EXPERT DECISION SUPPORT SYSTEM FOR TWO STAGE OPERATIONS PLANNING

By

TAM Chi-Fai

MASTER OF PHILOSOPHY THESIS

Presented to

The Graduate School

In Partial Fulfillment

of the Requirements for the Degree of

MASTER OF PHILOSOPHY

FACULTY OF BUSINESS ADMINISTRATION

THE CHINESE UNIVERSITY OF HONG KONG

August 1999

ABSTRACT

The joint use of an optimization model and a simulation model has constituted a two-stage approach which has been successfully applied in various operations planning such as hospital layout, freight operations, manufacturing, and defense logistic, etc. In the process, the optimization model determines a macro plan/design based on aggregate information, such as annual demand, average cost, and average utilization. Whereas the simulation model examines the characteristics of the recommended configurations at a micro level by considering the operational randomness and fluctuations. Modeler/decision-makers often have to manipulate the two models iteratively to gradually reach a planning solution. More importantly the decision-maker will have to use the two models periodically/repeatedly with updated data for evaluating, monitoring, or even modifying the existing plan during the planning time horizon. The iterative use of the two models requires both knowledge of the models and expertise of the domain problem. Even if the decision-maker is a modeler it would be very time consuming to manipulate (i.e., updating input data, modifying the models, and re-run the models) the two models. Also, the lessons learned from using the two models may not be able to pass on to the next exercise. Heavy burden is therefore imposed to the decision-maker. This paper presents an Expert Decision Support System (EDSS), which integrates a Decision Support System (DSS) with an Expert System (ES) to alleviate the problem. While the ES inference on the knowledge acquired from the decision-maker and the lessons learned from the previous use of the models, it will also call on the functions from the DSS to manipulate the data and models. We will also illustrate a PC-based EDSS prototype built for a service network planning project for a major air-express courier.

TABLE OF CONTENT

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor, Prof. Waiman Cheung for his continued guidance throughout my postgraduate program. Thanks for Prof. Cheung in leading me from the very beginning of my studies, giving invaluable advice and insight to help me through the completion of this thesis. Without his continued support, the research would not have been successfully completed.

I would like to express my sincere thanks to Prof. Lawrence Leung for his support and guidance throughout my postgraduate studies. Thanks for Prof. Leung in offering me chances to participate in different real-life research projects, especially the distribution network design for DHL HK, which becomes the basis of my thesis and gives me invaluable experience throughout my studies. Furthermore, I would like to thank for his patience and forgiveness to my mistakes made during my learning process. Without his kind consideration, this thesis cannot be completed successfully.

I would also like to express my thanks to Prof. Lee Tien Sheng, and Prof. Lu Ming Te as my thesis examiners, giving valuable comments , which help me to improve my thesis.

I would also like to express my special thanks to all teaching assistants in DSE department, technical staff in BA computer Lab and people who offered their help during my thesis writing. Lastly, but most importantly, I wish to thank my family, Prof. Lee Ching Chyi and Kathy, for their support during the preparation of this work.

CHAPTER 1

1. INTRODUCTION

The complementary use of an optimization model and a simulation model has constituted a two-stage approach which has been successfully applied in various operations planning. The use of this approach requires many skills and expertise from a modeler or decision-maker. We aim to alleviate their burdens from the modeler or decision-maker by proposing an Expert Decision Support System (EDSS) to the two stage planning approach. A PC-based EDSS prototype is built for a service network planning project to illustrate the concept.

1.1 Two Stage Operations Planning

Using optimization models in solving operations planning problem is common and undoubtedly useful. However, the problem size of most practical cases are so big that such models often produce planning design merely based on aggregated information without considering the detail operational characteristics like randomness and fluctuation behavior. In order to examine the operational feasibility and to examine the performance of the recommended configurations, a simulation model can be built for such purposes. The descriptive nature of the simulation methodology allows decision-maker to examine the behaviors of a complex system operating under a probabilistic environment. However, regular simulation model does not offer optimization power and it always works on behalf of a predefined configuration. To uptake the advantages of both optimization model and simulation model, researchers make joint use of the two models. Which constitutes as a two-stage planning approach for operations planning problems. Hence decision-maker can first determine a macro planning design with the optimization model and inspect this design in a simulation model for feasibility and performance tests. Such two-stage approach was proven to be successful in applying to various fields such as hospital layout [Butler *et al.*, 1992], freight operations [Moore *et al*, 1991], manufacturing [Leung *et al.*, 1993], and defense logistics [Nolan and Sovergin, 1972], etc.

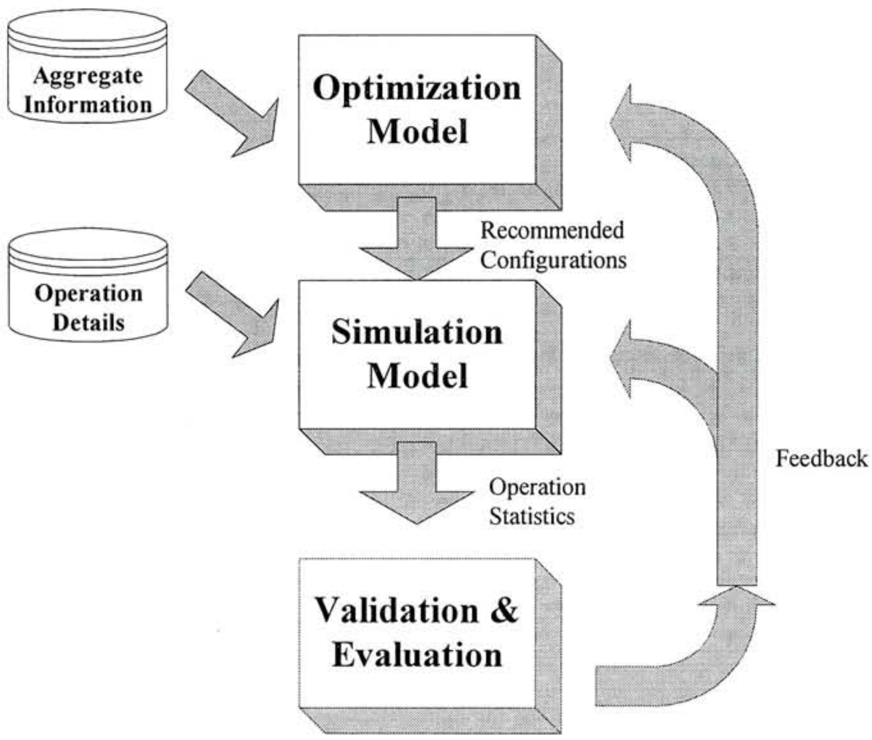


Figure 1-1 Two-stage Planning Approach

While using the two-stage approach, decision-maker often finds that planning is not an 'one-shot' process. The decision-maker needs to go through numbers of iterations before reaching to the final design. In other words, he/she needs to make repetitive use of both models with updated data and parameters throughout the planning process. A problem of using the two-stage planning approach is that the decision-maker not only needs to be a domain expert but also has to be familiar with the manipulation of data and the two models. A person with these expertise is not easy to find. Furthermore, many of the operations planning such as layout planning, and service network design are for a long term (i.e. five to ten years) purpose. The developed configurations should be re-evaluated periodically (e.g. three to six months). The repetitive exercises surely can be benefited from the knowledge learnt and accumulated from the previous planning processes. Therefore, there exists a problem of how this knowledge can be preserved or even be operationalized in the next exercise so that the burden to the decision-maker can be relieved.

1.2 Iterative Activities in the Two Stage Planning Approach

Decision-maker needs to go through numbers of repetitive runs of both optimization and simulation models during the process of fine-tuning the planning design. In fact, for each iteration, running the models require much expertise and effort. Activities including data preparation, model modification, result interpretation and alternatives evaluation, requiring decision-maker to equip with modeling, computing as well as understanding domain specific knowledge about the operations planning.

Data preparations and modification - As the problem size of many cases are huge, input data for the two models are often massive. For each run, data sets needed to be updated, verified and transformed into formats that are compatible to the two models. Such processes are time consuming, prone to errors, and require understanding about the data and models.

Model management and execution - In the process, decision-maker needs to execute and makes modifications to both models for analysis. Such model manipulations require decision-maker to possess good understanding about the models and knowledge on specific model development tools, like the syntax and commands of different software. Users without good understanding on both models or model manipulation skills may find extra burden with the analysis.

Result Validation and Evaluation - Before the decision-maker can finalize the planning design, the operational recommended configurations are validated and evaluated based on a set of pre-determined operational requirements. Such process requires decision-maker's experience or expertise in trade-offs among these criteria, which are often not incorporated in the models.

Sensitivity Analysis - Even if the planning design is validated and has satisfied all pre-determined requirements, it is possible that the decision-maker would like to accomplish more in certain criteria provided that the trade-offs are acceptable. This kind of analysis is highly desirable but requires the decision-maker to work very closely with the models.

In order to alleviate decision-maker from these loads, a computer aided system is proposed to assist the decision-maker throughout the two stage operations planning. The functionality of the proposed system should focus on handling both data and models, and provide intelligence in accessing the recommended results, and offer guidance for decision-maker to explore different planning designs to achieve his/her goals.

1.3 Expert Decision Support System for Two Stage Planning

The use of the two stage planning often requires a decision-maker to efficiently manage data, have a good understanding of both models, and posses the know-how of applying knowledge accumulated during the feedback process. A system which provides the capability of (1) data and model manipulation, as well as (2) offering intelligence guidance during the validation, evaluation and sensitivities analysis would alleviate major burdens from the decision-maker.

In the MIS literature, Decision Support System (DSS) is known to support decision-maker by providing efficient management of both data and models while Expert System (ES) offers recommendations to specific problem by inferencing the acquired domain knowledge. Applications of using DSS and ES were well-established and proven to be useful in supporting various business decisions. To take advantages of both the DSS and ES at the same time, researchers had worked to combine the two systems and called it an Expert Decision Support System (EDSS). Some successful applications of such EDSS can also be found in the literature.

Although the use of EDSS in the two stage planning approach, to our best knowledge, is not found in the literature. It is believed the functionalities of the EDSS well suit the needs of the two stage approach. In this study, we propose to integrate the two and will construct a conceptual architecture for the integration. A prototype EDSS will also be built to based on the architecture to prove the concept.

