal solutions from the solution set. These are used as reference points to aid in the identification of the ranges or potentials of change for each objective.

The nondominated set can be obtained by the weighted combination approach. The idea of assigning weights (a relative weight or worth) to several objective functions and then combining them into a single objective function for which solution methods exist is attributable to Zadeh (1963). He further proposed that the nondominated set can be generated by parametrically varying these weights and that the weighting method follows directly from the Kuhn-Tucker conditions for a nondominated solution. (Goicoechea, Hansen, and Duckstein, 1982) An extreme point solution is obtained for each of the objective functions and the entire nondominated set can be generated from linear combinations of these nondominated extreme points. (See Figure 2) The selection of the linear combinations, or weights, can



Ideal And Anti-Ideal Points,
And the Feasible Region

Figure 2. Nondominated Solution Set

influence the size of the nondominated set, the nature of the nondominated set, and the computation time. Several different weight sets can generate the same nondominated extreme point. And, a nondominated extreme point can be overlooked or skipped over as one moves from one set of weights to another. A weight set might yield a solution that is near an extreme point, but is not an extreme point. Therefore, a linear weighting scheme can only be said to approximate the nondominated set.

This approximation to the nondominated set is probably sufficient. If the size of the entire nondominated set is large, a DM is faced with recognizing subtle differences in trade-offs between close solutions, which could be difficult. Plus, the generation of all of the nondominated solutions could take costly computational time. There is a trade-off in the number of solutions desireable for a DM to adequately explore the efficient frontier and the computational time it takes to produce these solutions.

The weights were systematically varied for each objective function beginning at zero and ending at one in increments of .05.

Each solution generated by the algorithms is validated for non-dominance and if it is non-dominated, it is retained. The retained solution set will determine the "ideal" point, or that point which minimizes all objective functions. This point is established as a vector of all respective minimal values of $F_l(x)$ individually attainable over the feasible set X.

$$F^*_l = (F^*_1, \ldots, F^*_l)$$

where $F^*_l = \mathrm{Min} F_l(x)$, $l=1,\ldots,q$

The ideal point is rarely feasible for the set of objectives. The anti-ideal point can also be calculated for each objective function. The ideal point for each objective would be the minimal value for that objective function, or the point within the feasible region at which the distance is minimized to the ideal point. The anti-ideal is also determined from the nondominated solution set and is the mirror image of the ideal. This point is established as a vector of all respective maximal values of $F_k(x)$ individually attainable over the feasible set X.

$$F_{*l} = (F_{*1}, \ldots, F_{*l})$$

where $F*_l = \mathrm{Max} F_l(x)$, $l=1,\ldots,q$

It represents the point which is the least preferred in relation to all remaining values. The anti-ideal for each objective function is the worst possible, or maximal value that is obtainable for that objective function, in conjunction with values for other objective functions. These two reference points identify the ranges for change for each criteria. The decision maker begins at the solution that is closest, in a distance measure, from the ideal and then iteratively explores the limits and trade-off behavior for the particular network problem at hand.

Solution Algorithms

Both a minimal spanning tree algorithm and an all pairs shortest path algorithm are used to solve the problem. The rationale behind the two approaches is due to the difference in the nature of the objective functions. The first two objective functions seek to minimize the total setup cost and the total operating cost, while the third objective function seeks to minimize the transfer time between any pair of nodes. The minimal spanning tree solution methodology is well suited for generating solutions for total setup and operating cost, but is not well suited for minimizing the transfer time between any two nodes. The all pairs shortest path algorithm, however, is a better methodology for minimizing the transfer time objective. In the search for non-dominated solutions, it was felt that a combination of the two methodologies would provide the DM with more complete information about extreme point solutions and the trade-offs between the objectives.

The minimal spanning tree implementation is a modification to Prim-Dijkstra's minimal spanning tree algorithm (Aho, Hopcroft, and Ullman, 1983). The modification is to accommodate the duality constraint. The duality constraint forces a cycle in the network, which the minimal spanning tree algorithm does not usually allow. The minimal spanning tree algorithm is implemented as a classic Prim-Dijkstra algorithm. Any singly connected nodes are identified. A comparison is made between connecting two singly connected nodes directly and the least expensive way of connecting the two to other nodes already in the tree. The least expensive alternative is selected and the arc(s) are added to the tree.

The shortest paths between all pairs of nodes implementation uses a modification to Tabourier's modification of the Danzig All Pairs Shortest Path algorithm (Kelton and Law, 1978). The modification is again to verify that the duality constraint is met, and if it is not met, to connect any singly connected nodes in the most cost-effective manner, as described above.

The nondominated solution set contains solutions from both algorithms. As each solution is produced from either algorithms, it is checked against all nondominated solutions currently in the nondominated set. If this solution is not dominated by another solution, it is added to the nondominated set. The ideal point and the anti-ideal point is computed from the nondominated set and the minimum distance from the ideal is calculated for each solution in the set.

Minimal Spanning Tree Algorithm

The minimal spanning tree algorithm used in the model is a modification of Prim-Dijkstra's Minimal Spanning Tree Algorithm. This method starts with an arbitrary node and then adds nodes by successively adding a minimally weighted outgoing arc. The arbitrary initial node selection was based on the node-pair with the smallest arc weight. The algorithm terminates in N-1 iterations. A further modification was necessary to ensure that the nodes were dually connected.

Summary of Notation

Let w_l the parametric weight for objective function $_l$ coefficients

n the number of connected nodes

α the number of singly connected nodes

Γ_{ijk} the arc switch indicating candidacy for adding to the tree (0 = candidate, 1 = not a candidate)

τ^S the total setup cost for the tree

τ^O the total operating cost for the tree

τ^T the maximum transfer time between any pair of nodes in the tree

n_d the number of nondominated solutions

τ^S_g the total setup cost for a nondominated solution

τ^{o}_{g} the total operating cost for a nondominated solution

τ^{t}_{g} the maximum transfer time for a nondominated solution

$\Phi_{h,g}$ the arcs in a nondominated tree.

Ω_{ij} the arcs in the current tree

Ω^{s}_{f} the setup cost for arcs in the current tree

Ω^{o}_{f} the operating cost for arcs in the current tree

Ω^{t}_{f} the transfer time for arcs in the current tree

Modified Minimal Spanning Tree Algorithm.

Initialization

Step 0. Set n_d to 0.

Step 1. (Iteration initialization) Initialize Γ_{ijk} to 0 for $i = 1,\ldots,n$; $j = 1,\ldots,n$; $i \neq j$; $k = 1,\ldots,q$.

Step 2. (Individual arc feasibility) If $o_{ijk} > O$ or $s_{ijk} > U$ or $r_{ijk} > S$ then $\Gamma_{ijk} = 1$ for $i = 1,\ldots,n$; $j = 1,\ldots,n$; $i \neq j$; $k = 1,\ldots,q$.

Step 3. (Apply weights to objective function and sum) $d_{ijk} = w_1 * o_{ijk} + w_2 * s_{ijk} + w_3 * r_{ijk}$ for $i = 1,\ldots,n$; $j = 1,\ldots,n$; $i \neq j$; $k = 1,\ldots,q$

Prim-Dijkstra's initialization

Step 4. Sort the d_{ijk}'s into ascending order.

Step 5. Find first d_{ijk}; $\Gamma_{ijk} = 1$; Set $n = 1$.

Node Scan

Step 6. Locate next minimum d_{ijk} connected to any node already in the tree.

Step 7. If the addition of this d_{ijk} creates a cycle in the tree, $\Gamma_{ijk} = 1$; go to Step 6. Otherwise go to Step 8.

Step 8. Add the d_{ijk} to the tree. $\Gamma_{ijk} = 1$. Increment n by 1.

Step 9. If $n > N-1$, Go to Step 10. Otherwise, go to Step 6.

Dually Connected Constraint Verification

Step 10. (count singly connected nodes) $\alpha = 0$. If $node_e$ is singly connected, add 1 to α, for e = 1,...n.

Step 11. (connect singly connected nodes)
Case a. $\alpha = 0$. go to Step 12.
Case b. $\alpha = 1$. Locate the minimum d_{ijk} containing the singly connected node and add it to the tree. $\Gamma_{ijk} = 1$. Go to Step 12;
Case c. $\alpha > 1$. for each pair of singly connected nodes, locate the minimum d_{ijk} containing the two nodes, node i and node j (directly connecting); locate the minimum arc connecting node i to a node already in the tree (d_{iak}) and the minimum arc connecting node j to a node already in the tree (d_{bjk}); select $\min[d_{ijk}, d_{iak} + d_{bjk}]$; put this term on a temporary scan list; for e = 1,...,α. Locate smallest term on temporary scan list. If it is a direct connection, add the d_{ijk} to the tree; $\Gamma_{ijk} = 1$; $n = n + 1$; Go to step 10. If it is a dual connection, add d_{iak} and d_{bjk} to tree; $\Gamma_{iak} = 1$ and $\Gamma_{bjk} = 1$; $n = n + 2$; Go to step 10.

Feasibility Verification

Step 12. (Tree total costs and maximum transfer time) $\tau^S, \tau^O, \tau^T = 0$. $\tau^S = \tau^S + \Omega^S_{ii}$; $\tau^T = \tau^T + \Omega^O_e$ for e = 1,...,n. Longest Transfer Time = Σr_{ijk} of arcs in the longest path in the tree as defined by a depth-first search of the modified tree. (Aho, et al, 1983)

Step 13. (Tree Feasibility) If $\tau^S > U$ or $\tau^O > O$ or $\tau^T > S$, go to step 15.

Nondominated Check

Step 14. If $\tau^S > T^S_e$ AND $\tau^O > T^O_e$ AND $\tau^T > T^T_e$ for e = 1,...,n_d then go to Step 15. $n_d = n_d + 1$; $T^S_{nd} = \tau^S$; $T^O_{nd} = \tau^O$; $T^T_{nd} = \tau^T$; $\Phi_{nd,g} = \Omega_g$ for g = 1,...n.

Termination

Step 15. If additional weights are to be applied, go to step 1, else stop.

All Pairs Shortest Path Algorithm

The all pairs shortest path algorithm used in the model is a modification of Tabourier's modification to Dantzig's

All Pairs Shortest Path Algorithm. This algorithm was selected for implementation based on Kelton and Law's (1983) results of comparing the Floyd, Dantzig, and Tabourier's modification of the Dantzig algorithm, which found Tabourier's modification to be, in general, the most efficient of the three. The modification in the model was to verify that all nodes were dually connected.

The weights were systematically varied for each objective function beginning at zero and ending at one in increments of .05.

Summary of Notation:

$v_i(j,k)$ the total value of the shortest path from node j to node k using only nodes 1,...,i as intermediate nodes.

$E_i(j,k)$ the total travel time of the shortest path from node j to node k using only nodes 1,...,i as intermediate nodes

$y_i(j,k)$ the node immediately following node j on the shortest path from node j to node k using only nodes 1,...,i as intermediate nodes. $y_N(j,k)$ represents the optimal policy for all pairs of nodes.

$t^l_i(j,k)$ the temporary distances used in analysis.

All-Pairs Shortest Path Algorithm.

Iteration Initialization

Step 0. (Apply weights to objective function and sum) $d_{ijk} = w_1*o_{ijk} + w_2*s_{ijk} + w_3*r_{ijk}$ for $i = 1,\ldots,n;$ $j = 1,\ldots,n;$ $i \neq j;$ $k = 1,\ldots,q$

Step 1. For each pair of nodes, select the arc (communications media) with the minimum d_{ijk} and place in the distance matrix.

Dantzig's Algorithm Initialization

Step 2. Set $v_1(1,1)$ to 0; $y_1(1,1)$ to 1; $i = 2$

Step 3. Set $t^0{}_i(j,i)$ to ∞ and $y_i(j,i) = y_{i-1}(j,1)$, for $j = 1,\ldots,i-1$; set $t^0{}_i(i,k) = \infty$ and $y_i(i,k) = 1$ for $k = 1,\ldots,i-1$; set $l = 1$.

Tabourier's Modification

Step 4. (check arc for infinity) If $w_{li} \geq t^{l-1}{}_i(l,i)$, then set current distance to previous step's distance $t^l{}_i(j,i) = t^{l-1}{}_i(j,i)$, for $j = 1,\ldots,i-1$; go to Step 5. Otherwise, set $t^{l-1}{}_i(j,i)$ to $\min[t^{l-1}{}_i(j,i), v_{i-1}(j,l) + w_{li}]$. If second term minimizes, and l not = j then set $y_i(j,i) = y_{i-1}(j,l)$, else set $y_i(j,i) = i$; for $j = 1,\ldots,i-1$

Step 5. (check arc to see if improvement possible, and if not, adopt previous iteration's distance) If $w_{il} \geq t^{l-1}{}_i(i,l)$ then $t^l{}_i(i,k) = t^{l-1}{}_i(i,k)$; for $k = 1,\ldots,i-1$; go to Step 6. Otherwise, set $t^l{}_i(i,k) = \min[t^{l-1}{}_i(i,k), w_{il} + v_{i-1}(l,k)]$ and $y_i(i,k) = l$ if the second member minimizes.

Step 6. If $l = i-1$ then set $v_i(j,i) = t^{i-1}_{i-1}{}_i(j,i)$, for $j = 1,\ldots,i-1$ and set $v_i(i,k) = t^{i-1}{}_i(i,k)$, for $k = 1,\ldots,i-1$; go to Step 7. Otherwise, set $l = l + 1$; go to Step 4.

Remainder of Dantzig's Algorithm.

Step 7. (Check for negative cycle) Compute sum $v_i(i,l) + v_i(l,i)$ for $l = 1,\ldots,i-1$. If sum is negative then stop. Otherwise, set $v_i(i,i) = 0$ and $y_i(i,i) = i$; go to Step 8.

Step 8. Set $v_i(j,k) = \min[v_{i-1}(j,k), v_i(j,i) + v_i(i,k)]$ and $y_i(j,k) = y_{i-1}(j,k)$ if first member minimizes, otherwise $y_i(j,k) = y_i(j,i)$.

Step 9. (All Pairs Algorithm check for termination) If $i = N$ then go to Step 10. Else $i = i + 1$; go to Step 3.

Dually Connected Constraint Verification

Step 10, 11. Refer to step 10 and 11 of minimal spanning tree algorithm.

Feasibility Verification

Step 12. (Paths total costs and maximum transfer time) $\tau^S, \tau^O = 0$. $\tau^S = \tau^S + \Omega^S_e$; $\tau^O = \tau^O + \Omega^O_e$; for e = 1,...,n. $\tau^T = \max[T_N(j,k)$; for j = 1,...,N; for k = 1,...,N]

Step 13. (Solution Feasibility) If $\tau^S > U$ or $\tau^O > O$ or $\tau^T > S$, go to step 15.

Nondominated Check

Step 14. (Solution Nondominated) If $\tau^S > T^S_e$ AND $\tau^O > T^O_e$ AND $\tau^T > T^T_e$ for e = 1,...,nd then go to Step 15. $n_d = n_d + 1$; $T^S_{nd} = \tau^S$; $T^O_{nd} = \tau^O$; $T^T_{nd} = \tau^T$; $\Phi_{nd,g} = \Omega_g$ for g = 1,...n.

Termination

Step 15. If additional weights are to be applied, go to step 1, else stop.

CHAPTER IV

DECISION SUPPORT SYSTEM IMPLEMENTATION

The general system design philosophy was to develop an interactive, menu-driven, graphically animated decision support system which is user-friendly and enables the DM to understand the problem and its solutions. The DSS contains modules for relational data management, structured model management, and user-interface management (Jarke, Jelassi, and Stohr, 1984; Jelassi, Jarke, and Stohr, 1985; Jarke, Jelassi, and Shakun, 1987; Dos Santos and Bariff, 1988). (See Figure 3) The system incorporates visual interactive modelling techniques from the model specification stage through the results display stage (Hurrion, 1986; Billington, 1987). Where possible, user instructions are through a mouse, and where necessary, user-interaction is through the keyboard. (See Figure 4) (Note: Exhibits 1 through 8 in Appendix B describe the detail of the data flows and the screens used in the system)

The decision maker interactively defines the problem and the communications components to be evaluated from screen forms. The problem data is stored in a relational

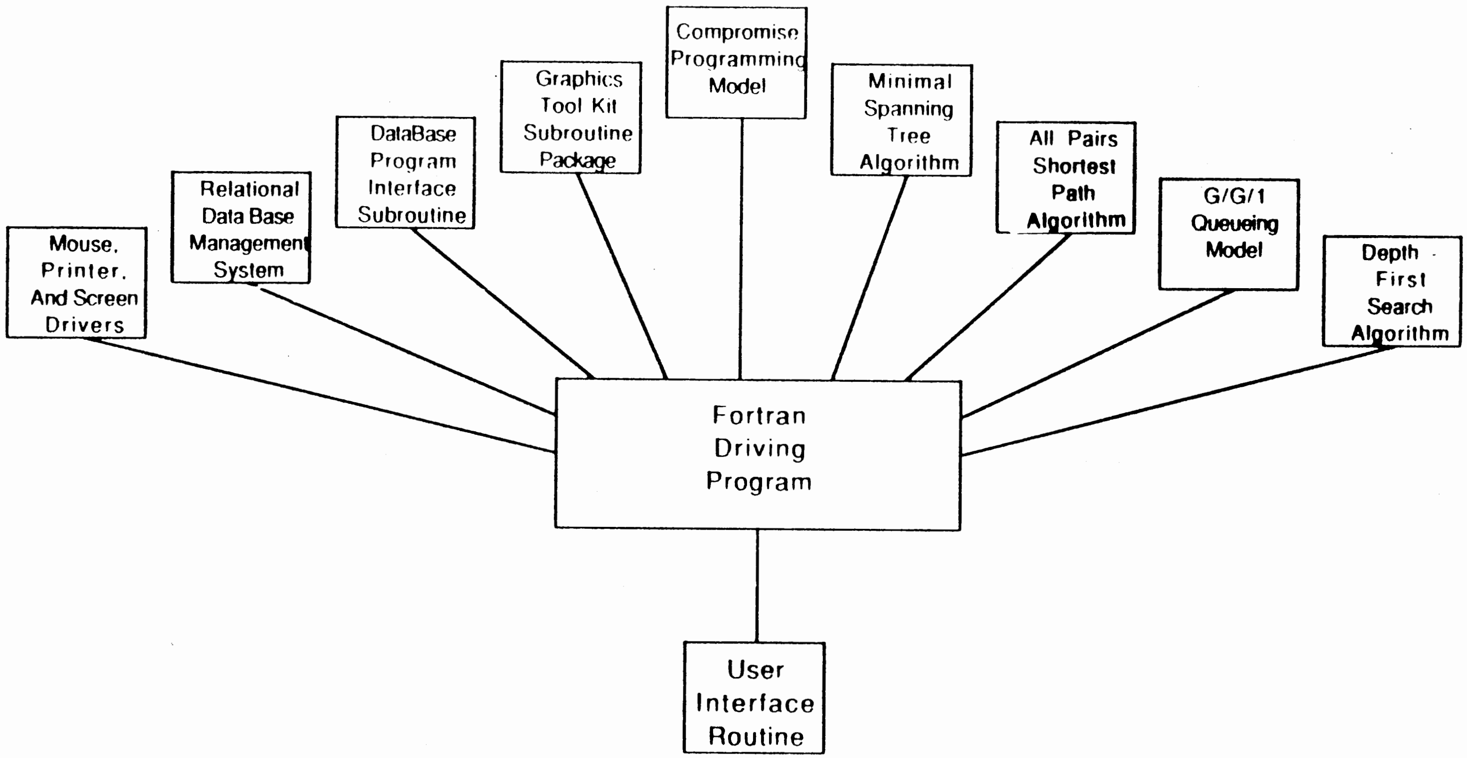


Figure 3. Tools Used in the Decision Support System

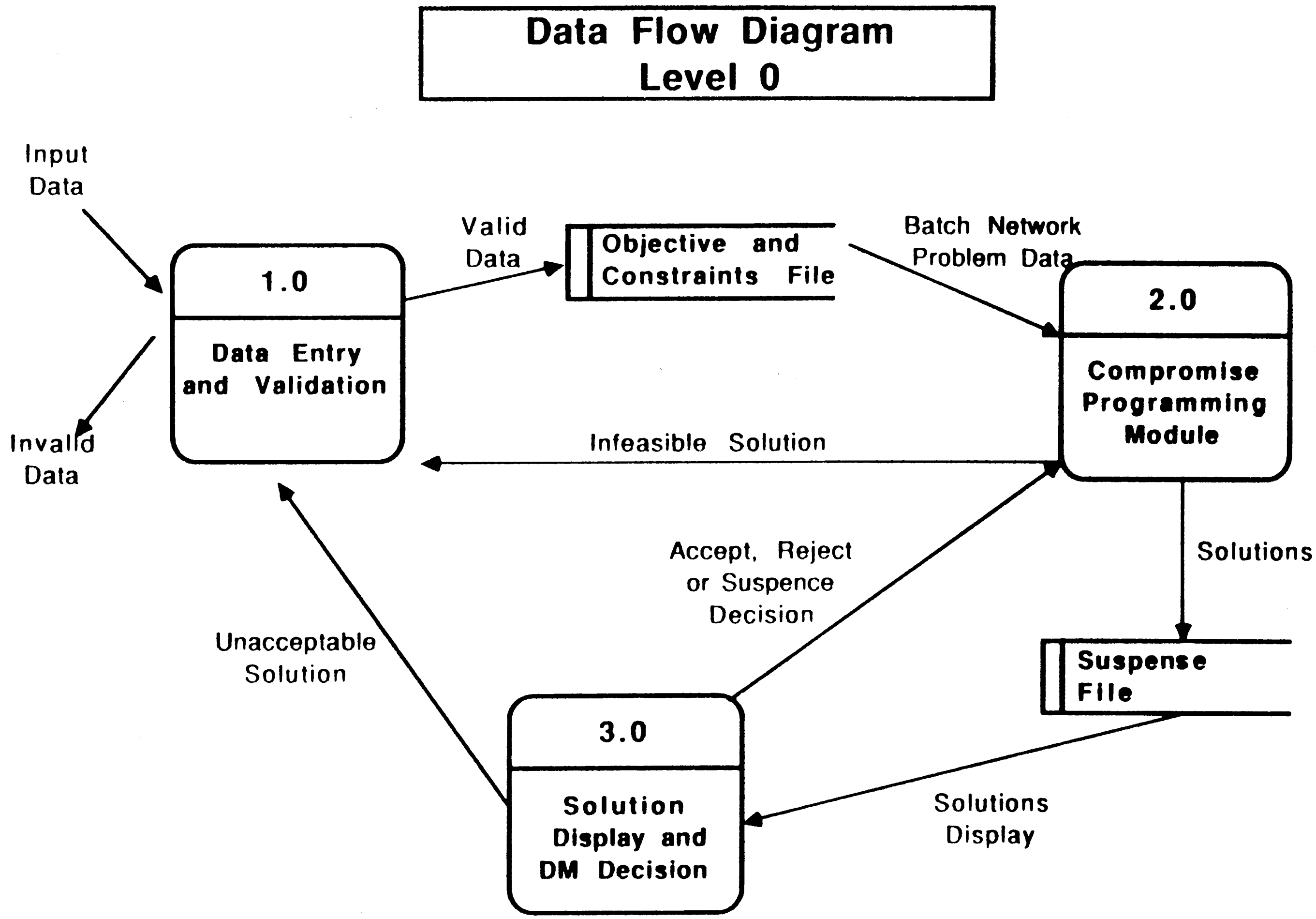


Figure 4. Context Diagram

data base for subsequent use by the user. A menu choice from the Compromise Program Setup Menu allows the user alternative options for: full model specification for a new problem; problem modification for an existing case or the generation of a new case that may be modified; the reuse of an existing problem for a network analysis; and deletion of an entire problem. (See Exhibit 9 in Appendix B)

The user may update data in all data tables by selection of updating choices from the Compromise Programming Update Menu (See Exhibit 10 in Appendix B). New data may also be added to any table except the general setup table through ADD choices from this menu.

The full problem showing node locations and all possible communications media connections between nodes is drawn on the screen. While the problem is being solved using linear combinations of the three objective functions by the minimal spanning tree algorithm, a blinking message 'COMPUTING' is displayed on the screen. The message switches to a blinking 'RECOMPUTING' when the all pairs shortest path algorithm is solving the problem. If the problem is infeasible, the user is notified and he or she must modify the original data and solve the problem again. Otherwise, the ideal point is calculated. If the ideal point is feasible, the solution will be graphically displayed, labeled as optimal, stored, and the process will cease. If the ideal point is not feasible, the solution with the minimum distance from the ideal will be displayed with the level of achievement of each objective graphically displayed. The user can then begin the iterative process of viewing other solutions and the trade offs in levels of achievement of the objectives with alternative solutions.

Data Input and Validation

This DSS is a menu driven system that first displays the problem Setup Menu. This menu allows the DM to elect to either use an existing problem, update and use an existing problem (either as a new case or as strictly an updated problem), enter a new problem, or delete a problem.

A new problem requires the user to interactively enter all of the data required by the model. This includes five tables of data: the general problem setup data; the node specific data, the communications media specific data, internode file sizes and traffic patterns, and internode setup costs and operating costs for each communications media.

A screen form for entering data into each of the tables was developed using the Graphics Development Toolkit subroutine package. The routines allow for data entry sequentially, from top to bottom, and left to right. The field for each data item to be entered is displayed in reverse-video. This serves to call attention to the data item to enter as well as indicating the maximum size of the field. The user is asked whether there is additional data to be entered after each form is completed, except when entering the general problem setup data. There is only one occurrence of general setup data. A partial problem can thus be entered and at another setting, through the update routines, the data entry process can be completed.

The user begins by defining the general problem setup: the problem name, number, and a case identifier; the number of nodes in the network; the number of types of communications media possible; the minimum acceptable service level between any two nodes; the maximum acceptable total setup cost; and the maximum acceptable total operating cost. The DSS calculates and displays the maximum number of interconnections between pairs of nodes. The number of objective functions is fixed and displayed as three. (See Exhibit 11 in Appendix B).

Each computer node is specified by an identifier, a name, and its location in terms of latitude and longitude. The latitude and longitude are used to map the node into a 30 x 30 space (the full screen is 100 x 100) in the upper right corner of the screen. There is no restriction on the size of the latitude and longitude. The mapping process first determines the range of the latitude and longitude values and scales the drawing accordingly. The name of a computer node is a maximum of eight characters and appears on the map drawing above the node. (See Figure 12 in Appendix B).

The communication media data includes the communications identifier, name, and the maximum possible transfer rate. The communications name is printed next to its assigned color on the color code legend. (See Exhibit 13 in Appendix B).

Two types of internode data are kept for each problem: static internode data-the distance, the traffic patterns (pessimistic, optimistic, and most likely), and the file sizes (pessimistic, optimistic, and most likely) between a pair of nodes; and communications media dependent data-the setup cost and the operating cost for a given communications media between the nodes. (See Exhibit 14 in Appendix B).

The DSS requires that internode data for all communications media between all pairs of nodes be present and does not allow for the prespecification of a link between nodes. The solution algorithms would require major modifications to accommodate the fixing of a particular link between nodes.

The setup costs and the operating costs are used as the objective function coefficients for the setup and operating objective functions respectively. The effective transfer times are calculated from the three estimates of file sizes and the three estimates of traffic patterns and the transfer rate for the given communications media. These times are used as objective function coefficients for the service time objective function. A user may use an existing problem by selecting this option and then may elect to retrieve solutions from a previous session, or to solve the problem and generate all nondominated solutions. If solutions are retrieved from a previous session, only the nondominated solutions that were saved from that session are retrieved. These are displayed to the user and can be used for studying trade offs and/or printed for further analysis. If a new solution is elected by the user, the problem is solved.

A user may elect to update an existing problem and is given a choice of generating a new case from the old problem or updating and retaining the problem and its current case. If a new case is requested, the problem data is copied to the data base with its new case identification. For a new or existing case, the user is presented with an update selection menu that will allow any of the data stored in the data base to be updated and/or new data to be added.

The user selects the data group that he/she wishes to update or add new data to, the data is retrieved, and displayed on the form for that data group. The data item to be updated is chosen by use of a mouse. The mouse is left-clicked on the data item to be updated, the field is displayed in reverse-video, and the user enters the update from the keyboard. The user indicates that he/she has completed updates for a data group by a right-click on the mouse. The user is returned to update selection menu where they can select another data group to update or add new data to, or exit from the update process. The problem is automatically solved when the user exits from the update process.

A problem can be deleted from the data base by selecting a delete problem menu option. The user is given one opportunity to change their mind and if their response is yes they wish to delete the problem, the problem is deleted from the data base.

The user exits the entire program by selecting the exit option from the Compromise Programming Setup menu.

The Compromise Programming Process

The compromise programming module assigns a weight to each of the objective functions, multiplies the coefficients of each objective function by its assigned weight, sums the three objective functions to form a single-objective function, and solves the problem. This weighting method approach generates the optimal solution for each of the individual objective functions, which establishes the nondominated extreme point solutions. The nondominated solution set is generated from these extreme points.

The weights selected for use in the model sum to 1 and are parametrically varied between 0 and 1 for each of the objective functions, in increments of .05. The weights are applied to the coefficients in order to generate multiple solutions which form an approximation of the nondominated solution set. The step size was selected as a good balance between the number of solutions generated and the computation time.

After the nondominated solution set is identified, the ideal point and the anti-ideal point are calculated for each objective function. The distance from the ideal for each nondominated solution is calculated as:

$$L(x) = \sum_{l} \alpha_l \frac{F_l^* - F_l(x)}{F_l^* - F_{lw}}$$

where α_l are the normalizing scaling factors applied to each objective: F_l^* is the optimum (ideal) value of the lth criterion; F_{lw} is the worst (anti-ideal) value obtainable for the kth criterion; and $F_l(x)$ is the result of implementing decision x with respect to the lth criterion. The L_δ-metric thus represents a weighted distance rather than an absolute distance. This serves to scale the objectives.

Graphic Solutions and User Interface

The system displays the ideal and the anti-ideal for each objective function and their achievement levels for a nondominated solution with the minimum distance from the ideal point. The DM can then begin the trade-off study and analysis by selecting the direction of improvement with respect to one or more of the objective functions.

The potentials between the ideal and the anti-ideal points for each objective function are displayed as a unit distance measure in bar graph form. The direction of improvements proceeds from the top to the bottom of each bar. The top of each of the bars represents the anti-ideal point for that objective function. A solution at this level has the potential for a 100% improvement. Whereas, the bottom of each bar represents the ideal point. A solution at this level has a 0% potential for improvement.

Each objective function is labeled by name and assigned a different bar color. The current solution level for each function is displayed as a dark shade of the color, while the unrealized portion of the objective is displayed as a light shade of the color. The actual amounts for the ideal point, the anti-ideal points, and the current achievement level are displayed at their achievement level (see Figure 5).

The network topology of the current solution is presented in the top right hand corner of the screen. The latitude and longitude coordinates for each node are mapped into this space and the nodes are drawn as circles and labeled by computer node name. The communications media selected by the model for connecting the nodes in the current solution are drawn as color-coded lines between the nodes and the color legend and communications media names

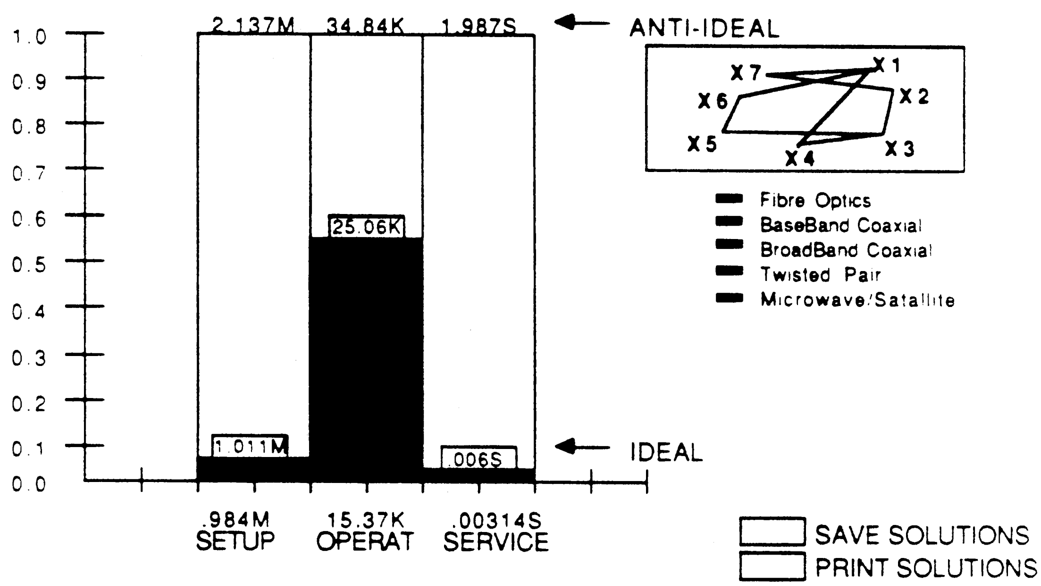


Figure 5. Final Solution Screen

are printed below the map rendering.

The decision maker is initially shown the solution with the minimum distance from the ideal and then has the choice of exploiting available potentials fully, objective by objective, or investigating the trade-offs and limits imposed by the interaction of two or more of the objectives. The desired level of one or more of the objective functions is indicated by a left click of the mouse input device and the DM indicates that their selection is finished by the right click of the mouse.

From the achievement levels indicated by the DM for one or more of the objective functions, a solution is identified from the nondominated solution set. The selection criteria for the next solution is the nondominated solution with the minimal distance to the DM's desired level for that objective function(s). A new bar chart and map rendering for this solution is then presented. The iterative process continues until the decision maker concludes the session by a right click on the mouse that was not preceded by a left click. The DM can elect to save and/or print any solution by positioning the mouse in the save box or the print box and clicking left.