1.4 Scope of the Study

In this thesis, we aim to design an Expert Decision Support System for the two stage operations planning approach. The system architecture with details of its components, functions and iterations is constructed. A PC-based EDSS prototype system with the major functionalities are implemented for a real life service network planning project. The knowledge acquisition and engineering exercise is done for result validation, performance evaluation, and feedback modification for the project. While the domain knowledge in the ES is different from case to case. We believe the architecture is generalizable without further verification in this study.

1.5 Organization of the Thesis

Chapter 2 provides a detailed literature review on the ES & DSS and the integrative use of optimization and simulation model. The details of using prototyping methodology for the study are illustrated in chapter 3. An architecture that describes the integration of the Expert Decision Support System and the two stage planning approach is presented in chapter 4. A prototype EDSS for real life project is described in chapter 5 to illustrate how the proposed system works in a two-stage planning approach. While the system analysis and evaluation is provided in chapter 6. Finally, chapter 7 gives the conclusions of the study.

CHAPTER 2

2. LITERATURE REVIEW

The literature review in this chapter includes three parts: (1) review on air express service network design, (2) previous research works done on the integrative use of the optimization model and simulation model, and (3) review on the integration of Expert System (ES) and Decision Support System (DSS) and their applications.

2.1 Network Design for Air Express Service

The general topic of service design, or network design has been widely researched and documented [Magnanti and Wong, 1984]. However, limited research works are found on the field of air express courier service. Chang and Ponder, [1979] pointed out several characteristics of the air express courier industry by using Federal Express Corporation (FEC) as an example. One factor which air express service, like FEC, over other traditional air freight service company, was the centralized operations and the hub-and-spoke concept. Research works had investigated the consequences of such network structure in the field of airlines [Kanafani and Ghobrial, 1985], general transportation [O'Kelly, 1986], air transportation system [Aykin, 1995], express service network design [Barnhart and Schneur, 1996], etc. Among the express services network design, most of the studies focused on the planning of the network, but few of these network designs included a detail operational analysis of the developed network.

On the other hand, efforts had shown on the use of simulations in the operation analysis under the field of air express and similar services. Research work are found on the topics of simulation and statistical analysis of vehicle routing with timing constraints [Cook and Russell, 1978] and the air terminal design [Cook and Rao, 1985]. However, many of the researches based on a deterministic result of the network design. Simulation model, in most cases, acted to analysis the operations of the network, but with no feedback to the design of the network.

In most of the research works on hub-and-spokes network design, few effort tried to combine other models to validate or even evaluate the efficacy of the developed solution on an operational level. Kamoun and Hall, [1996] demonstrated a network design of express mail service with an analytical model together with a simulation model. The analytical model tried to determine the number of hubs. In a feeder backbone network, together with their locations and the routings schedules of the pickup vehicles in a single time period. On the other hand, the simulation model concentrated on the operational analysis of the designed network by simulating the customers' call-in and pickup activities to analysis the two major components in the developed network, normally "feeder" and "backbone".

Such combinatory use of the simulation model with the optimization model enables decision-maker to examine the developed service network in a micro and specific aspect, which gives information for further improvement throughout the whole planning design. In fact, the joint use of an optimization and simulation model was widely applied and proven to be successful in many operations planning problems.

2.2 Integrative Use of Optimization and Simulation Model

When confronted with a system to be modeled, analysts usually think first of linear programming or other optimization techniques. However, these methods are often rejected as the "richness of detail" of the system is difficult to achieve within the realm of optimization model. Typically, the analysts then turn to simulation as the only alternative offering the desired richness of detail. In fact, the independent use of the optimization model in solving operations planning problem and the use of a simulation model to perform operational analysis is common and undoubtedly successful. However, due to the nature of some problems, the mere use of either model cannot yield satisfactory result. To capture both the advantages of the optimization and simulation models, researchers have make the complementary use of both. Such planning approach has been successfully applied in various fields. Hueter and Swart, [1998] developed a labor-management system with a forecasting model, an integer programming model, and a simulation model to solve the labor management problems. Sengupta, [1995] used an integer programming model and a simulation model to determine the optimum capacity of a food manufacturing environment and its future growth. Wellons *et al.*, [1994] made the joint use of an optimization system together with a simulation model to optimize the operation of the power plant. Leung *et al.*, [1993] provided a linear integer model together with simulation experiments used in FMS design. Moore *et al.*, [1991] built a mixed integer programming and simulation models to select and deploy carriers. Among these studies, two models were related but independent. Optimization model was used for macro and long term planning which rely on the analysis of average and aggregate behavior, while simulation was used to describe the system behavior account accurately for micro and operational level. In many of these studies, simulation focused as a tool to evaluate the performance of developed solution, and few of them emphasized on how the results from one model provides feedback to another during the whole decision making process.

In fact, researchers had addressed the feedback in the complementary use of two models. Nolan and Sovereign, [1972] demonstrated how a large analysis could be parsed into two separate "macro" and "micro" analysis by using a recursive approach. The recursive approach tried to divide the complementary use of two models into three steps:

(1) With estimated productivities and other parameters, determine the schedule of inputs which maximizes the values of outputs within resources levels available,

(2) For the schedule determined at step 1, simulate and test if the parameters are appropriate and can the detailed matching be performed at a desegregated and discrete level,

(3) If the parameters are not appropriate at step 2, revise the parameters and back to step 1.

Such recursive approach was successfully applied in other operations planning problems. Carlson *et al.*, [1979] applied this approach in analyzing the outpatient health care settings problem. By using the patient queue time resulted in the simulation runs, the study demonstrated how the optimal settings of health care providers were determined with a reasonable patient wait time. On the other hand, Butler *et al.*, [1992] described a two-phase recursive approach in hospital layout problem that incorporated an integer goal programming model and a detail simulation model. With similar planning approach, [Leung and Cheung, forthcoming] applied an integrative methodology to design a distribution network planning for the courier express service in our case study. A mixed integer programming model together with a SIMAN based simulation model was used to determine a ten-year horizon courier express service network, and to investigate of its operational performance.

Although the joint use of the optimization model and simulation model had proven to be successfully applied in dealing with various operations planning problem, most of these studies with such two stage approach need to undergo considerable numbers of iterations before a satisfying solution can be obtained. Moreover, due to the size of most problems, these iterations were time consuming and required heavy data and model manipulations. In some cases, decision-makers needed to accumulate heuristics and expertise during these iterations so as to examine the relationship between the performance measures and the input parameters for feedback process. Consider the network design problem in our case study, the problem included massive data and complex models, in which iterations between two models were time consuming and required domain specific expertise. Besides, as the developed network needed to be re-evaluated from time to time, such iterative activities demanded a considerable time and efforts. As a result, a more systematic and automated method should be developed in order to assist the decision making process. A system which could manipulate the use of data and models in a rapid and flexible way, together with intelligence in guiding the decision-maker during the operations planning, suits best to assist the design of the distribution network in our studies.

2.3 Expert System & Decision Support System

Decision Support System (DSS) and Expert System (ES) had been widely used in many fields in solving different types of problems. The DSS tries to support the decision-maker by providing a rapid and interactive manner to manipulate data and models while the ES provides suggestion to domain specific problems by its inference engine and knowledge base. Although the two systems although both aim at providing support to the decision-maker in solving problems, they have different characteristics, functions and approaches in solving problems.

2.3.1 Expert System

By the 1970s, it became apparent to the artificial intelligence community that inferences or strategies alone, even those augmented with heuristics were often inadequate to solve real life problems. These problems were so complex that, without the addition of more knowledge about the problem area, it was impossible to obtain the result. It also became apparent that for many problems, expert problem area knowledge was more important than the inference and strategies used to manipulate that knowledge. Such realization gave birth to the field of knowledge engineering, which focuses on "how" to bring expert knowledge to bear in problem solving. In particular, this has led to systems that are able to reason about inferences as well as about answering to current problems. The general definition of the expert system emphasized on the "the application of human expertise" in solving problems. And it is defined as a system that employs human knowledge captured in a computer to solve problems that ordinarily require human expertise.

Characteristics of ES

An expert system seeks to mimic the behavior of a human expert in applying knowledge to a specific task. Typically several features allow an ES to do this.

Three Levels of Knowledge Organization - In an expert system, the problem solving model appears explicitly as a knowledge base rather than implicitly as a part of coding, and the knowledge base is manipulated by a separate, clearly identifiable control strategy. Comparing ES with ordinary computer system, ES organize knowledge on three levels: data, knowledge base, and control [Yaghmai, 1984].

Explanation Capability - One unique feature of expert system is its ability to explain its advice or recommendations and to justify why certain action is suggested or not. Such features are treated as an essential function in an ES. "An expert system must be able to explain its line of reasoning to the users" [Keim and Swart, 1986]. In most of the ES, the explanatory action is done by a subsystem called justifier or explanation subsystem. Through out the decision process, users can ask for explanation for the suggested solution or query for why certain question is asked by the system.

Handling Uncertainty - Besides having an explanation facility, most ES can handle uncertainty. In real life, an expert is not right or 100 percent certain and usually factors a measure of uncertainty into his or her answers. Similarly, an ES typically has a mechanism for handling uncertainty in the set of facts and heuristics to be used and for allowing the user to enter a degree of uncertainty when using the ES [Rich, 1983].

2.3.2 Decision Support System

Decision support system (DSS) was first introduced as a concept in early '70s by Scott-Morton under the term "Management Decision System" [Scott Morton, 1971], in which firms and scholar which later categorized this concept as "an interactive computer-based system which helps decision-maker utilize data and models to solve unstructured problems".

Characteristics of DSS

The definition of the DSS was proven to be so restrictive that only a few actual systems can completely satisfied it. DSS, like MIS and ES, is actually content free expression, which has no universal accepted definition. However, most of the DSS are having the follwing characteristics.

Incorporate both data and models - the major function of DSS is to help the decision-maker to access the relevant data and information together with the help of some models for analysis. As a result, one of the characteristics is that, DSS tries to incorporate the use of both the models and data during the decision making process.

Focus in solving semi-structured task - DSS aims at providing less structured and under-specified problems that upper level managers typically face. These problems are not easily solved by a mere computer system such as EDP or MIS, nor by management science.

Support decision making rather than making decision - Unlike expert system, one major characteristic of DSS is that the system itself does not make any decision or judgment. It always "supports" the decision-maker to make the choice rather than suggesting decision. In fact most of the DSS are designed to help the decision-maker to solve problems by bringing "human judgment" and "computer information" together.