If none of the compromise solutions is found to be acceptable, the decision maker must redefine the problem.

Support Tools

The DSS used a variety of microcomputer-based software packages that were bound together with a driving program that was written in Microsoft's Fortran programming language.

Interactive data entry and solution files used Microrim's R:Base™ 5000 relational data base system and the R:Base™ PI (Program Interface) subroutine package to interface with the driving program. The database structure and all of the data tables were originally defined in R:Base 5000. Forms for data entry and update could have been defined in R:Base 5000, but in black and white only. Therefore the forms were created in the graphics package. All of the physical data record and file management needs were met with subroutines supplied in the PI package.

Data entry and update forms and graphical displays were generated using IBM's Graphics Development Toolkit subroutine package. The subroutines contained primitives for drawing lines, circles, and bars, and for displaying text. Color, position, and size attributes could be assigned to any item displayed.

The program was written for an IBM/AT microcomputer with 640K bytes of internal memory and requires approximately 500K of available memory. The program makes use of an IBM enhanced graphics adapter monitor, a Microsoft mouse with a parallel interface, and an IBM Proprinter.

CHAPTER V

A TEST OF THE DECISION SUPPORT SYSTEM

The Decision Support System was tested by having forty participants design one computer network by manual means and another network using the DSS. The participants included a combination of students, academicians, businessmen, engineers, and non-professionals. The designer's backgrounds encompassed a mix of experienced and inexperienced network designers, computer network designers, computer literates, and managers. The designers were randomly assigned to one of two groups which are referred to as Group 1 and Group 2. (See Table 1)

Both groups first designed a computer network composed of five computer nodes that could be connected by three communications media links. Group 1 designed the network manually and Group 2 used the DSS in the design process. Both groups then designed a second network composed of four computer nodes with four communications media links. Group 1 used the DSS in designing this network and Group 2 designed the network manually. Thus Group 1 used DSS second whereas Group 2 used the DSS first. This design was used to minimize the effect of sequencing and learning about the problem.

TABLE 1

EXPERIMENTAL DESIGN

Group	5 Node Design	4 Node Design
Group 1	Manual	DSS
Group 2	DSS	Manual

All subjects first were introduced to the general nature of the problem and the multiobjective decision criteria. Subjects were told there was not a single, optimum solution to the problem, but several solutions were plausible, depending upon their personal preference structure concerning the multiobjectives. The participants were given specific instructions and a practice session prior to each design phase.

Both groups answered two specific questionnaires addressing their reaction to and their results from the manual design and the DSS supported design. A brief, general background demographic questionnaire designed to identify the participants network design and optimization skills was included in one of the questionnaires. Both questionnaires used a seven point Likert-type scale (1=strongly disagree to 7=strongly agree). The thirty-point decision support system questionnaire contained all of the questions from the twenty-point manual design questionnaire, plus decision support system-specific questions. Both questionnaires were designed to isolate subjects attitudes toward the design process enhancing their problem-solving skills, their confidence in decision quality, and perceptions of the process in arriving at a decision.

The hypotheses were designed to attempt to quantify the subjective qualities of a decision support system. The main issue was whether a decision support system did indeed aid in improving decision making. The issue was compounded by the fact that in a multiobjective decision making environment there is no single best answer; rather many solutions exist from which a decision maker must select the best alternative according to his or her own personal preferences.

The first group of hypotheses tested were related to the costs involved in a computer network: the initial building, or setup cost; the operating cost; and the transfer time (a time cost). As has been stated, the minimization of these costs are inherently in conflict. In order to minimize one, another cost must be sacrificed. Therefore, these hypotheses are an attempt to measure which cost the designers were valuing highest during each of the design processes.

H_S: Setup Cost of $\text{Design}_{\text{DSS User}} \leq$ Setup Cost of $\text{Design}_{\text{Manual}}$

H_0:Operating Cost of Design$_{DSS\ User}$≤Operating Cost of Design$_{Manual}$

The next hypothesis measures the number of decision alternatives viewed. It has been suggested that the decision caliber is increased when more alternatives are examined. (Sharda, Barr, and McDonnell, 1988) This hypothesis examines the extent to which this phenomenon was present in the study.

H_A:# Alternatives Viewed$_{Manual}$≤# Alternatives View$_{DSS\ User}$

The final hypothesis attempts to measure the decision confidence level present when using a decision support system. Sharda, Barr, and McDonnell (1988) report mixed results from studies examining DSS use and user's attitudes such as confidence. Their study investigating group decision confidence showed, for the most part, a higher reported level of confidence, but not statistically significantly higher confidence levels between the DSS users and the non-DSS user.

H_F:Confidence in Design$_{Manual}$≤Confidence in Design$_{DSS\ User}$

CHAPTER VI

RESULTS AND DISCUSSION

The participants overall perception of the design problem was that of being difficult to solve. The majority had never designed a network of any type and most had never used optimization techniques. Only a few were familiar with multiobjective analysis tools.

The network designers using the DSS selected and printed their preferred solution after reviewing alternatives from the nondominated solution set produced by the system. The manual designers performed their design with the aid of a calculator and recorded their design on a worksheet. Both designs yielded a setup cost and an operating cost. Only a few manual designers performed the calculations for the maximum transfer time between any pair of nodes. Instead, the majority considered only very fast communications media and assumed that the maximum allowable transfer time constraint would be met.

The average setup cost for the computer network designed manually was lower than the average setup cost for the computer network designed with the aid of the DSS. And, the average operating cost for the manually designed network was higher than the average operating cost for the DSS user.

This apparently conflicting result could have several plausible explanations. The units of measure may have influenced the designers. The setup cost was expressed in hundreds of thousands of dollars and the operating cost was expressed in thousands of dollars for both designs. The size of the numbers may have swayed manual designers to place greater emphasis on the setup cost. (See Table 2)

The DSS displays the three objective functions as a unit distance from the ideal over the range between the ideal and anti-ideal, which visually equalizes the unit disparities. This may have swayed designers to try to select a solution where all three objective function values were approximately equal. This was true of the most popular solution. Another factor may have been that the solution selected by the system for both the four and five node problem with the minimum distance from the ideal was a solution that minimized operating cost. This may have prejudiced DSS designers towards lower operating cost solutions.

The number of alternatives reported viewed was significantly higher for the DSS user than the manual designer. An interesting phenomenon was that some DSS designers reported viewing more solutions than there actually were. There are five nondominated solutions for the four node design problem and nine for the five node problem and the average number of solutions viewed was over seven. (See Table 3)

TABLE 2

COMPARISON OF THE COSTS OF MANUAL VERSUS DSS DESIGNS

	4 Node Network		5 Node Network	
	μ	(σ)	μ	(σ)
SETUP COST				
Manual	868684.21	(216976.97)	1004736.82	(320858.20)
DSS	927100.00	(106181.03)	1110894.74	(221041.70)
OPERATING COST				
Manual	10807.37	(6868.52)	14123.68	(4087.26)
DSS	6410.00	(1884.40)	8010.53	(1629.81)

TABLE 3

COMPARISON OF THE NUMBER OF ALTERNATIVES VIEWED OF MANUAL VERSUS DSS DESIGNS

Group	Manual		DSS		Significance Level of the Difference
	μ	(σ)	μ	(σ)	
Group 1	2.737	(1.147)	7.118	(4.608)	.000
Group 2	2.842	(1.119)	9.105	(6.674)	.000
Both	2.857	(1.115)	7.971	(5.762)	.000
Significance Level of the Difference	n.s.		n.s.		

The two groups reported a neutral attitude concerning the statement "I'm not confident about my solution" for the manually designed system and slightly disagreed with the statement for the DSS aided design, with no statistical significance between groups for either question. There was, however, a statistically significant difference between the answer given for this question after the manual design and the answer given after the DSS aided design. Both groups expressed more confidence in the solution for the DSS aided design. (See Table 4)

TABLE 4

COMPARISON OF CONFIDENCE IN SOLUTIONS OF MANUAL VERSUS DSS*

Group	Manual μ	Manual (σ)	DSS μ	DSS (σ)	Significance Level of the Difference
Group 1	4.1	(1.65)	3.05	(1.43)	.014
Group 2	4.0	(2.05)	3.211	(1.99)	n.s.
Both	4.025	(1.82)	3.132	(1.71)	.006
Significance Level of the Difference	n.s.		n.s.		

*A lower number indicates higher confidence

CHAPTER VII

RECOMMENDATIONS FOR FURTHER STUDY

The results of this study generally support the concept of a decision support system aiding in the decision-making process. The number of alternatives reviewed was several times more for the DSS users than the non-DSS users. The confidence in the preferred solution was significantly higher for the DSS user. The operating costs, on average, were lower for the DSS user, but not significantly lower. The setup costs, on average, were higher for the DSS user, but not significantly higher. The last two results, as discussed, may have been due to the format of graphical displays.

The test of the decision support system gave some indication of presentation bias that could be more fully explored. Many participants leaned toward a nondominated solution where the three objective achievement levels were visually equal, disregarding the difference in the units involved in individual objective functions. The results presentation screen could be modified to present results in a normal scale instead of a unit distance format to determine if the bias does exist.

There was evidence that the group which used the decision support system first felt more positively about the problem and the approach to solving the problem than the group which initially designed the manual system. A possible explanation might be that the group which first used the DSS first better understood the problem and strategies for solving the problem than the group which first manually designed the network. If this phenomenon could be isolated and proven, it might indicate that a strong use for decision support systems could be for training purposes. A different experimental design could be employed to study the strength, if any, of this learning effect.

As previously stated, a weakness in our DSS is the inability to 'fix' a communications media link between a given pair of computer nodes. This modification would require a major modification to both the minimal spanning tree algorithm and the all pairs shortest path algorithm, but it would add to the credibility and the usefulness of the system. Its implementation is planned as one of the first revisions to the system.

The system is presently limited to small networks. It occupies approximately 500K of internal memory and the necessary drivers for the program occupy approximately 124K. This is approaching the DOS limitation of 640K. When the DOS limitation is raised, the program can be expanded to handle larger networks.

The results of the study also produced several new areas that deserve additional attention. The two algorithms that were used to generate the nondominated solutions produced disjoint solutions. The minimal spanning tree algorithm gave a cluster of solutions that tended to be low for the costs, but higher for the time. The opposite was true for the all-pairs algorithm. This could indicate that there are more solutions on the efficient frontier that the algorithms are incapable of generating. One or more additional algorithms could possibly explore the nondominated solution set more fully. (See Figure 6)

This DSS, which was specifically developed for the computer network design problem, could be modified for more general use with other network application areas, such as cash flow management, assignment problems, scheduling and distribution problems, or transportation and transshipment problems. Glover and Klingman (1975) predicted that maybe as high as 70 percent of the real world mathematical programming problems can be stated directly as a network problem or can be converted into a network problem. If their prediction is anywhere close to being accurate, then the potential for modifying the DSS for other optimization problem areas that have multiple objective criteria merits further study.

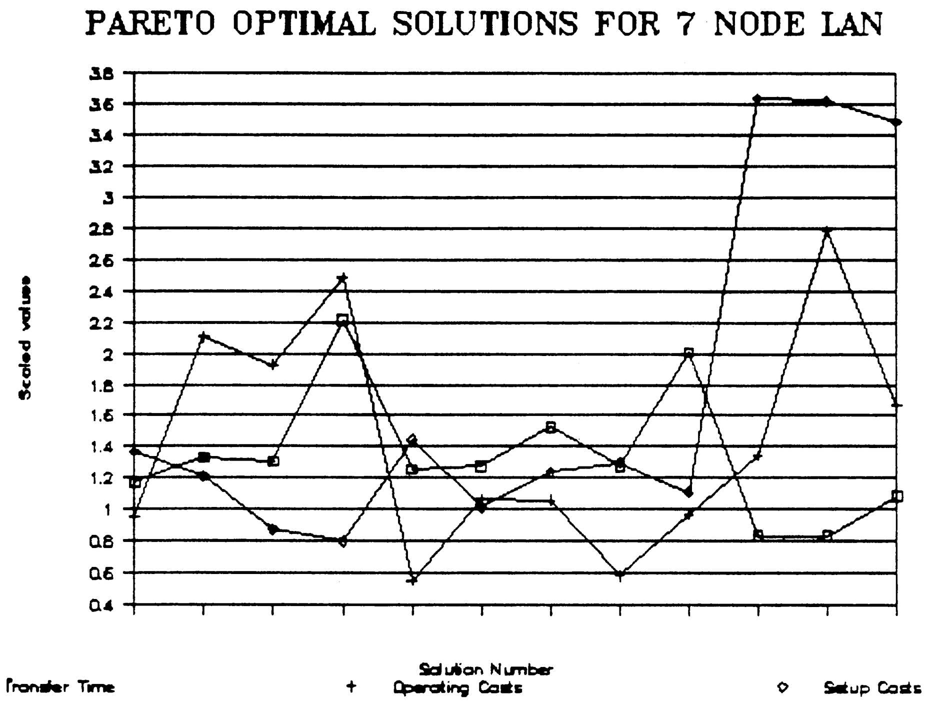


Figure 6. Nondominated Solutions for a Seven Node and Five Communications Media Problem.

CHAPTER VIII

SUMMARY AND CONCLUSIONS

The purpose of this study was to combine the areas of MCDM, design of computer networks, and DSS into a microcomputer-based system and to perform an experimental test to determine if the system was useful to a decision maker. A Compromise Programming approach was used as the multiple criteria decision making tool with the weighted method of generating nondominated solution points from extreme point solutions identified from Minimal Spanning Tree and All Pairs Shortest Path Algorithms. The system was developed for an IBM PC/AT and was written in Fortran IV programming language. The system utilized R:Base 5000 relational database and IBM Graphics Development Toolkit subroutine package.

The system was tested for its usefulness to a decision maker in designing a computer network. The test involved forty people. Each person designed one computer network by manual means and another computer network using the decision support system. A questionnaire addressing their reactions to and their results from the design was completed at the conclusion of each design phase. The specific hypotheses tested related to the costs involved in a computer network, the number of decision alternatives viewed, and the decision maker's confidence in his or her solution. The participants generally perceived the computer network design problem as being difficult to solve.

The average setup cost was lower for the manually designed system than the setup cost for the DSS designed system. The average operating cost was higher for the manually designed system than the operating cost for the DSS designed system. The layout of the final solution screen and the solution choice of the system may have influenced DSS designers to select a lower operating cost solution.

The number of alternatives viewed was significantly higher for the DSS designer than the manual designer. This result was expected, but the reporting of viewing more alternatives than actually existed was surprising.

The DSS designers had slightly more confidence in their solution than the manual designers. The combined group and the group that manually designed the computer network first had a significant improvement in confidence in their solution after the DSS design. There was not a significant difference in the confidence level of the group who designed the computer network with the DSS first after the manual design. This result could be interpreted as a training effect: the group who designed with the DSS first may have understood the problem and a solution strategy better. Further investigation would need to be done to prove this interpretation.

The results of the study pointed out several new directions to investigate: the effect of the solution display screen on designers; the effect of providing an initial solution on designers; and the use of a DSS as a training tool.

The algorithms and methodology employed in the DSS could also be enlarged. Other network algorithms could be incorporated into the system to determine if additional nondominated solutions exist. The system could be expanded to accommodate larger network problems and the methodology could be extended for use with other network problems.