Designed to be user-friendly which enable interactive use - Another major characteristic of the DSS is that it can help decision-makers to make decisions in an interactive and flexible way. In which the effectiveness of decision making is highly concerned. As a result, most of the DSS is designed with a user-friendly interface.

In conclusion, a DSS can help decision-makers utilize data and models to solve unstructured problems in a rapid and flexible way, while the ES mimics an expert to provide solutions and guidance in specific problems. In order to assist the network design in our case, both advantages of DSS and ES need to be included. One way of including both advantages of the DSS and ES is to build a system by integrating the two systems.

2.3.3 ES / DSS Integration

In the early 80s, most of the ES and DSS were not integrated. ES operated as independent expert consultation systems while DSS operated as support devices to decision-maker. However, researchers found that there were potential benefits in bringing the DSS and ES together to solve the problems. "In certain problem domains both ES and DSS may have distinct advantages that, when combined, can yield synergetic results" [Turban, 1988]. Today, the integrative use of the DSS and ES are still active. Artificial Intelligence, heuristics and quantitative models are designed to integrate in ES / DSS integration models. Many implemented systems and numerous systems prototypes have been developed using this concept.

ES / DSS Integration Models

Many studies had drilled in the advantages of the synergy between the DSS and ES. Turban tried to address the logic and benefits in integrating the DSS and ES, and proposed two alternatives of ES / DSS integration model [Turban, 1988].

- ES attached into different DSS components
- ES as a separate component in DSS

Specific ES are proposed to attach with different DSS components in the first model (Figure 2-1). Function of the different parts in the DSS are enhanced by introducing ES in the respective components. An ES attached with the database management subsystem can enhance the data abstraction power of the DBMS. For example, query like "display all student who fails in the exam", or question like "why John get the scholarship?" may be raised. ES integrated with the model base management subsystem can provide guidance to the manipulation of the models and ES connected with the interface can improve the flexibility of the system.

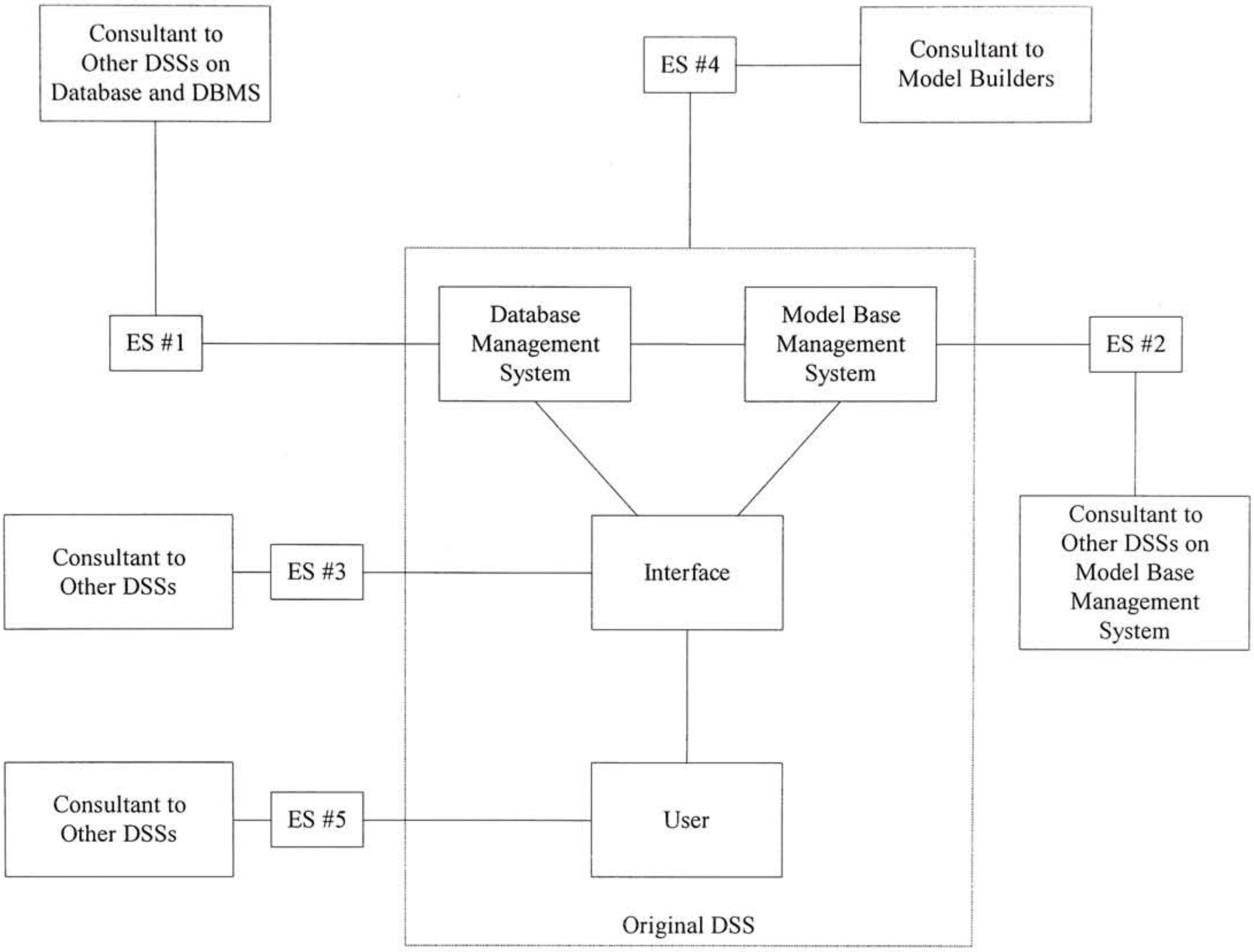


Figure 2-1 Integration of ES into all DSS Components

In most of the decision making process, the decision-maker may need to identify the nature and category of the problem and select the appropriate tools or models in the DSS for analysis with specific expertise. On the other hand, the computerized quantitative analysis provided by the DSS may be directed to a group of experts for the purpose of evaluation before making decision. These activities which require domain specific knowledge were proposed to be done with the help of an ES. The second proposed models try to integrate ES as an additional component in the DSS to help these activities (Figure 2-2). Output of the ES may serve as the input of the DSS, or vice versa.

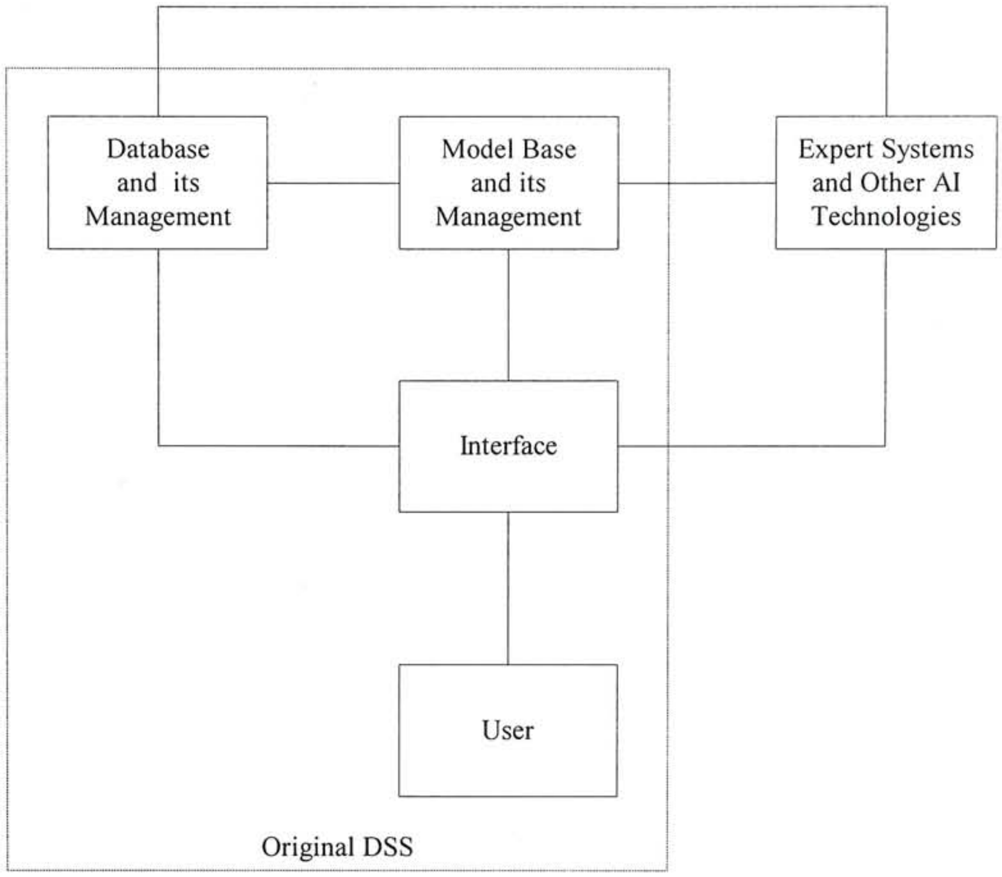


Figure 2-2 ES as a Component of DSS

These two theoretic models were later reviewed and modified by researchers. New models of ES / DSS integration is proposed. Expert Decision Support System (EDSS), Intelligent DSS (IDSS), and Knowledge Based DSS (KBDSS) were raised according to similar ideas in ES / DSS integration. Bidgoli, [1993] introduced his Ideal DSS model, with the idea of the second model of Turban. Instead of integrating an ES as an additional component in the DSS, this model tried to redesign the components of the system by integrating ES components into different part of the traditional DSS. On the other hand, El-Najdawi, [1993] proposed another ES / DSS integration model which named as Expert support system (ESS). This model brought in the idea of Turban first model, in which multiple expert systems were added into different components of the DSS.

Research works on case implementations were found. Jungthirapanich and Benjamin, [1995] illustrated the design and implementation of an expert decision support system for facility location problem. The developed system incorporated an expert system in a traditional DSS, the ES of which served to elicit user's needs via a friendly interface, and inferenced the location models suitable for the analysis. Moore *et al.*, [1992] conducted another case study that developed a prototype expert decision support system for the market appraisal of real estate. Among the two conceptual models of ES / DSS integration, implementations in the field of operation planning problem tended to concentrate on the later one. In most of the ES / DSS integration, two systems tended to be used separately in dealing with different parts of jobs, few of them tried to design the system by incorporating functions and components in an integrated manner. In this paper, we try to borrow the second conceptual model by Turban as our methodology to design an EDSS for the two stage operations planning approach. The proposed system will focus on providing a rapid, flexible and intelligent way to manipulate data and models, hence to relief the loads of the decision-maker during the network design process.

CHAPTER 3

3. RESEARCH METHODOLOGY

In this study, we try to approach our problem in four steps: (1) review on Expert System (ES) and Decision Support System (DSS) integration models, (2) design and construct system architecture, (3) prototyping, and (4) analyze and evaluate the developed system.

3.1 Review on DSS / ES Integration

The integration of ES/DSS has been successfully applied in various fields. Throughout these studies, developers have tried to design their systems by capturing both the advantages of ES and DSS in different ways. Turban, [1988] pointed out two fundamental ES/DSS integration models: (1) ES integration into DSS components (Figure 2-1), and (2) ES as a separate component in the DSS (Figure 2-2).

In fact, the application of the ES/DSS integration can be divided into two categories following the model proposed by Turban. For the first model, the integration of ES aimed to enhance the function of particular components in the DSS. For example, integration of an ES to the database system in a DSS adds reasoning capability to the operation of the Database Management System (DBMS). Such integration enables users to perform higher level queries such as asking 'why' or 'how' questions. Besides, studies have focused on the intelligent of selecting, revising, and developing models in a DSS. Integration of an ES with the Modelbase Management System (MBMS) serves such purpose. For example, an ES stored with the knowledge to interpret the user's problem can be integrated with the MBMS to assist decision-maker in selecting/developing the appropriate model(s) for analysis.

Another area which may improve the quality and user-friendliness of the existing DSS would be the integration of ES capability into the dialogue component of DSS. Examples of application features like explanation capability of DSS, symbolic presentation and native language presentation, etc are categorized as this type of integration. According to Turban's first model, the integration of ES in DSS components could be applied independently or as combinations of these three. Hence different ES can be acted in serving specific enhancement for a particular component in DSS.

On the other hand, Turban's second alternative for ES/DSS integration is to add an ES as a separate component in the DSS. He pointed out such model would best suit for a design which needs both ES and DSS functions, together with two systems working independently but related. For example, output results from the DSS may be directed to an ES for evaluation or ES may first be used to conclude the importance/category of the problem and then directed to the DSS for analysis. According to this approach, ES can complement DSS in one or more steps in the decision making process. Such integration may be visualized as the use of ES to play the role of a human expert which the user can call upon when in need of expertise in strategy formulation like interpretation and evaluation of information/results.

Following the idea of this approach, we adopt the methodology of Turban's second model and design an Expert Decision Support System (EDSS). The system consists of both ES and DSS functions which work independently but related. DSS is responsible for the both data and models manipulation for analysis while the ES act as an expert to provide expertise for decision making process during the two stage planning iterations.

3.2 System Design

The objective of the proposed system is to provide assistance in both data and model manipulation and intelligence guidance throughout the two stage planning. Our proposed EDSS is designed in a way to serve such purpose. With respect to the characteristics of DSS and ES, we designate different functions to the two subsystems. DSS, which is characterized by its functions to help decision-maker in solving problems by utilizing both data and models, is responsible for data management and model manipulation during the planning process. While ES, which is characterized by its ability to capture and operationalize knowledge in solving specific problem, serves to assist decision-maker by providing intelligence during feedback process.

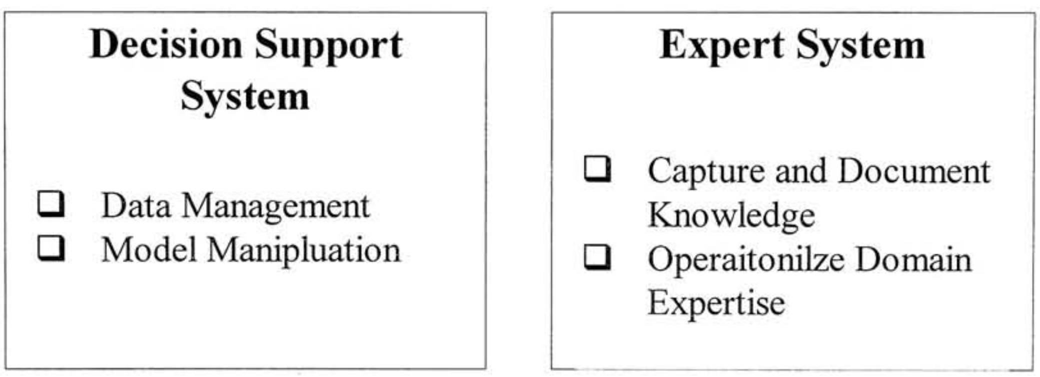


Figure 3-1 Functions of DSS and ES

Figure 3-1 summarizes the functions of the proposed system. DSS here focuses on the data management and model manipulation for analysis, while ES captures, documents and operationalizes the knowledge acquired during the planning. Two system components are independent but working together to assist the operations planning design process. Result from the DSS will be directed to the ES for interpretation. Based on the ES / DSS integration model by Turban, the proposed EDSS brings the functions of both DSS and ES together by integrating the ES as an extra component in a DSS. Figure 3-2 presents the overview of the architectural design of our proposed system. Decision-maker can directly manipulate data and model via the DSS or with the guidance of the ES. The ES component interacts with the DSS, which plays the role of an expert guiding the use of the DSS throughout planning design process.

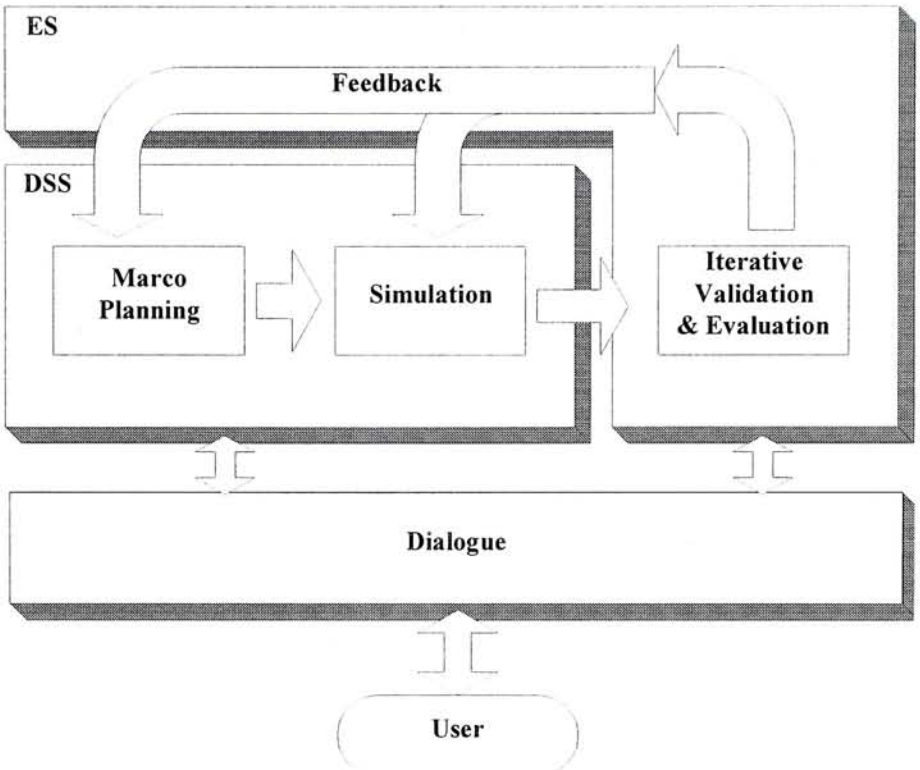


Figure 3-2 System Design Overview

3.3 Prototyping

To demonstrate the design of our Expert Decision Support System, a prototype system is built based on an express service network design project to illustrate how the proposed system is operationalized in an operations planning problem. Besides, it serves to demonstrate the functions of data and model manipulation and intelligence guidance in the two stage planning. Development of the system prototype will be divided into two parts. The first part consists of a database system and a model base system which is responsible for the role of DSS. For the second part, we will go through to the details of the knowledge engineering of the ES in our case study.

3.4 Analysis and Evaluation

Prototyped system is implemented and tested with our case study scenario. Illustration of functions of the prototyped system is performed based on a selected network planning scenario. Results are listed to compare and contrast the advantages of using our proposed system versus the existing manual approach during the planning design. In this exercise, we evaluate the pros and cons of introducing the EDSS in the network planning project and hence generalize the use of EDSS in the two stage operations planning approach.

Chapter 4

4. System Architecture and Knowledge Modeling

4.1 Architecture Overview

During the two stage operations planning, decision-makers need to repetitively manipulate both data and models. In most cases, these data are huge in amount and need to be updated, verified and transformed into specific formats as input for the two models. Also decision-makers need to have good understanding of the models as well as modeling syntax or language in order to execute and modify the models. Moreover, decision-makers need to posses good domain knowledge and expertise to interpret and evaluate the results generated from the models, so that the decision-maker can provide feedback for appropriate modifications to gradually come to a satisfactory solution. In order to offer assistance to decision-makers, the Expert Decision Support System (EDSS) is proposed to focus on providing data and model manipulation together with the intelligence guidance throughout the operations planning design.

The architectural overview of the EDSS is shown in Figure 4-1. The EDSS is composed of two subsystems: a Decision Support System (DSS) and an Expert System (ES). The DSS is responsible for both data management and model manipulation in the operations planning while the ES functions to capture and reapply the expertise, heuristics, and experience accumulated to assist the decision-maker throughout the planning design process. Two subsystem are working independently but related. In a typical two stage operations planning, the decision-maker can either manipulate models via the guidance of the ES subsystem or directly execute and modify data and models by the DSS.