SELECTED BIBLIOGRAPHY

Aggarwal, K. K., V. C. Chopra, and Z. S. Bajwa, Capacity Consideration in Reliability Analysis of Communication Systems, IEEE Transactions on Reliability, Vol. R-31, No. 2, June, 1982

Aho, Alfred V., John E. Hopcroft, and Jeffrey D. Ullman, Data Structures and Algorithms, Addison-Wesley, 1983

Aldag, Ramon J. and Daniel J. Power, An Empirical Assessment of Computer-Assisted Decision Analysis, Decision Sciences, Vol. 17, 1986, pp. 572-588

Ali, A. Iqbal, Ellen P. Allen, Richard S. Barr, and Jeffery L. Kennington, Reoptimization Procedures for Bounded Variable Primal Simplex Network Algorithms, European Journal of Operational Research, Vol. 23, pp. 256-263, 1986

Aneja, Y. P. and K. P. K. Nair, Bicriteria Transportation Problem, Management Science, Vol. 25, No. 1, January, 1979, pp. 73-78

Angell, Ian O., A Practical Introduction to Computer Graphics, Halsted Press, 1982

Arthur, J. L. and A. Ravindran, A Branch and Bound Algorithm with Constraint Partitioning for Integer Goal Programming Problems, European Journal of Operational Research, Vol. 4, No. 6, pp. 421-425, 1980

Aubin, J. P. and B. Naslund, An Exterior Branching Algorithm, Working paper 72-42, European Institute in Management, Brussels, November, 1972

Azad, Fereshteh and Gary W. Dickson, Data Communications Systems: A Managerial Perspective, Working paper, Summer, 1981

Bailey, James E. and Sammy W. Pearson, Development of a Tool for Measuring and Analyzing Computer User Satisfaction, Management Science, Vol. 29, No. 5, May, 1983, pp. 530-545

Bardossy, A. and I. Bogardi, Network Design for the Spatial Estimation of Environmental Variables, Applied Mathematics and Computation, Vol. 12, 1983, pp. 339-365

Baronas, Ann-Marie, K. and Meryl Reis Louis, Restoring a Sense of Control During Implementation: How User Involvement Leads to System Acceptance, MIS Quarterly, March, 1988, pp. 111-124

Beale, E. M. L. Applications of Mathematical Programming Techniques, American Elsevier, 1970

Benayoun, R., J. de Montgolfier, J. Tergny, and O. I. Laritchev, Linear Programming with Multiple Objective Functions: STEP Method (STEM), Mathematical Programming, Vol. 1, No. 3, pp. 366-375, March, 1971

Benson, Harold P. and Thomas L. Morin, A Bicriteria Mathematical Programming Model for Nutrition Planning in Developing Nations, Management Science, Vol. 33, No. 12, December, 1987, pp. 1593-1601

Bertsekas, Dimitri P. and Paul Tseng, Relaxation Methods for Minimum Cost Ordinary and Generalized Network Flow Problems, Operations Research, Vol. 36, No. 1, pp, 93-114

Bertsekas, Dimitri P. and Robert Gallagher, Data Networks, Prentice-Hall, 1987

Billington, J. N., Visual Interactive Modelling and Manpower Planning, European Journal of Operational Research, No. 30, 1987, pp. 77-84

Binbasioglu, Meral and Matthias Jarke, Domain Specific DSS Tools for Knowledge-Based Model Building, Decision Support Systems, Vol. 2, 1986, pp. 213-223

Bitran, Gabriel R., Linear Multiple Objective Programs with Zero-One Variables, Mathematical Programming, Vol. 13, No. 2, 1977, pp. 121-139

Bitran, Gabriel R., Theory and Algorithms for Linear Multiple Objective Programs with Zero-One Variables, Mathematical Programming, Vol. 17, No. 3, 1979, pp. 362-390

Bitran, Gabriel R. and Kenneth D. Lawrence, Locating Service Offices: A Multicriteria Approach, Omega, The Journal of Management Science, Vol. 8, No. 2, pp. 201-206

Bixby, Robert E. and Robert Fourer, Finding Embedded Network Rows in Linear Programs I. Extraction Heuristics, Management Science, Vol. 34, No. 3, March, 1988, pp. 342-376

Black, Uyless D., Data Communications, Networks and Distributed Processing, Reston Publishing Co., 1983

Bogardi, István, András Bárdossy, and Lucien Duckstein, Multicriterion Network Design Using Geostatistics, Water Resources Research, Vol. 21, No. 2, February, 1985, pp. 199-208.

Bokhari, Shahid H., A Shortest Tree Algorithm for Optimal Assignments Across Space and Time in a Distributed Processor System, IEEE Transaction on Software Engineering, November, 1981

Boorstyn, Robert R., and Howard Frank, Large-Scale Network Topological Optimization, IEEE Transactions on Communications, Vol. Com-25, No. 1, January, 1977

Boot, John C. G., Quadratic Programming Algorithms-Anomalies-Applications, North-Holland Publishing Company, 1964

Bowman, V. J. Jr., On the Relationship of the Tchebyscheff Norm and the Efficient Frontier of Multiplecriteria Objectives, Multiple Criteria Decision Making, edited by H. Thiriez and S. Zionts, Lecture Notes in Economics and Mathematical Systems No. 130, Springer-Verlag, Berlin-New York, New York, pp. 76-86, 1976

Brown, Gerald G. and Richard D. McBride, Solving Generalized Networks, Management Science, No. 12, December, 1984

Bui, Tong and Mathais Jarke, Communications Requirements for Group Decision Support systems, Journal of Management Information Systems, Spring, 1986, Vol. 2, No. 4

Burke, Thomas J. M. and Maxwell Lehman, Communication Technologies and Information Flow, Pergamon, 1981

Cats-Baril, William L., and George P. Huber, Decision Support Systems for Ill-Structured Problems: An Empirical Study, Decision Sciences, Vol. 18, 1987, pp. 350-372

Ceri, Stafano and Guiseppe Pelagatti, Distributed Databases Principles & Systems, McGraw-Hill, New York, 1984

Chandler, John, A Multiple Criteria Approach for Evaluating Information Systems, MIS Quarterly, March, 1982, pp. 61-73

Chandy, K. M. and R. A. Russell, The Design of Multipoint Linkages in a Teleprocessing Tree Network, IEEE Transactions on Computers, Vol. C-21, No. 10, October, 1972

Charnes, A. and W. W. Cooper, Management Models and Industrial Applications of Linear Programming, Vols. I and II, No. 5, John Wiley and Sons, New York, New York, 1961

Chen, S. and R. Salgal, A Primal Algorithm for Solving a Capacitated Network Flow Problem with Additional Linear Constraints, Networks, Vol. 7, pp. 59-79, 1977

Chou, W. and M. Gerla, A Unified Flow and Congestion Control Model for Packet Networks, Proceedings of the Third ICCC, pp. 475-482, 1976

Clarke, G. and J. W. Wright, Scheduling of Vehicles from a Central Depot to a Number of Delivery Points, Operations Research, Vol. 12, pp. 568-681, July-August, 1964

Climaco, Joao Carlos Namorado and Ernesto Queiró Vieira Martins, A Bicriterion Shortest Path Algorithm, European Journal of Operational Research, Vol. 11, pp. 399-404, 1982

Colbourn, Charles J., The Combinatorics of Network Reliability, International Series of Monographs on Computer Science, Oxford University Press, 1987

Cowan, Robert, Teleconferencing Maximizing Human Potential, Reston Publishing Company, 1984

Dantzig. G. B. and J. H. Ramser, The Truck Dispatching Problem, Management Science, Vol. 6, pp. 80-91, 1959

Dantzig, G. B., All Shortest Routes in a Graph, Operations Research House, Stanford University, Technical Report 66-3, November, 1966

Davis, Gordon B. and Margrethe H. Olson, Management Information Systems, Conceptual Foundations, Structure and Development, McGraw-Hill, 1985

Dean, M., How Computers Should Talk to People, IBM System Journal, Vol. 21, No. 4, 1982

Dial, R., F. Glover, D. Karney, and D. Klingman, A Computational Analysis of Alternative Algorithms and Labeling Techniques for Finding Shortest Path Trees, Networks, Vol. 9, pp. 215-248, 1979

Diaz, Jose A., Finding a Complete Description of All Efficient Solutions to a Multiobjective Transportation Problem, Ekonomicko-Matematick y Obzor, Vol. 15, No. 1, 1979, pp. 62-73

Dijkstra, E. W., A Note on Two Problems in Connexion with Graphs, Numerical Mathematics, Vol. 1, pp. 269-271, October, 1959

Dos Santos, Brian L. and Martin L. Bariff, A Study of User Interface Aids for Model-Oriented Decision Support Systems, Management Science, Vol. 34, No. 4, April, 1988, pp. 461-468

Dutta, Amitava and Hemant Jain, A DSS for Distributed Computer System Design in the Presence of Multiple Conflicting Objectives, Decision Support Systems, 1986

Dutta, Amitava, Gary Koehler, and Andrew Whinston, On Optimal Allocation in a Distributed Processing Environment, Management Science, Vol. 28, No. 8, August, 1982, pp. 839-853

Ecker, J. G. and N. E. Shoemaker, Multiple Objective Linear Programming and the Trade-off-Compromise Set, Multiple Criteria Decision Making Theory and Application, edited by G. Fandel and T. Gal, Springer-Verlag, Berlin-New York, New York, 1980, pp. 60-73

Efe, Kemal, Chris Miller, and K. Hopper, The Kiwinet- Nicola Approach: Response Generation in a User-Friendly Interface, IEEE Transactions on Computers, 1983, pp. 66-77

Evans, Gerald, An Overview of Techniques for Solving Multiobjective Mathematical Programs, Management Science, Vol. 30, No. 11, November, 1984, pp. 1268-1282

Even, S., An Algorithm for Determining Whether the Connectivity of a Graph is at least k, SIAM Journal of Computing, Vol. 4, pp. 393-396, Sept. 1975

Evren, Ramazan, Interactive Compromise Programming, Journal of the Operational Research Society, Vol. 38, No. 2, February, 1987, pp. 163-172

Fandel, G. and T. Gal, Editors, Multiple Criteria Decision Making Theory and Application, Proceedings of the Third Conference, Hagen/Königswinter, West Germany, August 20-24, 1979, Springer-Verlag, 1980

Feldman, E., F. A. Lehrer, and T. L. Ray, Warehouse Location Under Continuous Economies of Scale, Management Science, Vol. 12, No. 9, May, 1966

Feltman, Charles, Local Networks, Popular Computing, April, 1983, pp. 115-123

Ferguson, Michael J., and Lorne G. Mason, Network Design for a Large Class of Teleconferencing Systems, IEEE Transactions on Communications, Vol. Com 32, No. 7, July, 1984

Floyd, R. W., Algorithm 97, Shortest Path, Communications of the ACM, Vol. 5, 1962, p. 345

Ford, L. R. and D. R. Fulkerson, Flows in Networks, Princeton University Press, 1962

Frank, H., and W. Chou, Topological Optimization of Computer Networks, Proceedings of the IEEE, Vol. 60, pp. 1385-1397, November, 1972

Frank, H. and I. Frisch, Communication, Transmission and Transportation Networks, Addison-Wesley, 1971

Frank, H., I. Frisch, and W. Chou, Topological Considerations in the Design of the ARPA Computer Network, Proceedings SJCC, pp. 581-587, 1970

Freimer, M. and P. L. Yu, Some New Results on Compromise Solutions for Group Decision Problems, Management Science, Vol. 22, No. 6, February, 1976, pp. 688-693

Gearhart, W. B., Compromise Solutions and Estimation of the Non-inferior Set, Journal of Optimization Theory and Applications, Vol. 28, May, 1979, pp. 29-47

Gearhart, W. B., Analysis of Compromise Programming, MCDM: Past Decade and Future Trends, Edited by Milan Zeleny, JAI Press, Greenwich, Connecticut, 1984, pp. 85-100

Geoffrion, A. M., J. S. Dyer and A. Feinberg, An Interactive Approach for Multicriterion Optimization with an Application to the Operation of An Academic Department, Management Science, Vol. 19, No. 4, pp. 357-368, December, 1972

Gerla, Mario and Leonard Kleinrock, On the Topological Design of Distributed Computer Networks, IEEE Transactions on Communications, Vol. Com-25, No. 1, January, 1977

Gershon, Mark and Lucien Duckstein, A Procedure for Selection of a Multiobjective Technique with Application to Water and Mineral Resources, Applied Mathematics and Computation, Vol. 14, 1984, pp. 245-271

Gershon, Mark, The Role of Weights and Scales in the Application of Multiobjective Decision Making, European Journal of Operational Research, Vol. 15, 1984, pp. 244-250

Gibson, Ronald W., A Primer for Evaluation and Purchasing Today's Local Networks, Data Communications, June, 1982

Girard, André and Yves Côté, Sequential Routing Optimization for Circuit Switched Networks, IEEE Transactions on Communications, Vol. Com-32, No. 12, December, 1984

Glover, Fred, and Darwin Klingman, Real World Applications of Network Related Problems and Breakthroughs in Solving Them Efficiently, ACM Transactions on Mathematical Software, Vol. 1, No. 1, March, 1975, pp. 47-55

Goicoechea, Ambrose, Don R. Hansen, and Lucien Duckstein, Multiobjective Decision Analysis with Engineering and Business Applications, John Wiley & Sons, 1982

Golden, B., M. Ball, and L. Bodin, Current and Future Research Directions in Network Optimization, Computers and Operations Research, Vol. 8, 1981, pp. 71-81

Goul, Michal, Barry Shan, and Fred M. Tonge, Using a Knowledge-Based Decision Support System in Strategic Planning Decisions. An Empirical Study, Journal of Management Information Systems, Spring, Vol. II, No. 4, 1986.

Gross, Donald and Carl M. Harris, Fundamentals of Queueing Theory, Second Edition, John Wiley & Sons, 1985

Hansen, P., Bicriterion Path Problems, Multiple Criteria Decision Making: Theory and Applications, G. Fandel and T. Gals, eds., Springer Heidelberg, 1980, 109-127

Haimes, Y. Y., W. A. Hall and H. T. Freedman, Multiobjective Optimization in Water Resources Systems: the Surrogate Worth Trade-Off Method, Elsevier Publishing Company, New York, New York, 1975

Henig, Mordechai I, The Shortest Path Problem with Two Objective Functions, European Journal of Operational Research, Vol. 25, 1985, pp. 281-291

Hessel, Mark and Milan Zeleny, Optimal Systems Design: Towards New Interpretations of Shadow Prices in Linear Programming, Computers and Operations Research, Vol. 14, No. 4, 1987, pp. 265-271

Hillier, Frederick S. and Gerald J. Lieberman, Introduction to Operations Research, Holden-Day, 1980

Ho, James K., Robust Goal Programming, Working Paper No. 26, The University of Tennessee, Department of Management Science, 1985

Holtzman, Albert G, Mathematical Programming for Operations Researchers and Computer Scientists, Dekker, 1981

Hurrion, R. D., Visual Interactive Modelling, European Journal of Operational Research, No. 23, 1986, pp. 281-287

Hwang, C. L. and A. S. M. Masud, Multiple Objective Decision Making--Methods and Applications, Lecture Notes in Economics and Mathematical Systems No. 164, Springer-Verlag, Berlin-New York, New York, 1979

Hwang, C. L., S. R. Paidy, and K. Yoon, Mathematical Programming with Multiple Objectives: A Tutorial, Computers and Operations Research, Vol. 7, pp. 5-31, 1980

Jarke, Matthias, M. Tawfik Jelassi, and Melvin F. Shakun, MEDIATOR: Towards a Negotiation Support System, European Journal of Operational Research, No. 31, 1987, pp. 314-334

Jarke, Matthias, Mohammed T. Jelassi, and Edward Stohr, A Data Driven User Interface for a Generalized Multiple Criteria Decision Support System, IEEE, 1984, pp. 127-133

Jarvenpaa, Sirkka L., The Effect of Task Demands and Graphical Format on Information Processing Strategies, Management Science, Vol. 35, No. 3, March, 1989

Jelassi, M. Tawfik, Matthias Jarke, and Edward A. Stohr, Designing a General Multiple Criteria Decision Support System, Journal of Management Information Systems, Vol. 1, No. 4, 1985

Keeney, R. L., Utility Functions for Multiattributed Consequences, Management Science, Vol. 18, No. 5 (Part 1), pp. B276-B287, 1972

Kelton, W. D., and A. M. Law, A Mean-Time Comparison of Algorithms for the All-Pairs Shortest-Path Problem with Arbitrary Arc Lengths, Networks, Vol. 8, 1978, pp. 98-106

Kiziltan, Gülseren, and Erkut Yucaoglu, An Algorithm for Multiobjective Zero-One Linear Programming, Management Science, Vol. 29, No. 12, December, 1983, pp. 1444-1453

Klein, H. K. and R. Hirschheim, Fundamental Issues of Decision Support Systems: A Consequentialistic Perspective, Decision Support Systems, Vol. 1, 1985, pp. 5-24

Kleinrock, L. and M. Gerla, Flow Control: A Comparative Survey, IEEE Transactions on Communications, Vol. COM-28, pp. 155-174, January, 1977

Kleitman, D., Methods for Investigating the Connectivity of Large Graphs, IEEE Transactions on Circuit Theory, Vol. CT-16, pp. 232-233, May, 1969

Klingman, Darwin and Nancy V. Phillips, Topological and Computational Aspects of Preemptive Multicriteria Military Personnel Assignment Problems, Management Science, Vol. 30, No. 11, November, 1984, pp. 1362-1375

Kriebel, Charles H. and Artur Raviv, An Economics Approach to Modeling the Productivity of Computer Systems, Management Science, No. 3, March, 1980, pp. 297-311

Kuehn, Alfred A. and Michael J. Hamburger, A Heuristic Program for Locating Warehouses, Management Science, Vol. 9, pp. 643-666, July, 1963

Kuhn, H. W., and A. W. Tucker, Nonlinear Programming in J. Neyman (ed.), Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, 1951

Kwak, N. K. and Stephen A. Delurgio, Quantitative Models for Business Decisions, Wadsworth, 1980

Lambrou, Fred H., DDP in the Corporate Business Plan, Journal of Systems Management, September, 1980, pp. 28-31

Laudon, Kenneth C., and Jane Price Laudon, Management Information Systems, a Contemporary Perspective, Macmillan Publishing Co., 1988

Lavia, A. and E. G. Manning, Perturbation Techniques for Topological Optimization of Computer Networks, Proceedings of the Fourth Data Communications Symposium, pp. 4-16 to 4-23, 1975

Lawler, Eugene, Combinatorial Optimization Networks and Matroids, Holt, Rinehart, and Winston, 1976

Lazaroui, D. F., and E. Staicut, A Markov Model for Availability of a Packet-Switching Computer Network, IEEE Transactions on Reliability, Vol. R-32, No. 4, October, 1983

LeBlanc, Larry J., and Ronald V. Simmons, A New Class of Models for Capacity Design of Packet-Switched Telecommunication Networks, Working Paper #87-04, Vanderbilt University, March, 1987, Revised, March, 1988

LeBlanc, Larry J., and Ronald V. Simmons, Design and Operation of Packet-Switched Networks Under Uncertain Message Requirements and Alternate Circuit-Switching Capabilities, Working Paper #88-04, Vanderbilt University, March, 1988

Lee, Jae K., E. Gerald Hurst, Jr., and Jae Sik Lee, The Third Generation Multiple Criteria Decision Support Systems, Working Paper, Korea Advanced Institute of Science and Technology, Department of Management Science, 1985

Lee, Yann-Hang, and Kang G. Shin, Optimal Reconfiguration Strategy for a Degradable Multimodule Computing System,, Journal of the Association for Computing Machinery, Vol. 34, No. 2, April 1987, pp. 326-348

Leigh, William E., and Clifford Burgess, Distributed Intelligence: Trade-offs and Decisions for Computer Information Systems, South-Western Publishing Company, Cincinnati, Ohio, 1987

Lenard, Melanie L., Representing Models as Data, Journal of Management Information Systems, Spring, 1986, Vol. 2, No. 4

Levine, P., and J.-Ch. Pomerol, PRIAM, an Interactive Program for Chosing Among Multiple Attribute Alternatives, European Journal of Operational Research, No. 25, 1986, pp. 272-280

Li, Han-Lin, Solving Discrete Multicriteria Decision Problems Based on Logic-Based Decision Support Systems, Decision Support Systems, Vol. 3, 1987, pp. 191-119

Little, D., A Proof for the Queueing Formula: L = Lambda * W, Operations Research, Vol. 9, pp. 383-387, May, 1961

Locks, Mitchell O., Recursive Disjoint Products: A Review of Three Algorithms, IEEE Transactions on Reliability, Vol. R-31, April, 1982

Locks, Mitchell O., Topological Reliability: A Tutorial, Research Series 84-7, Oklahoma State University, February, 1984

Loomis, Mary E. S., Data Communications, Prentice-Hall, 1983

Lootsma, F. A., Numerical Methods for Non-Linear Optimization, Academic Press, 1972

Lorin, Harold, DDP: How to Fail, Datamation, February, 1981, pp. 60-65

Madey, Gregory R. and Burton V. Dean, Strategic Planning for Investment in R&D Using Decision Analysis and Mathematical Programming, IEEE Transactions on Engineering Management, Vol. EM-32, No. 2, May, 1985, pp. 84-90

Malhora, V. M., M. P. Kumar, and S. N. Maheshwari, An O(¦V¦3) Algorithm for Finding Maximum Flows in Networks, Information Processing Letters, Vol. 7, pp. 277-278, October, 1978

Marney-Petix, Victoria C., Networking and Data Communications, Reston Publishing Co., 1986

Martin, James, Teleprocessing Network Organization, Prentice-Hall, 1970

McGregor, P. and D. Shken, Network Design: An Algorithm for the Access Facility Location Problem, IEEE Transactions on Communications, Vol. COM-25, pp. 61-73, Jan, 1977

McQuillin, J. M. and D. C. Walden, The ARPA Network Design Decisions, Computer Networks, Vol. 1, pp. 243-289, August, 1977

Meadow, Charles T. and Albert S. Tedesco, Telecommunications for Management, McGraw-Hill, 1985

Meilach, Dona Z., Dynamics of Presentation Graphics, Dow Jones-Irwin, 1986

Minch, Robert P., and G. Lawrence Sanders, Computerized Information Systems Supporting Multicriteria Decision Making, Decision Sciences, Vol. 17, No. 3, Summer, 1986, pp. 365-413

Minieka, Edward, Optimization Algorithms for Networks and Graphs, Dekker, 1978

Monarchi, D. E., C. E. Kisiel, and L. Duckstein, Interactive Multiobjective Programming in Water Resources: A Case Study, Water Resources Research, Vol. 9, No. 4, pp. 837-850, 1973

Monarchi, D. E., J. E. Weber, and L. Duckstein, An Interactive Multiple Objective Decision-Making Aid Using Nonlinear Goal Programming, Multiple Criteria Decision Making, edited by M. Zeleny, Lecture Notes in Economics and Mathematical Systems No. 123, Springer-Verlag, Berlin-New York, New York, pp. 235-253, 1976

Morse, J. N., Reducing the Size of the Nondominated Set: Pruning by Clustering, Computers and Operations Research: Special Issue on Mathematical Programming with Multiple Objectives, edited by M. Zeleny, Vol. 7, No. 1-2, pp. 55-60, 1980

Mote, John, Ishwar Murthy, and David L. Olson, An Empirical Investigation of the Number of Pareto-Optimal Paths Obtained for Bicriterion Shortest Path Problems, Paper presented at ORSA/TIMS, Denver, Colorado, 1988

Nakayama, H., T. Tanino and Y. Sawaragi, An Interactive Optimization Method in Multicriteria Decisionmaking, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-10, No. 3, pp. 163-169, March, 1980

Newman, William M., Designing Integrated Systems for the Office Environment, McGraw-Hill Book Company, New York, New York, 1987

Nijkamp, P., and J. Spronk, Interactive Multiple Goal Programming, Centre for Research in Business Economics Report. 7803/A, Erasmus University, Rotterdam, 1978

Nijkamp, P., and J. Spronk (eds.), Multiple Criteria Analysis: Operational Methods, Gower Press, London, 1981

Ozernoj, V. M., Using Preference Information in Multistep Methods for Solving Multiple Criteria Decision Problems, G. Fandel, and T. Gal (eds.) Multiple Criteria Decision Making Theory and Application, Springer, Berlin, 1979, pp. 314-328

Orman, Levent, Flexible Management of Computational Models, Decision Support Systems, Vol. 2, 1986, pp. 225-234

Paller, Alan T. Improving Management Productivity with Computer Graphics, IEEE Computer Graphics and Applications, Vol. 1, No. 4, pp. 9-16, October, 1981

Perl, Jossef, and Mark S. Daskin, A Unified Warehouse Location-Routing Methodology, Journal of Business Logistics, Vol. 5, No. 1, pp. 103-111

Pomerol, J. Ch., and T. Trabelsi, An Adaptation of PRIAM to Multiobjective Linear Programming, European Journal of Operational Research, No. 31, 1987, pp. 335-341

Pouzin, L., Flow Control in Data Networks-Methods and Tools, Proceedings of the Third ICCC, pp. 467-474, 1976

Powers, Charles H., Vilfredo Pareto, Masters of Social Theory, Volume 5, Sage Publications, 1987

Rasmussen, L. M., Zero-One Programming with Multiple Criteria, European Journal of Operational Research, Vol. 26, pp. 83-95, 1986

Remus, William and Jeffrey E. Kottemann, Semi-Structured Recurring Decisions: An Experimental Study of Decision-Making Models and Some Suggestions for DSS, MIS Quarterly, June, 1987

Rosner, Roy D., Distributed Telecommunications Networks via Satellites and Packet Switching, Lifetime Learning Publications, 1982

Sadagopan, S., and A. Ravindran, Interactive Algorithms for Multiple Criteria Nonlinear Programming Problems, European Journal of Operational Research, No. 25, 1986, pp. 247-257

Sakawa, Masatoshi, and Fumiko Seo, Interactive Multiobjective Decisionmaking for Large-Scale Systems and Its Application to Environmental Systems, IEEE Transactions, Vol. SMC-10, No. 12, December, 1980

Schwartz, M. and T. E. Stern, Routing Techniques Used in Computer Communications Networks, IEEE Transactions on Communications, Vol. COM-28, pp. 539-552, April, 1980

Sen, Tapan, Farhad M. E. Raiszadeh, and Parthasarati Dileepan, NOTES, A Branch-and-Bound Approach to the Bicriterion Scheduling Problem Involving Total Flowtime and Range of Lateness, Management Science, Vol. 34, No. 2, February, 1988, pp. 254-260

Sharda, R., and J. W. Gentry, Modeling for Competitive Bidding with IFPS, Transactions of the Third International Conference on Decision Support Systems, Providence, RI: Institute of Management Science, 1983

Sharda, Ramesh, Steve H. Barr, and James C. McDonnell, Decision Support System Effectiveness: A Review and an Empirical Test, Management Science, Vol. 34, No. 2, February, 1988, pp. 139-159

Shetty, Bala, David L. Olson, M. A. Venkataramanan, and Ishwar Murthy, Network Reoptimization Procedures for Multiobjective Network Problems, Presented Paper at ORSA/TIMS, Denver, Colorado, 1988

Shier, D., An Empirical Investigation of Some Bicriterion Shortest Path Algorithms, to appear in The European Journal of Operations Research

Shycon, Harvey N. and Richard B. Maffei, Remarks on the Kuehn-Hamburger Paper, Management Science, Vol. 9, pp. 667-668, July, 1963

Simon, H., The New Science of Management Decisions, Harper and Row, New York, 1960

Srinivasan, V., and G. L. Thompson, Algorithms for Minimizing Total Cost, Bottleneck Time and Bottleneck Shipment in Transportation Problems, Naval Research Logistics Quarterly, No. 23, 1976, pp. 567-595

Stallings, William, Data and Computer Communications, MacMillan Publishing Company, 1988

Steiglitz, K., P. Weiner, and D. J. Kleitman, The Design of Minimum-Cost Survivable Networks, IEEE Transactions on Circuit Theory, Vol. CT-16, pp. 455-460, November, 1969

Steuer, R. E., An Interactive Multiple Objective Linear Programming Procedure, in Multiple Criteria Decision Making, TIMS Studies in the Management Sciences, M. K. Starr and M. Zeleny (eds.), Vol. 6, North-Holland Publishing, Amsterdam, 1977, pp. 225-239

Steuer, R. E., and A. T. Schuler, An Interactive Multiple Objective Linear Programming Approach to a Problem in Forest Management, Operations Research, Vol. 26, No. 2, 1978, pp. 254-269

Steuer, R. E., Multiple Criteria Optimization: Theory, Computation, and Application, Wiley, New York, 1986

Stewart, William R., Jr., Accelerated Branch Exchange Heuristics for Symmetric Traveling Salesman Problems, Networks, Vol. 17, Winter, 1987, pp. 423-437

Stone, Harold S., Multiprocessor Scheduling with the Aid of Network Flow Algorithms, IEEE Transactions on Software Engineering, January, 1977

Stohr, Edward, Decision Support and Knowledge-Based Systems: A Special Issue, Journal of Management Information Systems, Vol. II, No. 4, Spring, 1986

Tabourier, Y., All Shortest Distances in a Graph. An Improvement to Dantzig's Inductive Algorith, Discrete Mathematics, Vol. 4, 1973, pp. 83-87

Tanenbaum, Andrew S., Computer Networks, Prentice Hall, 1981

Thierauf, Robert J., Robert C. Klekamp, and Marcia L. Ruwe, Management Science, A Model Formulation Approach with Computer Applications, Charles E. Merrill, 1985

Thomas, J. C. and J. M. Carroll, Human Factors in Communication, IBM SYSt J, Vol. 20, No. 2, 1981

TIMS: Studies in the Management Sciences, Vol. 6, Multiple Criteria Decision Making, Edited by Martin K. Starr and Milan Zeleny, 1977

Villarreal, Bernado, Mark H. Karwan, and Stanley Zionts, An Interactive Branch and Bound Procedure for Multicriterion Integer Linear Programming, G. Fandel, and T. Gal (eds.) Multiple Criteria Decision Making Theory and Application, Springer, Berlin, 1979 pp. 448-467

Voelcker, John, Helping Computers Communicate, IEEE Spectrum, March, 1986, pp. 61-70

Vincke, Phillippe, Analysis of Multicriteria Decision Aid in Europe, European Journal of Operational Research, No. 25, 1986, pp. 160-168

Walsh, Myles E., Database & Data Communications Systems: A Guide for Managers, Reston Publishing, 1983

Warburton, Arthur, Approximation of Pareto Optima in Multiple-Objective, Shortest-Path Problems, Journal of Operations Research, Vol. 35, No. 1, January-February, 1987, pp. 70-79

Weber, E. Sue, Systems to Think with: A Response to A Vision for Decision Support Systems, Journal of Mangement Information Systems, Spring, 1986, Vol. 2, No. 4

Weitzman, Cay, Distributed Micro/Minicomputer Systems, Structure, Implementation, and Application, Prentice-Hall, 1980

White, D. J., The Set of Efficient Solutions for Multiple Objective Shortest Path Problems, Computers and Operations Research, Vol. 9, No. 2, 1982, pp. 101-107

Yaverbaum, Gayle J., Critical Factors in the User Environment: An Experimental Study of Users, Organizations and Tasks, MIS Quarterly, March, 1988, pp. 75-88

Yilmaz, Mustafa, A Theory of the Displaced Ideal, MCDM: Past Decade and Future Trends, Edited by Milan Zeleny, JAI Press, Greenwich, Connecticut, 1984, pp. 101-116

Yu, P. L., A Class of Solutions for Group Decision Problems, Management Science, Vol. 19, No. 8, April, 1973, pp. 936-946

Yu, P. L. and G. Leitmann, Compromise Solutions, Domination Structures and Salukvadze's Solution, Multicriteria Decision Making and Differential Games, edited by G. Leitmann, Plenum Press, New York, New York, 1976, pp. 85-101

Zadeh, L., Optimality and Non-Scalar-Valued Performance Criteria, IEEE Transactions on Automatic Control, AC-9, No. 59, 1963.

Zeleny, Milan, A Concept of Compromise Solutions and the Method of the Displaced Ideal, Computers and Operations Research, Vol. 1, Nos. 3-4, pp. 479-496, 1974

Zeleny, Milan, Games with Multiple Payoffs, International Journal of Game Theory, Vol. 4, No. 4, pp. 171-191, 1976

Zeleny, Milan, The Theory of the Displaced Ideal, Multiple Criteria Decision Making, edited by M. Zeleny, Lecture Notes in Economics and Mathematical Systems No. 123, Springer-Verlag, Berlin-New York, New York, pp. 153-206, 1977

Zeleny, Milan, A Case Study in Multiobjective Design: De Novo Programming, Multiple Criteria Analysis: Operational Methods, edited by Nijkamp and J. Sprong, Gower Press, London, England, 1981

Zeleny, Milan, Multiple Criteria Decision Making, McGraw-Hill, New York, New York, 1982

Zeleny, Milan, Optimal System Design with Multiple Criteria: De Novo Programming Approach, Elsevier Science Publishers, 1986

Zionts, S. and J. Wallenius, An Interactive Programming Method for Solving the Multiple Criteria Problem, Management Science, Vol. 22, No. 6, 1976, pp. 652-663

Zionts, S., Multiple Criteria Mathematical Programming: An Updated Overview and Several Approaches, June, 1987

APPENDIXES

APPENDIX A

SAMPLE DATA BASE

SAMPLE DATA BASE

1. Table - General Network Information
   a. Problem ID
   b. Problem Description
   c. Case ID
   d. Case Description
   e. Number of Nodes
   f. Maximum Number of Connections (Calculated)
   g. Number of Objective Functions
   h. Number of Unique Communications Media Connections
   i. Maximum Allowable Service Level
   j. Maximum Allowable Setup Cost
   k. Maximum Allowable Operating Cost
   l. Pessimistic/Most Likely/Optimistic Scenario Switch

2. Table - Node Data
   a. Compromise Programming Problem Number
   b. Node Number
   c. Node Name
   d. Node Description
   e. Node Latitude
   f. Node Longitude

3. Table - Internode Data
   a. Problem ID
   b. Node i Name
   c. Node j Name
   d. Pessimistic Traffic Pattern between Node i and Node j
   e. Most Likely Traffic Pattern between Node i and Node j
   f. Optimistic Traffic Pattern between Node i and Node j
   g. Pessimistic File Size between Node i and Node j
   h. Most Likely File Size between Node i and Node j
   i. Optimistic File Size between Node i and Node j
   j. Distance between Node i and Node j, in miles

4. Table - Communication Media Data
   a. Problem ID
   b. Communication Media ID
   c. Communication Media Description

5. Table - Objective Function Coefficients
   a. Problem ID
   b. Node i Name
   c. Node j Name
   d. Communication Media ID
   e. Set Up Costs
   f. Operating Costs
   g. Effective Transfer Rates