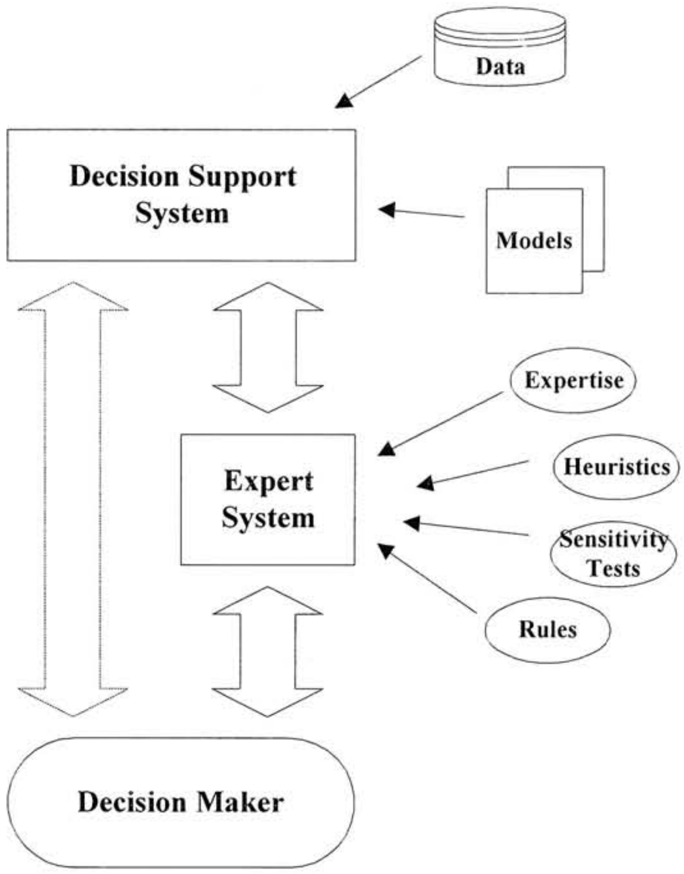


Figure 4-1 EDSS Architectural Overview

DSS performs data management, by a database management system to collect, update, retrieve and append all data sets need for the model runs, while the execution and modification of the optimization and simulation models is done by a model base management system. Hence, decision-makers are unloaded from manipulating both data and models directly. ES on the other hand, receives the planning results generated from the DSS and performs inferences for feedback and modifications. A set of rules in describing how the result are interpreted, validated and evaluated are stored in the form of knowledge base and used to make inferences for the operations planning. Thus, the ES subsystem can mimic an expert in validating and evaluating the generated results according to accumulated expertise and heuristics. As a whole, the two subsystems act interactively to help the decision-maker in making an operations planning design.

4.1.1 System Architecture and Interactions

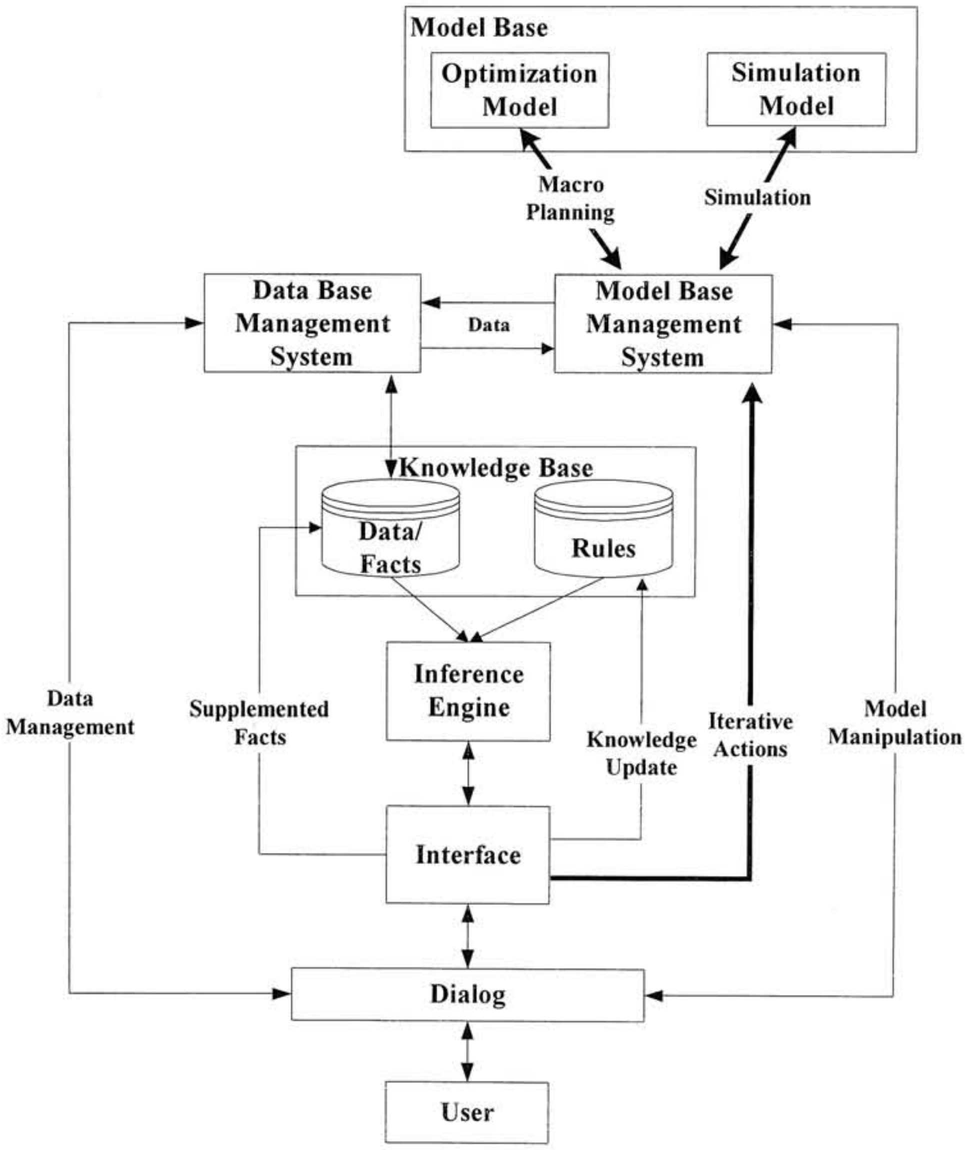


Figure 4-2 System Architecture and Interactions

Figure 4-2 shows the detail system architecture of the EDSS together with the system components interactions. Throughout the operations planning process, the decision-maker can either manipulate data and models via the guidance of the ES interface or to make a direct manipulation with the DBMS and MBMS respectively. Database subsystem which stores the data sets for analysis can cope with the MBMS during the macro planning and simulation. Appropriate data is transferred to the MBMS for model execution, while the results are send back to DBMS for storage and further evaluation. In the evaluation and feedback process, these results are passed on to the Expert System and referenced as part of the knowledge base. By using these facts together with the predefined rules, the ES performs the validation and evaluation with the inference engine. Concluded recommendations are finally passed to the interface, where it is displayed to the decision-maker and transferred back to the MBMS in the form of various recommended actions.

4.1.2 Decision Support System

Major functions of the DSS focus on providing data and model manipulation. Throughout the operations planning design, decision-makers need to retrieve, store and update different types of data and results in running of models. DSS here provides the data management function by storing all necessary types of information and data with a Database Management System (DBMS). On the other hand, the iteratively use of optimization and simulation model requires the repetitive parameters modifications and execution of both models. DSS offers the model manipulation by a set of programs and subroutines to perform specific functions call.

The structure of the DSS subsystem is shown in Figure 4-3. The DSS subsystem is similar to a traditional DSS which composes of four components: a Database, a Database Management System (DMBS), a Model base, and a Model Base Management System (MBMS). During the macro planning and operations simulation, the MBMS communicates with the DBMS to request relevant data as the input of the optimization and simulation model, while the results generated by the models are send back from the MBMS to the DBMS for storage.

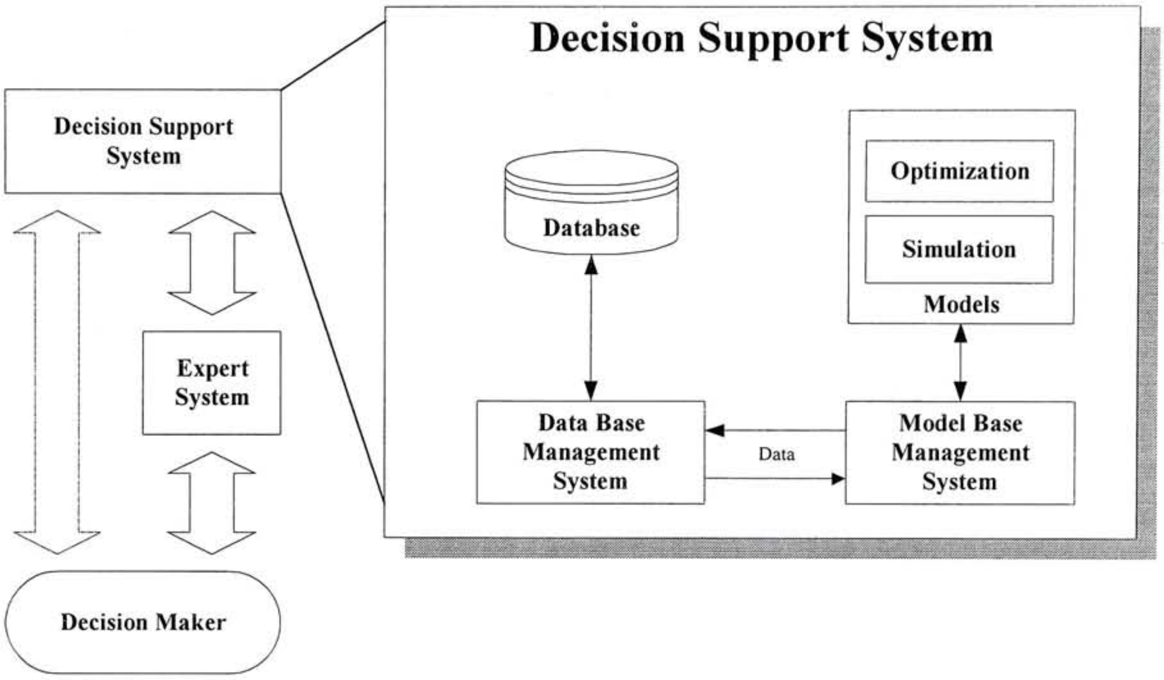


Figure 4-3 EDSS Subsystem - Decision Support System

Database Management System

The DBMS is one of the important components in the EDSS. It helps to provide the data required to build, use and maintain the models. The output from the models is stored in the database, making the results accessible to other models and hence allowing integration among models. In most of the operations planning, data and information required for the planning are huge, comprehensive, and may exist in different formats. Decision-maker may need to go through many pre-processing to make the data available as the model inputs. Moreover, throughout the planning horizons, data and information need to be updated from time to time. The series of generated results and iteration history are needed to be managed in a systematic way. DBMS provides the standard data management functions throughout the operations planning. The model data, generated results and iteration activities are stored in the database which can be retrieved, updated or append via the DBMS. In the operations planning, DBMS provides a direct manipulation channel for the decision-maker to manage the data, results and other information. Besides, the DBMS cooperates with the MBMS to retrieve model data and store planning results in the database during the operations planning.