APPENDIX B

DATA FLOW DIAGRAMS AND SCREEN MENUS

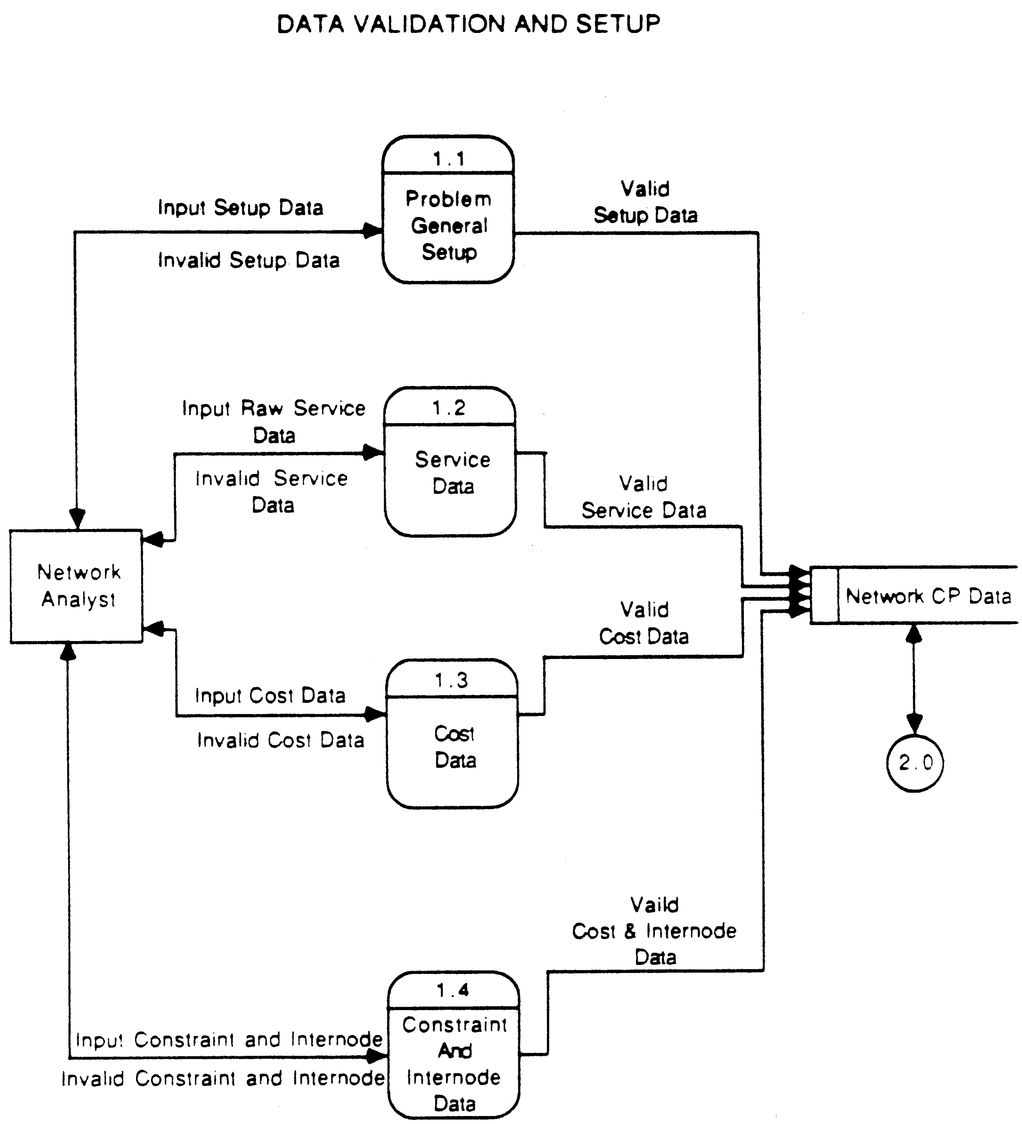


Exhibit 1. General Setup Data Flow Diagram

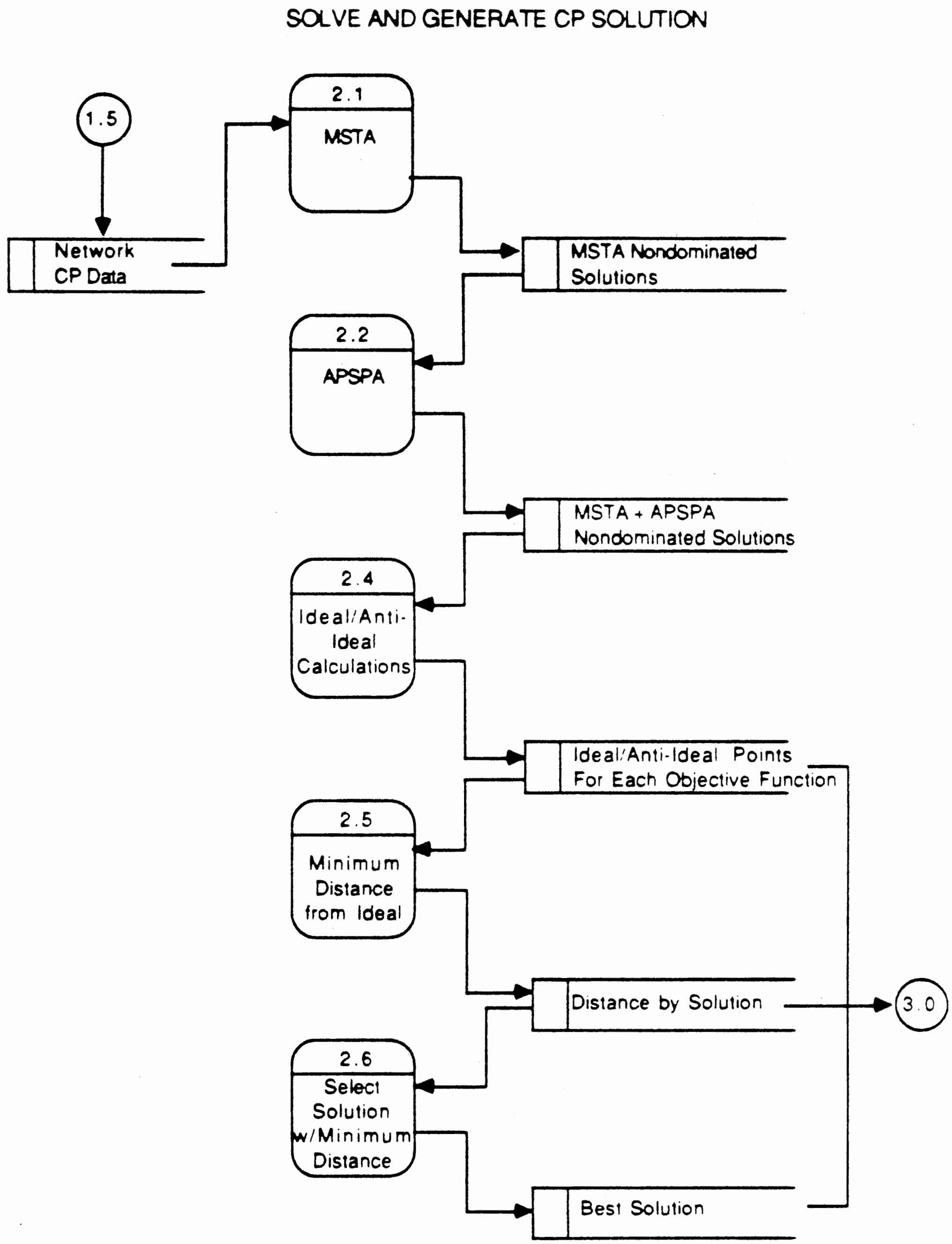


Exhibit 2. Solve and Generate Solutions Data Flow Diagram

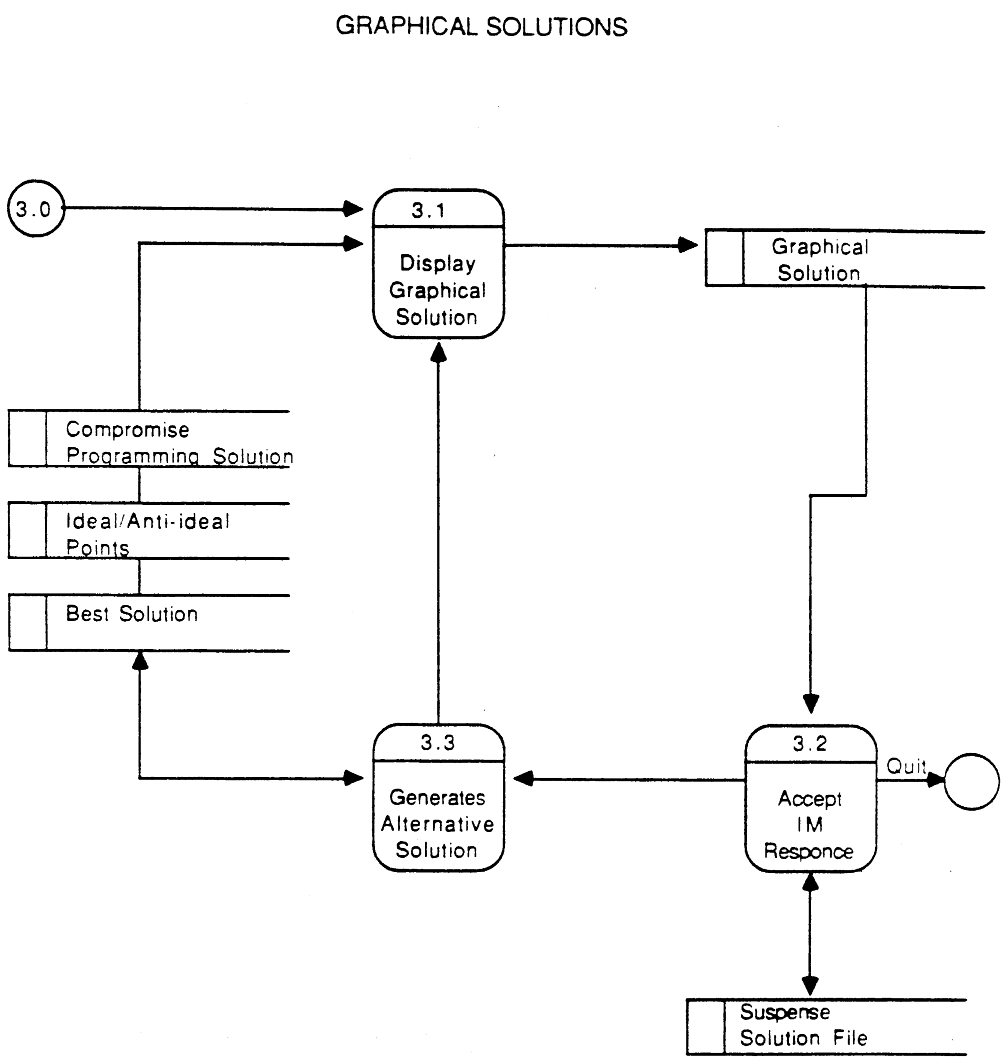


Exhibit 3. Graphical Solutions Data Flow Diagram

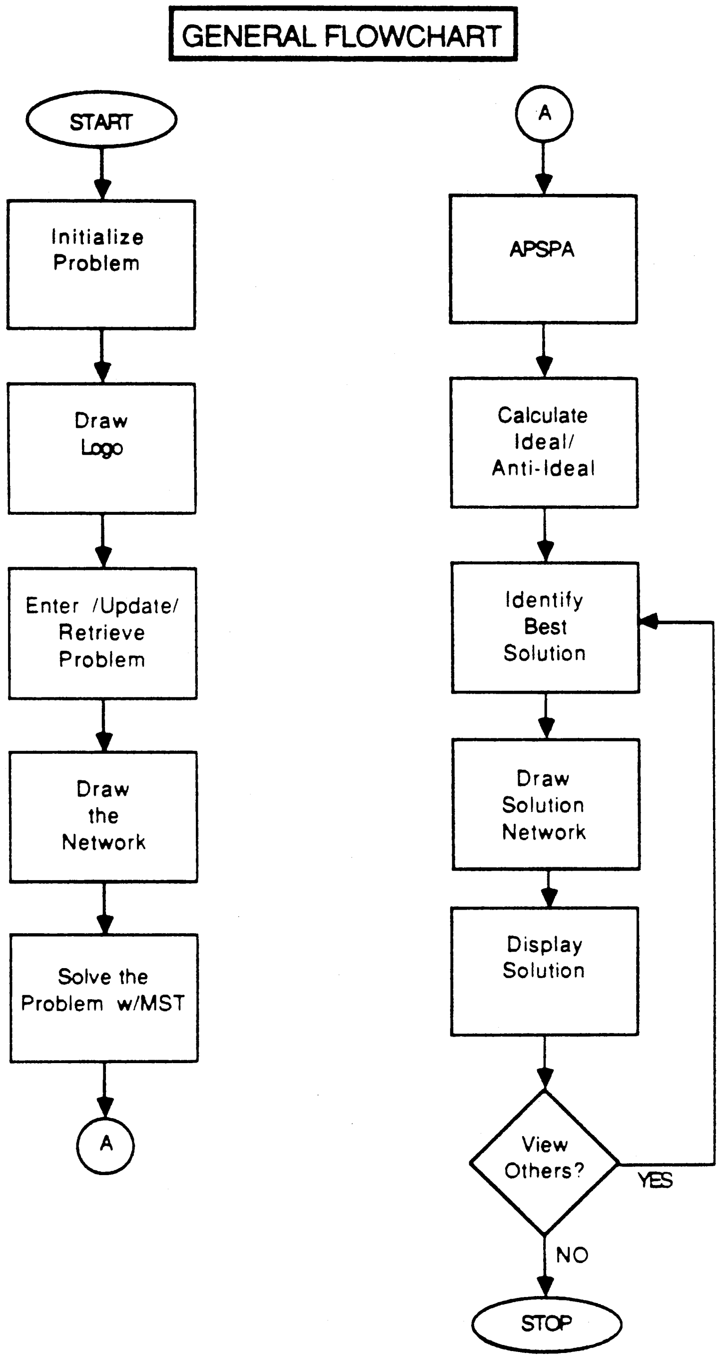


Exhibit 4. General DSS Flowchart

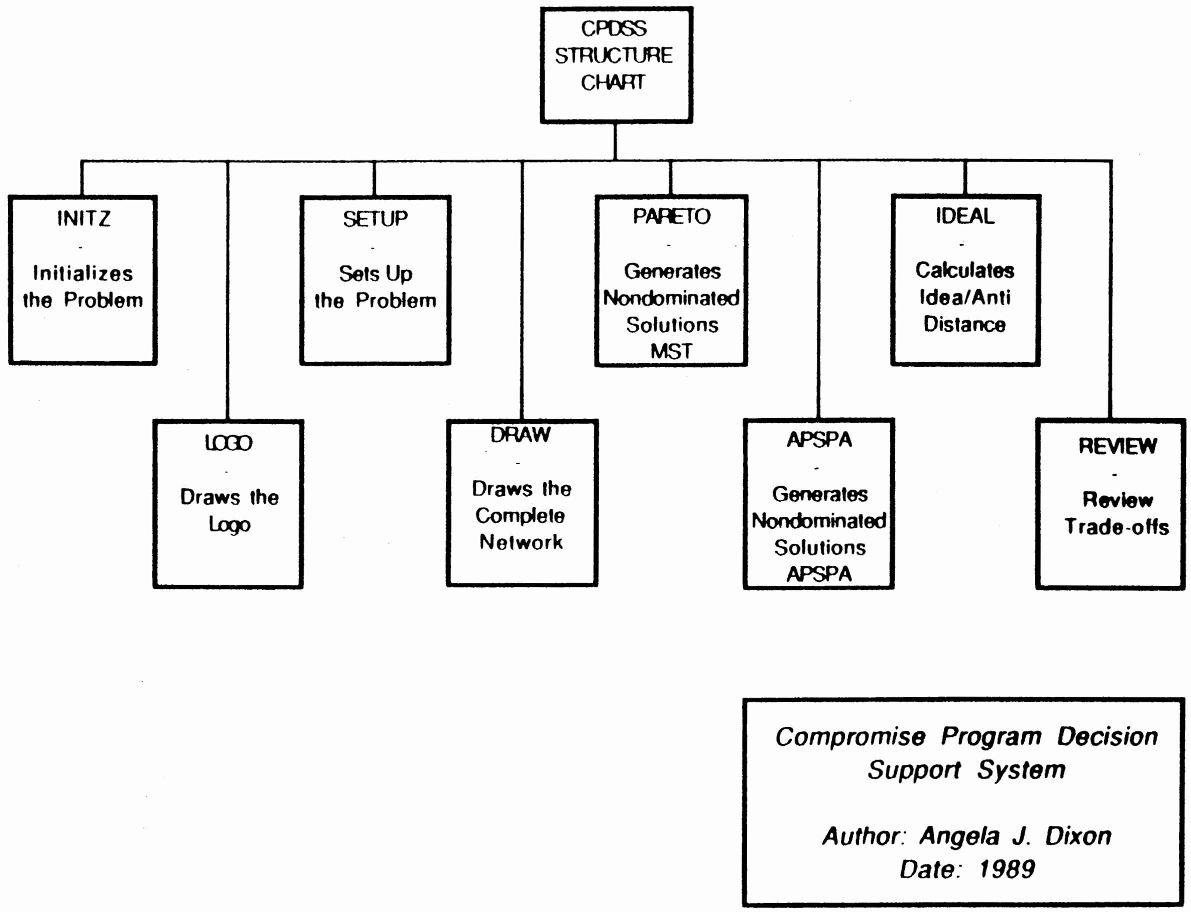


Exhibit 5. CPDSS Structure Chart

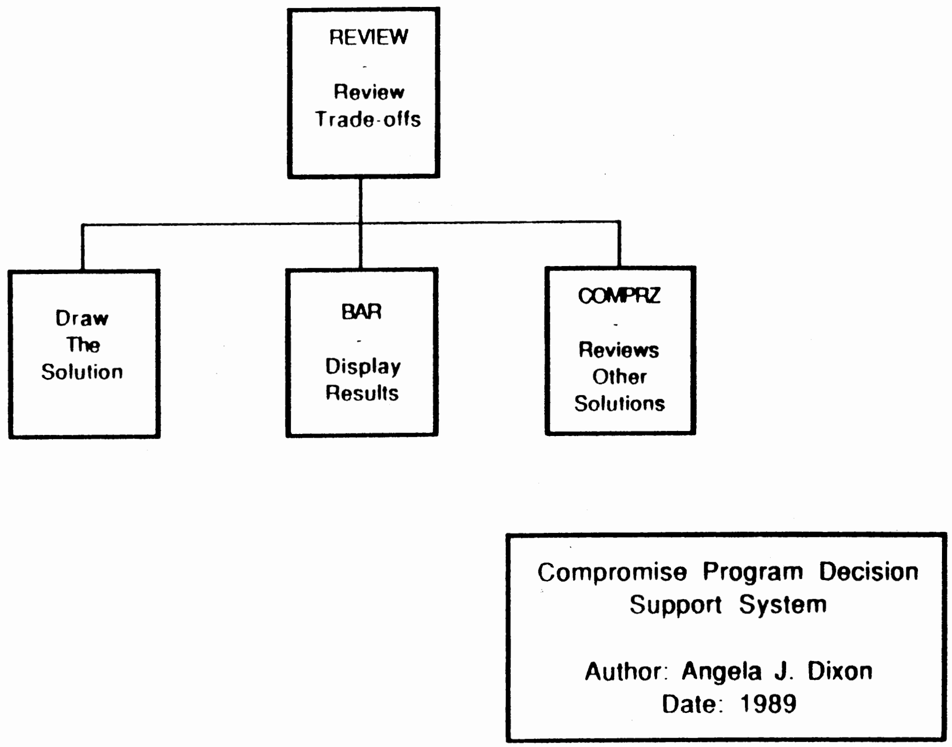


Exhibit 6. Review Tradeoffs Structure Chart

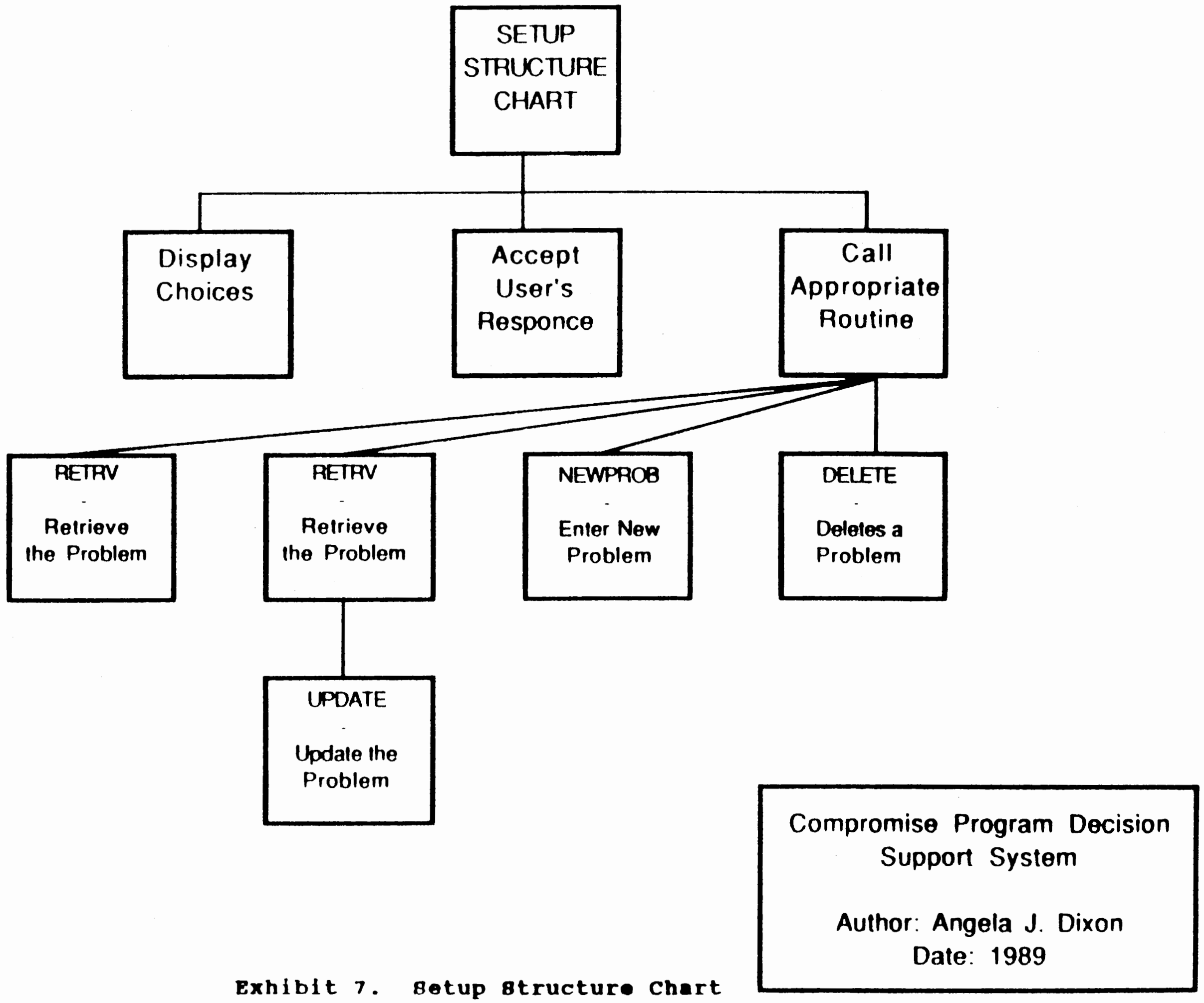


Exhibit 7. Setup Structure Chart

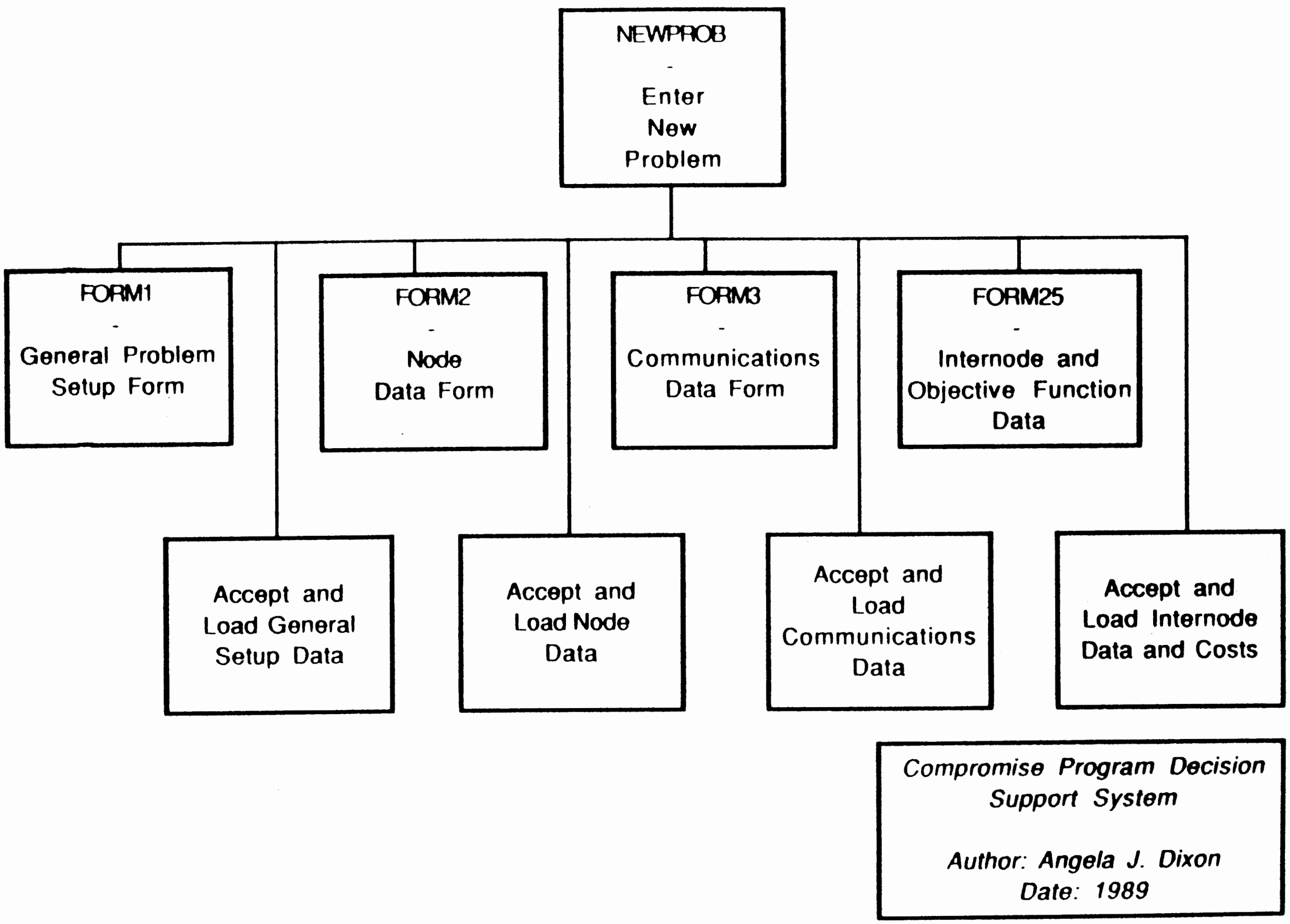


Exhibit 8. New Problem Sructure Chart

COMPROMISE PROGRAMMING SETUP MENU

1. USE EXISTING PROBLEM
2. UPDATE AND USE EXISTING PROBLEM
3. ENTER NEW PROBLEM
4. DELETE EXISTING PROBLEM
5. EXIT

Please Enter Your Choice ■

Exhibit 9. Compromise Programming Setup Menu

COMPROMISE PROGRAMMING UPDATE MENU

1. GENERAL PROBLEM SETUP
2. ADD NEW NODE
3. CHANGE EXISTING NODE
4. ADD NEW OBJECTIVE FUNCTION / INTERNODE DATA
5. CHANGE EXISTING OBJECTIVE FUNCTION / INTERNODE DATA
6. ADD NEW COMMUNICATIONS MEDIA
7. CHANGE EXISTING COMMUNICATIONS MEDIA
8. EXIT

Please Enter Your Choice ■

Exhibit 10. Compromise Programming Update Menu

Compromise Program General Problem Setup

Identification:	Case ID:
Name:	
# of Nodes in Network :	
Max # Connect. = N!/[(N-2)!2!] :	
# Objective Functions :	
# Types of Communications Media :	
Maximum Allowable Delay :	
Maximum Acceptable Setup Costs :	
Maximum Acceptable Operating Costs :	

Exhibit 11. Compromise Programming General Problem Setup

Compromise Program Node Data
Problem Number   Case ID

Node Number

Node Identification

Node: Latitude   Longitude

Exhibit 12. Compromise Programming Node Data Screen

Compromise Program Communications Media Names
Problem Number Case ID

Communications Identification

Communications Name

Maximum Transfer Rate

Exhibit 13. Compromise Programming Communications Media Data Screen

NOTE: The following data values are reasonable estimations for the data transfer rate for various communications media:

TABLE V

DATA TRANSFER RATES

Medium	Data Transfer Rate (bps)
Twisted Pair	50-60 K
Baseband Coax	50 M (max)
Broadband Coax	350 M (max)
Fiber Optics	500K-1 G
Microwave	56-256 K

where K = Kilobits/sec
M = Megabits/sec
G = Gigabits/sec

Sources: Marney-Petix (1986), p. 98
Laudon and Laudon (1988), p. 278

Compromise Program Internode Data
Problem Number Case ID

Node I	Node J	Distance	
Communications Type	Transfer Rate	Setup Cost	Operating Cost
	Traffic	File Size	Effec. Rate
Optimistic			
Most Likely			
Pessimistic			

NOTE: Optimistic is minimum traffic/smallest file size

Exhibit 14. Compromise Programming Internode Data Screen

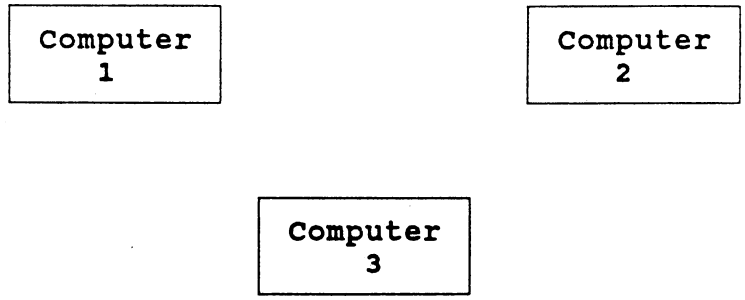
APPENDIX C

COMPUTER NETWORK DESIGN CASES

Computer Manual Network Design Case

The purpose of this exercise is to manually design a computer network. The nodes in the network represent computer sites and these computer sites may be connected by one of several different communications media. The aim of the design is to satisfy several diverse objectives which are inherently in conflict with one another. The three specific design objectives are to minimize the setup cost for the network; to minimize the operating cost of the network; and to minimize the transfer time between any two nodes in the network. Because these objectives are in conflict, there is not just one optimal design solution, rather there are many different solutions. The choice of the best solution will be based upon the user's preferences in minimizing either setup cost, operating cost, or transfer time between nodes, or some compromise between these objectives.

For example, if you were designing a network with three computers and two possible communications media (fibre optics and baseband coaxial cable) between each pair of computers,



you could connect computer 1 to computer 2 with fibre optics and computer 2 to computer 3 with coaxial cable; or you could connect computer 1 to computer 3 with fibre optics and computer 1 to computer 2 with coaxial cable; or you could use only fibre optics to connect, etc. But, if you are trying to minimize total setup cost, then you would need to consider how much it initially costs to set up a connection between each of the computers for each of the communications media and base your decision on the costs.

For example, pull out the SAMPLE PROBLEM from your packet. This computer network has three nodes, three objective functions, and two communications media. The maximum allowable transfer time is 2 seconds; the maximum allowable setup cost is $500,000; and the maximum allowable operating cost is $33,000. The two communications media are fibre optics and baseband coaxial cable and their respective transfer rates are 100,000,000 bits per second (bps) and 500,000 bps. The three nodes are named X1, X2, and X3. The distance, traffic patterns (largest number of files, most likely number of files, and smallest number of files), and the file sizes (largest, most likely, and smallest) are also given for each pair of nodes. And, finally the setup and operating cost for each communications media is given for each pair of nodes.

If your design was to connect X1 to X2 using communications media 1 (fibre optics) and to connect X1 to X3 using communications media 2 (baseband coaxial cable) then the total setup cost would be $$230,000 (120,000 + 110,000) and the total operating cost would be $23,000 (10,000 + 13,000). The transfer time can be estimated in several ways. An easy method is to multiply the largest number of files by the largest file size times 8 (bits per byte) and divide by the maximum transfer rate for that media. Or, if you are familiar with queueing theory, you can use a more sophisticated method of estimating transfer rate.

The current project is included in your packet. You may design the network using any criteria or scheme or algorithm that you wish, but your design must adhere to these rules:

- All nodes must be connected
- Each pair of nodes may be connected with one or more of the communications media
- Each node must be dually connected (connected to at least two other nodes)

- o The total setup cost must not exceed $4,000,000.
- o The total operating cost must not exceed $30,000.
- o The transfer time between any two nodes is based upon the file sizes and traffic patterns between the nodes and the communications media transfer rate
- o The total transfer time between any two nodes must not exceed 2 seconds

The data for the problem is included in your packet. A calculator and several generalized drawings of the computers have been supplied for your use.

Computer Decision Support System

Network Design Case

The purpose of this exercise is to design a computer network using a decision support system. The nodes in the network represent computer sites and these computer sites may be connected by one of several different communications media. The aim of the design is to satisfy several diverse objectives which are inherently in conflict with one another. The three specific design objectives are to minimize the setup cost for the network; to minimize the operating cost of the network; and to minimize the transfer time between any two nodes in the network. Because these objectives are in conflict, there is not just one optimal design solution, rather there are many different solutions. The choice of the best solution will be based upon the user's preferences in minimizing either setup cost, operating cost, or transfer time between nodes, or some compromise between these objectives.

To become familiar with the system, data for a seven node computer network has been entered and may be used to design an initial network. Then the user will review all of the data for a smaller network problem, let the system solve the smaller problem, and select the best design according to his/her personal criteria.

The directions for using the Decision Support System (DSS) are:

Type in at the C: prompt CPDSS to start the program.

The logo will print. Hit any key to get into the main program.

Select menu option 1 (USE AN EXISTING PROBLEM) from the Compromise Programming Setup Menu and Hit Return. When the list of available problems prints, select Problem LAN, Case A. (Hit enter after typing in LAN and hit enter after typing in A.)