Database

The database stores all the data, information and activities logs during the operations planning. Data stored in the database is divided in three categories:

Model data – this refers to the data sets used as the inputs for the optimization model and simulation model. This includes the aggregate information as the inputs for the optimization model together with the operational details used in the simulation model. Data are either stored in separate set of files or in forms of tables in the DBMS.

Result output – this refers to the intermediate results obtained from the output of two models. During the planning process, a set of results are generated in each iteration, including the configurations and simulation statistics. These results are stored in the form of database which serves as references in the knowledge base during the feedback process.

Iterations history – this refers to the history and activities logs for the iterations. Throughout the operations planning, two models may be re-run with different parameters. Iteration history records the parameters used, iteration results and respective action for each iteration stored that can be used for future reference.

Model Base Management System

In a two-stage operations planning, the decision-maker needs to come up with a satisfactory solution by repeatedly manipulating the two models. Frequent updates, modifications and re-runs need to be performed for data analysis. Decision-makers need to possess background knowledge of both models and the hands-on techniques of manipulating the models. Even if the decision-maker is a modeler, it would be time consuming in performing such activities. The MBMS is the component that is responsible for manipulating all models stored in the Model Base by providing two major functions: (1) model execution and (2) model modification.

Model Execution – The decision-maker needs to iteratively use the two models during data analysis. MBMS pre-programmed a series of functions which call for the execution of different models stored in the model base and communicates with the DBMS to request for the appropriate data and transform the retrieved data into input files in the specific format for the model to run. Hence the decision-maker can be unloaded from the complicated syntax for different model execution.

Model Modification – Besides model execution, model modification is an important activity for analysis. For example, in a facility layout planning design, after evaluating results from several runs, the decision-maker may find that he/she needs to change the planning criteria in the optimization model due to some management issues. In another scenario, the decision-maker would like to introduce new policies for the layout planning. In these cases, the modeler may need to add or change some of the constraints in the model so as to incorporate the changes. For decision-maker who has no idea about the models, such modifications will become a harsh task. Even for the modeler, such activities may be time consuming and prone to error. MBMS pre-programmed a set of function modules, which provide different program subroutines for specific modifications. As a result, decision-makers can modify the model by means of the appropriate function calls or programs, which assist decision-makers by providing a rapid and user-friendly interface in model manipulations.

Model Base

Models in the EDSS provide data analysis capabilities. For operations planning, various types of models are commonly used, such as linear programming models, integer programming, goal programming, simulation models, and regression models, etc. Model Base acts as a component to hold different types of models for the different planning design, and hence managed by the MBMS for execution and modification. In a two-stage operations planning, decision-makers combine the use of an optimization model together with a simulation model to design the planning solution which is stored as part of the model base in the EDSS.

Optimization Model - This refers to the model which determines the macro optimal planning solution based on aggregate information. For an example, in a hospital layout design, Butler *et al.*, [1992] used a quadratic goal programming model to determine the optimal hospital layout configurations with the deterministic variables like the numbers of equipment to be installed, capacity of each room, and the location of these rooms, etc. based on a defined objective and a set of constraints. In most cases, because of the complexity and the sizes of the problems, mathematical models can hardly include all the issues concerned, and can only focus on determining an optimal planning based on a macro level.

Simulation Model - This refers to the model which examines the configurations computed by the optimization model by considering the operational factors at a micro level. As the mere use of an optimization model in operations planning limits the solution at the aggregate level, it is necessary to study the operations of the developed configurations at the operational level. The simulation model validates the feasibility and the performance by considering most of the fluctuations, randomness and dynamics of the actual system.

4.1.3 Expert System

The major functions of the ES focus on providing intelligent guidance throughout the planning design. Other than manipulation of the data and models, the decision-maker needs to accumulate knowledge and expertise to interpret, evaluate and provide feedback to the result obtained from the data analysis. ES subsystem here mimics an expert in making inferences and drawing conclusions during the process of validation and evaluation with the expertise acquired.

The structure of the ES subsystem is shown in Figure 4-4. The ES subsystem composes of three components: a Knowledge Base, an Inference Engine, and an Interface. During validation and evaluation, the ES requests data and analysis results from the DBMS and stores them as facts in the knowledge base. These entries together with the predefined rules form the knowledge base, is used to makes inferences by the inference engine to interpret, validate and evaluate the planning design. Hence the ES subsystem can draw conclusion and recommended feedback action during the planning design process. The interface here uptakes the prompt from the decision-maker and initiates the result back to the user as well as the DSS subsystem for specific action.

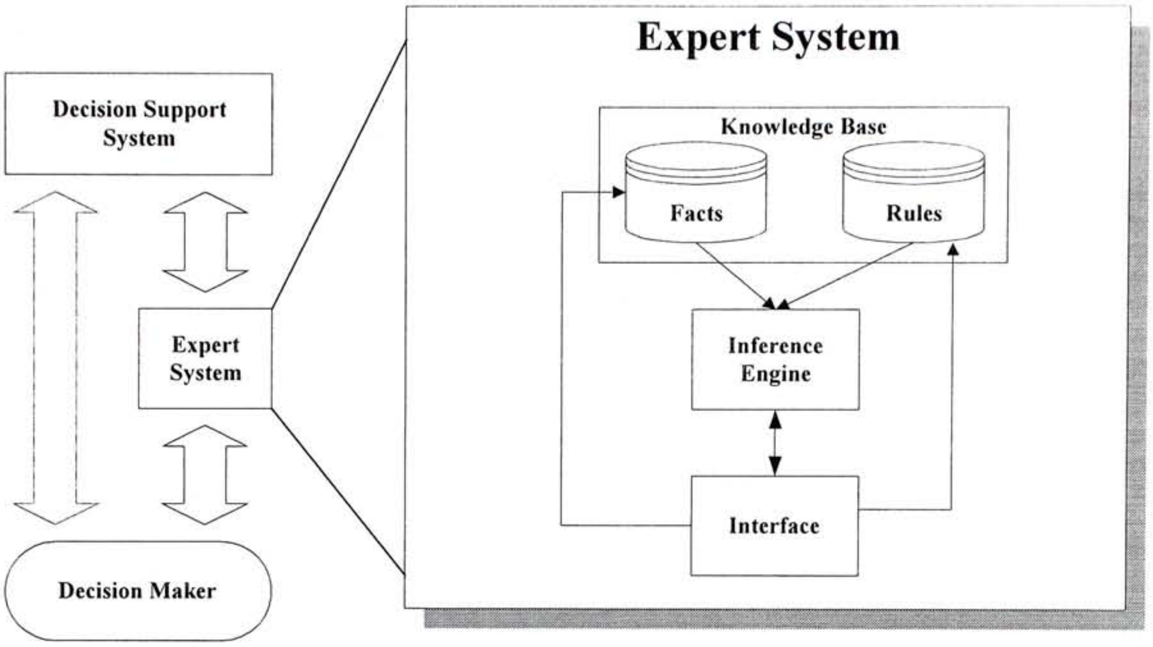


Figure 4-4 EDSS Subsystem - Expert System

Knowledge Base

In the ES subsystem, the knowledge base is a component which stores up the necessary knowledge for interpreting, validating and evaluating the operations planning problem. It captures all the expertise and the knowledge accumulated for the operations planning development. Knowledge is stored in the form of knowledge base entries which are problem specific. In fact, we categorize the knowledge base entries in the form of two basic elements: Facts and Rules.

Facts – This refers to the information defined by the decision-maker and the data analysis result generated by the models. Knowledge entries are referenced from the DBMS in the DSS subsystem, predefined by the decision-maker/modeler or resulted from the prompt response by the decision-maker. For example, in a validation process, the intermediate results of planning configurations and simulation results are referenced as the fact in the knowledge base, while the system parameters for validation is initiated in the knowledge base by the decision-maker.

Rules – These are the knowledge base entries used to make inferences in determining the appropriate feedback action. By eliciting the expertise and heuristics from field expert, the knowledge in describing how to interpret, validate and evaluate the planning design is stored in these form of different sets of "If-Then" rules. For example, in the evaluation process, a set of rules is used to determine how the performance is evaluated based on a numbers of measures.

Inference Engine

Inference engine acts as the "brain" of the ES, in which it is known as the control level in the ES subsystem. It functions to make decisions about how to use the knowledge base entries by organizing and controlling the steps taken to solve the problem. The inference engine in the ES subsystem incorporates a backward chaining method aiming at working out different goals in the planning design. A goal in the planning design is referred as a decision or conclusion during the feedback iterations. For example, the decision-maker needs to determine the operational feasibility of the developed configuration before going on to evaluate its performance. A series of facts, which refer to the results from the simulation runs and developed configurations, together with a set of reasoning and heuristics, are used to determine whether the configurations are operational feasible. Inference Engine in the subsystem acts to make inference of these facts based on the sets of rules in the knowledge base and draw conclusions to goals, hence determinig the appropriate actions and feedback.

Interface

The interface is one of the most important components in the ES subsystem. It is a means to provide communication channel between the ES subsystem with the DSS subsystem and decision-maker. In the operation planning, the interface either receives prompt response from the decision-maker, or displays the reasoning, results and conclusion during the validation and evaluation stages. Moreover, the interface is responsible for passing the concluded recommendation/action resulted from the inference engine, back to the MBMS in the DSS subsystem for respective actions, such as configurations redesign, data update or simple model modification, etc. Hence, the iteration between data analysis and evaluation can be automated with the help of the interface.

4.2 System Operations

In a typical two stage operations planning, the decision-maker starts with a macro planning design followed by a simulation. Results generated are then validated, evaluated and feedback with appropriate modification for next iteration, until the decision-maker comes up to a satisfactory planning solution. The EDSS divides the decision process into four steps: (1) data collection and management, (2) model manipulation and analysis, (3) results validation and evaluation, and (4) feedback and concluding recommendation. Figure 4-5 illustrates the operations flow and the system interactions.

4.2.1 Operations Flow

Data Management - In the operations planing, various data like aggregate information and operations details are to be collected and manipulated to be used as the input of the optimization and simulation model. EDSS collects all the relevant data and stored in the DBMS in forms of database. Hence these data sets can be easily updated, modified, and retrieved during the operations planning. These data are later requested by the MBMS for analysis, or updated via other function calls.

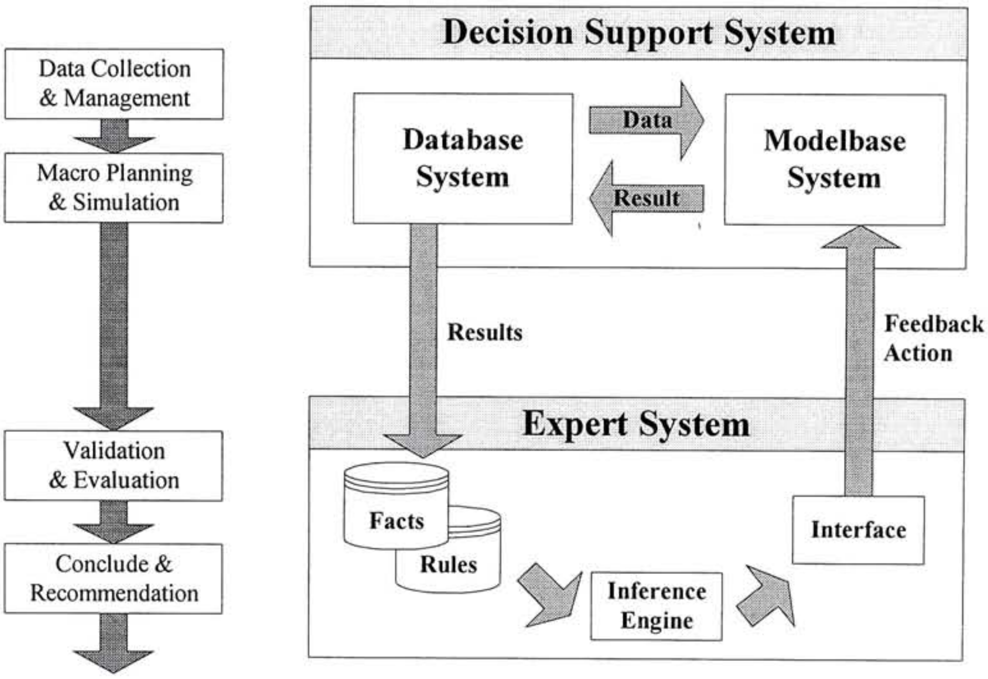


Figure 4-5 System Operational Diagram

Model Manipulation - For the macro planning and operational simulation, optimization model and simulation model are run respectively. MBMS receives either direct manipulation from the decision-maker or recommended actions via the ES subsystem responsible for the model execution. Sets of program modules, in forms of program subroutines, is used to perform specific model manipulation functions, like model execution and model modification. Besides, MBMS communicates with the DBMS to perform data retrieval and result storage. As a result the appropriate data can be retrieved for the model inputs while the generated results can be referenced during validation and evaluation.

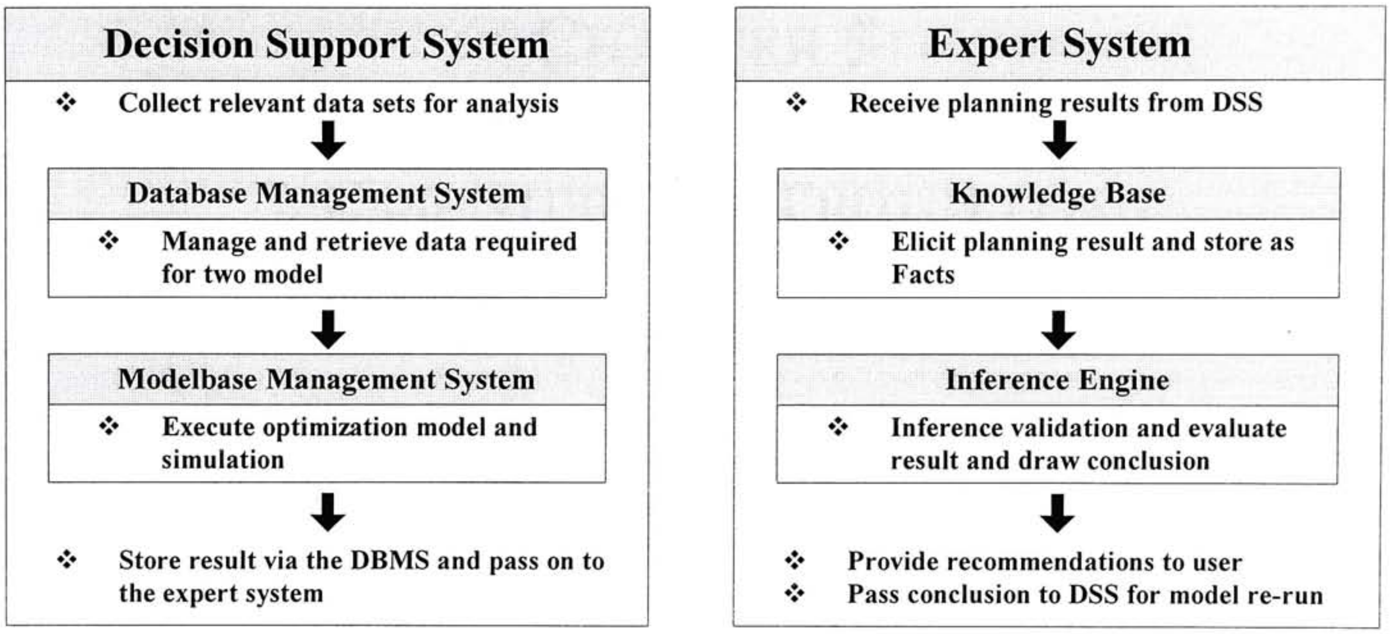


Diagram (a) Operations Flow in DSS

Diagram (b) Operations Flow in ES

Figure 4-6 System Operations Flow

Validation and Evaluation - With the planning configurations and simulation statistics resulted from the DSS subsystem, the ES subsystem mimics an expert to validate the operational feasibility and to evaluate the performance of the configurations. The generated results are passed to the knowledge base and reserved as part of the facts. Inference engine utilizes these facts together with the rules, which holds the knowledge of how the planning design should be interpreted, validated and evaluated, starts a consultation to conclude the best action for feedback and iterations.

Conclude and feedback - The concluded action drawn from the consultation is finally passed to the interface of the ES subsystem. In which the recommended action is displayed to the decision-maker via the dialog and transferred to the MBMS for specific model and data manipulation. Throughout the operations planning, evaluated results are feedbacked with specific modification. The whole process steps are repeated until the decision-maker up to a validated and satisfying result.

CHAPTER 5

5. CASE STUDY AND PROTOTYPING

In this chapter, we aim at investigating the feasibility of the Expert Decision Support System (EDSS) for the two stage operations planning. A prototype system is built based on a network design project for a major air-express courier. This exercise serves two purposes: (1) to proof the conceptual design of the EDSS can be operationalized and use for operations planning, (2) to demonstrate the two major functions of the EDSS, namely data and model management and intelligent guidance for the integrated use of the two models.

5.1 Case Background

The selected case study, including the case background, planning methodology and the feedback algorithm, is quoted from the research project by [Leung and Cheung, 1999] with DHL(HK).

With the impacts of the relocation of the Hong Kong international airport to Chek Lap Kok, major infrastructure developments are taking place rapidly in supporting the changes in logistic services and shifting in customer demands. In response to such changes, DHL, one of the world-wide leading air-express courier services companies, seeks to redesign its distribution network in HK to capitalize opportunities, minimize cost, and improve customer service.

5.1.1 The Service Network

DHL's planned service network, as schematically shown in Figure 5-1, consists of demand zones, satellite depots, service centers, and the airport.

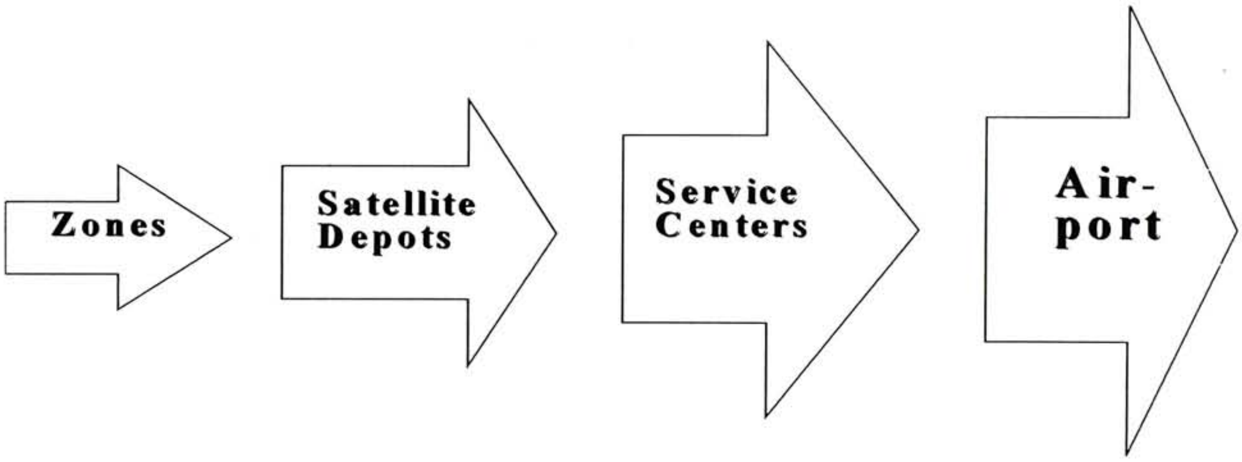


Figure 5-1 Components of the Distribution Network

Demand zones are predetermined service areas organized according to the level of customer demand as well as geographical characteristics. There are more zones within busy and commercial areas where order pattern is concentrated, while most outlying couriers are assigned to a specific satellite depot which covers pickups in several zones. At the depots, packages will be consolidated and consolidated load will then be delivered to the corresponding service center responsible for the depot. A service center, which also functions as a depot is responsible for several depots. At the service center, all major processing such as labeling, X-ray screening, re-weighing, sorting, documentation and formality following-up, etc. are done. Shipments will further consolidated into air containers or bags and be transported to the airport for transfer onto the corresponding aircraft.