The program will read the problem data from the data base and solve the problem. The entire network with all possible communications media links will show on the upper right of the screen. When the COMPUTING message appears on the screen, the problem is being solved by a network algorithm and when the RECOMPUTING message appears, the problem is being solved by another network algorithm. The two algorithms will generate several candidate solutions and will display on the screen the best (ideal or minimum) and worst (anti-ideal or maximum) possible achievement level among the candidate solutions. An individual solution is selected by the decision support system and displayed to the user. The values are displayed as a unit distance from the ideal achievement level.

The objective is to study different solutions and their achievement levels in order to ascertain the solution that best meets the user's personal preferences regarding minimizing setup costs, operating costs, and transfer time. There are trade-offs among the objectives: a minimum setup cost might have a maximum transfer time between any pair of nodes; or a minimum operating cost might have a high level of setup costs. Again, there is no 'best' or optimal answer to the problem, just a preferred solution by a user.

Position the MOUSE CURSOR on one or more objectives at the desired percentage of the ideal that you would like to consider and CLICK LEFT. When all preference levels have been indicated, CLICK RIGHT and the program will find a solution that is the closest to the indicated preferences and display it. There are several solutions, so several different levels may be selected for one or more objective functions and different solutions will be obtained.

After reviewing a few or several of the solutions select a preferred solution, position the MOUSE CURSOR on the PRINT SOLUTION option and CLICK LEFT. The solution will be printed. Return to the main menu by a RIGHT CLICK on the mouse that is not preceded by a LEFT CLICK.

The Design Problem

NOTE: Refer to the printout of the data for your particular problem during this part of the exercise. Select the UPDATE AN EXISTING PROBLEM option. Enter the problem identification of LAN and the case identification of either 4 or 5 as you have been instructed. Answer N to the question 'Do you wish to retrieve previous solutions' and the COMPROMISE PROGRAMMING UPDATE MENU will appear. Select the GENERAL PROBLEM SETUP option (number 1) and the data provided on your data sheet will appear on the screen. Any data on this screen or subsequent screens may be changed by positioning the mouse cursor over the data and clicking left and then entering the new value from the keyboard, with two exceptions. The only data that cannot be altered is the number of objective functions and the maximum # connections. The number of objective functions is fixed at three and the maximum number of connections is calculated as [N!/(N-2)!2!].

After viewing the general setup data, return to the update menu by clicking right on the mouse. Then select the CHANGE EXITING NODE option (number 3). Cycle through the node data by clicking right on the mouse. Note the latitude and longitude data values. These are used for drawing the network on the screen. The abbreviated node name is used to label the node on the drawing. After all nodes have been viewed (either four or five, depending on which problem you have been assigned) a right click on the mouse will return you to the update menu.

Select the CHANGE EXISTING COMMUNICATIONS MEDIA option next (number 7). If you were assigned the four node problem, there are four communications media, and if you were assigned the five node problem, there are three communications media. Cycle through the communications media by clicking right on the mouse. Make particular note of the maximum transfer rate for each of the communications media. It is used in calculating the effective transfer time between a pair of nodes. The communications media name is used for the legend of color assignments under the network drawing. After all media have been viewed, a click right will return you to the update menu.

The last data to view is the internode data. Select the CHANGE EXISTING OBJECTIVE FUNCTION/INTERNODE DATA (option 5). For each pair of nodes there is one set of data about traffic patterns and file sizes that remains constant for the pair. There is also communications media-dependent data that changes for each media: the setup cost and the operating cost, which is entered by the user; and the effective transfer time, which is calculated based on traffic patterns, file sizes, and the maximum transfer rate for the media. Pay particular attention to the costs and the transfer time. In some cases, both the setup cost and the operating cost are lower for one communications media than for all remaining communications media between a pair of nodes. When solving the problem, the model will select this communications media, if it chooses to connect these two nodes and no other criteria are violated. For the slower communications media, the effective transfer rate sometimes violates the maximum transfer time assigned in the general setup of 2 seconds. The model will never select this media to connect the nodes due to the time violation. Cycle through all of the internode data by clicking right on the mouse and when all data have been viewed, a right click will return you to the update menu.

Select the EXIT option from the menu and the problem will be solved. Follow the same procedure as before: examine alternative solutions; select the 'best' solution according to your preferences; and print this solution.

At this point, please complete the questionnaire provided in your packet, and turn in all of your results.

Thank you for your participation.

Data for the 3 Node Sample Problem

Number of Nodes: 3

Number of Communications Media: 2

Total Setup Cost: $500,000.

Total Operating Cost: $33,000.

Maximum Allowable Transfer Time between nodes: 2 seconds

Communnications Media 1 is Fibre Optics and the maximum transfer rate is 100,000,000 bits per second.

Communications Media 2 is Baseband Cable and the maximum transfer rate is 500,000 bits per second.

The traffic and file size patterns between nodes is:

		File Sizes			# Files Sent			
From	To	Most Likely	Biggest	Smallest	Most Likely	Biggest	Smallest	Dist.
X1	X2	15000	18000	12000	1500	1800	1200	2.2
X1	X3	12000	15000	10000	2450	2500	2200	2.9
X2	X3	1800	2000	1200	1600	1750	1200	2.8

The Setup cost and The Operating Cost for Each Communications media and for each pair of nodes is:

From	To	By	Setup Cost	Operating Cost
X1	X2	1	120000	10000
		2	135000	9300
X1	X3	1	190000	10000
		2	110000	13000
X2	X3	1	100000	12200
		2	82000	13600

Data for a Four Node Design Network Problem

Number of Nodes:	4
Number of Communications Media:	4
Maximum Allowable Transfer Time between nodes:	2 seconds
Total Setup Cost:	$4,000,000.
Total Operating Cost:	$30,000.

Node number 1 is named X1. Latitude: 26 Longitude: 87

Node number 2 is named X2. Latitude: 36 Longitude: 83

Node number 3 is named X3. Latitude: 36 Longitude: 97

Node number 4 is named X4. Latitude: 36 Longitude: 92

Communnications Media 1 is Fibre Optics and the maximum transfer rate is 100,000,000 bits per second.

Communications Media 2 is Broadband Cable and the maximum transfer rate is 500,000,000 bits per second.

Communications Media 3 is Microwave/Satellite and the maximum transfer rate is 60,000 bits per second.

Communications Media 4 is Twisted Pair and the maximum transfer rate is 50,000 bits per second.

The traffic and file size patterns between nodes is:

			Traffic	File Size	Distance
X1	X2	Smallest	1200.	10000.	2.2
		Most Likely	1500.	12000.	
		Biggest	1800.	15000.	
X1	X3	Smallest	2200.	10000.	2.8
		Most Likely	2450.	12000.	
		Biggest	2500.	15000.	
X1	X4	Smallest	20000.	3000.	1.8
		Most Likely	22500.	4500.	
		Biggest	23000.	6000.	
X2	X3	Smallest	1200.	1200.	2.7
		Most Likely	1600.	1800.	
		Biggest	1750.	2000.	
X2	X4	Smallest	1200.	30000.	2.3
		Most Likely	1500.	50000.	
		Biggest	1750.	75000.	
X3	X4	Smallest	1250.	2200.	1.0
		Most Likely	1325.	2650.	
		Biggest	1550.	3500.	

The Setup cost and The Operating Cost for Each Communications media and for each pair of nodes is:

From	To	By	Setup Cost	Operating Cost
X1	X2	1	195000.	6500.
		2	235000.	9300.
		3	201000.	2100.
		4	125000.	3200.
X1	X3	1	190000.	1000.
		2	250000.	3300.
		3	175000.	2600.
		4	160000.	4500.
X1	X4	1	170000.	550.
		2	145000.	2550.
		3	150000.	3250.
		4	235000.	5450.
X2	X3	1	250000.	2200.
		2	182000.	3600.
		3	175000.	3500.
		4	272500.	3400.
X2	X4	1	225000.	1800.
		2	250000.	3400.
		3	248000.	3600.
		4	252000.	3500.
X3	X4	1	315000.	2200.
		2	225000.	3800.
		3	200000.	3500.
		4	205000.	3100.

Data for a Five Node Design Network Problem

Number of Nodes:	5
Number of Communications Media:	3
Total Setup Cost:	$4,000,000.
Total Operating Cost:	$30,000.
Maximum Allowable Transfer Time between nodes:	2 seconds

Node number 1 is named X1. Latitude: 26 Longitude: 87

Node number 2 is named X2. Latitude: 36 Longitude: 83

Node number 3 is named X3. Latitude: 35 Longitude: 97

Node number 4 is named X4. Latitude: 36 Longitude: 92

Node number 5 is named X5. Latitude: 27 Longitude: 97

Communnications Media number 1 is named Fibre Optics and the maximum transfer rate is 100,000,000 bits per second.

Communications Media number 2 is named Broadband Cable and the maximum transfer rate is 350,000,000 bits per second.

Communications Media number 3 is named Microwave/Satellite and the maximum transfer rate is 60,000 bits per second.

The traffic and file size patterns between each pair of nodes is:

From	To		Traffic	File Size	Distance
X1	X2	Smallest	1200	12000	2.1
		Most Likely	1500	15000	
		Biggest	1800	18000	
X1	X3	Smallest	2200	10000	2.8
		Most Likely	2450	12000	
		Biggest	2500	15000	
X1	X4	Smallest	20000	3000	2.3
		Most Likely	21000	4500	
		Biggest	23000	6000	
X1	X5	Smallest	12000	4000	2.0
		Most Likely	15000	5000	
		Biggest	26000	7500	
X2	X3	Smallest	1200	1200	2.8
		Most Likely	1600	1800	
		Biggest	1750	2000	
X2	X4	Smallest	1200	30000	1.8
		Most Likely	1500	50000	
		Biggest	1750	75000	
X2	X5	Smallest	12000	13000	3.3
		Most Likely	15000	16000	
		Biggest	20000	18000	
X3	X4	Smallest	1250	2200	1.0
		Most Likely	1325	2650	
		Biggest	1600	3600	
X3	X5	Smallest	1300	2300	1.6
		Most Likely	1350	2700	
		Biggest	1600	3600	
X4	X5	Smallest	25	1250	2.0
		Most Likely	75	1500	
		Biggest	100	2000	

The setup cost and the operating cost for each communications media for each pair of nodes is:

		By	Setup Cost	Operating Cost
X1	X2	1	195000.	6500.
		2	235000.	9300.
		3	201000.	2100.
X1	X3	1	190000.	1000.
		2	250000.	3300.
		3	175000.	2600.
X1	X4	1	170000.	550.
		2	145000.	2550.
		3	150000.	3250.
X1	X5	1	445000.	650.
		2	400000.	2950.
		3	415000.	2750.
X2	X3	1	250000.	2200.
		2	182000.	3600.
		3	175000.	3500.
X2	X4	1	225000.	1800.
		2	250000.	3400.
		3	248000.	3600.
X2	X5	1	425000.	1900.
		2	355000.	3800.
		3	361000.	3500.
X3	X4	1	315000.	2200.
		2	225000.	3800.
		3	200000.	3500.
X3	X5	1	75000.	1500.
		2	48000.	2800.
		3	45000.	2900.
X4	X5	1	90000.	1300.
		2	62000.	2700.
		3	58000.	3100.

APPENDIX D

COMPUTER NETWORK DESIGN QUESTIONNAIRES

Questionnaire Administered after Manual Design

NAME: ______________

OCCUPATION: ________________

1. I really feel like I accomplished something.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

2. I don't think I know more about network design than I did before.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

3. The approach taken to solving the network design was very structured.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

4. My network solution was a good one.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

5. It took too much time to solve the network.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

6. I'm pleased with the approach used to analyze the network.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

7. Analyzing the network improved my problem-solving skills.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

8. I wish I had approached the network design differently.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

9. I'm not sure my solutions were appropriate.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

10. Analyzing the network design frustrated me.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

11. I really felt lost in trying to tackle the network design.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

12. The time and effort used to analyze the network design were well spent.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

13. My analysis of the network was systematic.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

14. Analyzing the network design was a useful learning experience.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

15. I may have missed important things in the network design.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

16. Analyzing the network design was interesting.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

17. The approach used to analyze the network design wasn't worth the effort.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

18. I'll be able to handle future problem situations better because of the approach I used to analyze the network design.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

19. I'm not confident about my solutions.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

20. I analyzed the network design in a step-by-step manner.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

21. How many networks did you design before settling on a solution?

22. How much time did you spend?

23. What approach did you use?

Directions:

READ each statement carefully and DECIDE how well the statement describes you.

1. People differ in terms of how much effort they put into jobs. In some jobs people get very involved and spend a lot of effort; in other jobs people exert very little effort. Considering the decision tasks in this network design did you exert:

very little effort	1	2	3	4	5	6	7	8	9	a great deal of effort

2. A number of things can affect the amount of effort we put into a job. How important were the following to you:

a. to show I was capable

very little effort	1	2	3	4	5	6	7	8	9	a great deal of effort

b. my previous experience/or qualifications

very little effort	1	2	3	4	5	6	7	8	9	a great deal of effort

c. the intrinsic satisfaction of doing well

very little effort	1	2	3	4	5	6	7	8	9	a great deal of effort

Questionnaire Administered after DSS Design

NAME:_______________

OCCUPATION:_______________

1. **While using the Decision Support System I felt challenged to do my best work**

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

2. **I felt frustrated by the Decision Support System.**

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

3. **Using the Decision Support System was fun.**

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

4. **I really feel like I accomplished something.**

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

5. **Using a computer to perform network design seems like a good idea to me.**

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

6. **While using the Decision Support System I felt comfortable.**

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

7. **I enjoyed using the Decision Support System.**

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

8. **Even otherwise interesting material would be boring when presented by the computer.**

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

9. I don't like the Decision Support System.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

10. I learned a lot using the Decision Support System.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

11. While using the Decision Support System I had to be at my best.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

12. I don't think I know more about network design than I did before.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

13. The approach taken to solving the network design was very structured.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

14. My network solution was a good one.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

15. It took too much time to solve the network.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

16. I'm pleased with the approach used to analyze the network.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

17. Analyzing the network improved my problem-solving skills.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

18. I wish I had approached the network design differently.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

19. I'm not sure my solutions were appropriate.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

20. Analyzing the network design frustrated me.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

21. I really felt lost in trying to tackle the network design.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

22. The time and effort used to analyze the network design were well spent.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

23. My analysis of the network was systematic.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

24. Analyzing the network design was a useful learning experience.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

25. I may have missed important things in the network design.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

26. Analyzing the network design was interesting.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

27. The approach used to analyze the network design wasn't worth the effort.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

28. I'll be able to handle future problem situations better because of the approach I used to analyze the network design.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

29. I'm not confident about my solutions.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

30. I analyzed the network design in a step-by-step manner.

Strongly Disagree	Moderately Disagree	Slightly Disagree	Neutral	Slightly Agree	Moderately Agree	Strongly Agree

31. What is your current occupation?

32. Have you ever designed a computer network before?

33. Have you ever used optimization techniques before?

Directions:

READ each statement carefully and DECIDE how well the statement describes you.

1. People differ in terms of how much effort they put into jobs. In some jobs people get very involved and spend a lot of effort; in other jobs people exert very little effort. Considering the decision tasks in this network design did you exert:

very little effort	1	2	3	4	5	6	7	8	9	a great deal of effort

2. A number of things can affect the amount of effort we put into a job. How important were the following to you:

a. to show I was capable

very little effort	1	2	3	4	5	6	7	8	9	a great deal of effort

b. my previous experience/or qualifications

very little effort	1	2	3	4	5	6	7	8	9	a great deal of effort

c. the intrinsic satisfaction of doing well

very little effort	1	2	3	4	5	6	7	8	9	a great deal of effort

VITA 2

Angela Eppler Dixon

Candidate for the Degree of

Doctor of Philosophy

Thesis: THE DEVELOPMENT AND IMPLEMENTATION OF A DECISION SUPPORT SYSTEM FOR DESIGNING COMPUTER NETWORKS USING COMPROMISE PROGRAMMING

Major Field: Business Administration

Biographical:

Personal Data: Born in Anadarko, Oklahoma, November 9, 1943, the daughter of Roy Thomas and Delphine Eppler. Married August 6, 1966 to James Peebles Dixon. Two children, James Richmond, born October 27, 1971, and Jennifer Wista, born January 25, 1974.

Education: Graduated from Central High School, Tulsa, Oklahoma, 1961; received Bachelor of Arts degree with a major in Mathematics and a minor in Psychology from the University of Tulsa in July, 1965; received Master of Business Administration from the University of Tulsa in July, 1980; completed requirements for the Doctor of Philosophy degree at Oklahoma State University in July, 1989.

Professional Experience: Instructor of MIS, The University of Tulsa, 1982 to present; Instructor of Management, The University of Tulsa, 1978 to 1982; Teaching Assistant, The University of Tulsa, 1976 to 1978; Lead Analyst, Operations Research Programming Support, Amoco Production Company, 1968 to 1970 ; Scientific Programmer Analyst, Skelly Oil Company, 1965 to 1968.