DHL(HK) must manage effectively the processes of pickup, consolidation, processing, further consolidation, and delivery to the airport. The service network is at the heart of this process. The critical decisions in the design of the service network are installation decisions of depots and service centers:

- Locations of the depots and their coverage of demand zones.
- Locations of service centers and their coverage of depots.
- Capacities of these facilities.
- Installation schedule of these facilities.

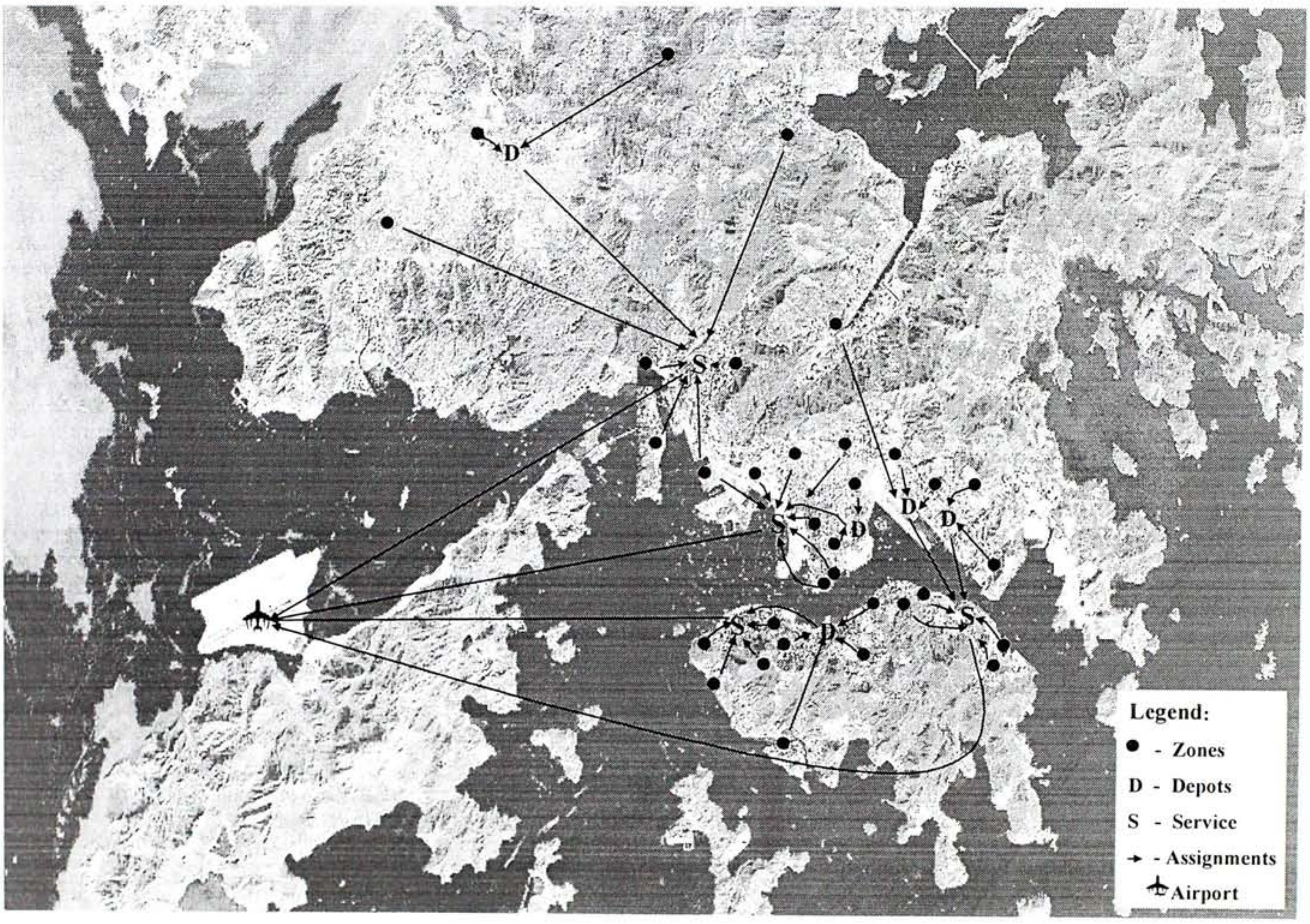


Figure 5-2 Map of Hong Kong with Principal Facility Locations and Assignments

5.1.2 Objectives of the Project

The objective of this project is to develop a service network that would be most economically and operationally desirable for DHL over a ten-year period. A courier's service network is closely tied to its service operations. The network must be designed with long-term considerations as well as short-term operational goals. It should include strategic and timely installations of depots and service centers, and should also be developed with judicious examination of the service performance. The entire study requires designing the overall framework, formulating models, collecting and preparing data, interpreting results, setting operating rules and policies, and making recommendations to the top management. The principal strategic recommendations are:

- Installation decision of depots and service centers
- Strategic cut-off time that balance capturing more business and missing service promise

5.1.3 Network Design Methodology

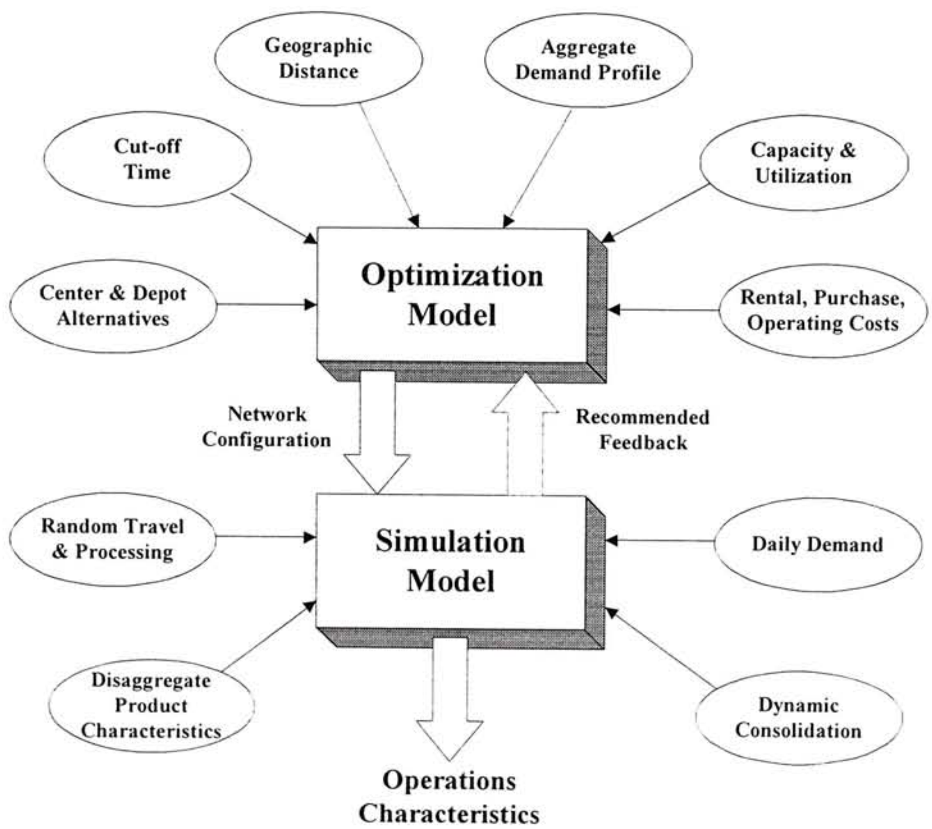


Figure 5-3 Two-Stage Network Planning Approach

The two-stage approach (Figure 5-3), with a mixed integer programming (MIP) model together with a SIMAN based simulation model is used to raise the suggested network. The two models are independent but related. The MIP model tries to seek for a ten-year horizon network planning with the minimal cost settings, while the simulation model is used to evaluated the performance of the developed network given by the MIP model at the operational level.

Macro Planning Model

The planning model seeks to determine a 10-year distribution network. The objective of this optimization model is to design a network with minimal cost settings subject to customer demands, facility capacities and response time. Major decisions are the locations of two types of facilities (*satellite depots and service centers*), their corresponding capacities and the year of installation together with the assignment of the shipment. The planning model was implemented with a PC-based mixed integer programming software MPSIII[1], with 45,900 continuous variables, 1,050 zero-one variables, and 820 constraints.

Decision Variables

X_{ijrt} = Shipments in zone i picked up by depot j processed in service center r in period t.

Y_{jkt} = 1, if pick-up capacity type k is installed in depot j in period t.

= 0, otherwise

Z_{rmt} = 1, if service capacity type m is installed in service center r in period t.

= 0, otherwise

Parameters

---

[1] MPSIII is a registered trademark of Ketron.

VC_{ijt} = Unit cost of picking up in zone i by depot j in period t

Vs_{jrt} = Unit cost of transporting from depot j to service center r, plus unit cost of processing in service center r in period t

FC_{jkt} = Fixed cost of installing kth capacity in depot j in period t

FS_{rmt} = Fixed cost of installing mth service capacity in depot r in period t

d_{it} = Demand (Shipments) in zone i in period t

C_{kt} = Pick-up capacity of kth type in period t

S_{mt} = Service capacity of mth type in period t

μ = Utilization limit of facilities (from planning policy)

td_{ijt} = Pick-up time in period t (travel from depot j to i and return)

ts_{jrt} = Travel time from depot j to service center r in period t

tp_{rt} = Processing time in service center r in period t

ta_{rat} = Travel time from service center r to airport in period t

$T_{max,t}$ = Time window, maximum allowable flow time in period t

$TZ_{i,max}$ = Maximum response time allowed in zone i

Objective Function

The objective function minimizes the sum of present-value costs of transportation and facility installation. The variable transportation cost is dependent on the assignment of shipments from zones to depots and from depots to service centers. The installation costs are dependent on the installation decision, the choice of capacity level, as well as the schedule of the installations, for both depots and service centers.

(1) $$\textbf{Min: } \Sigma_i \Sigma_j \Sigma_r \Sigma_t \{ VC_{ijt} + VS_{jrt} \} X_{ijrt} + \Sigma_j \Sigma_k \Sigma_t FC_{jkt} Y_{jkt} + \Sigma_r \Sigma_m \Sigma_t FS_{rmt} Z_{rmt} \}$$

Constraints

Assignment of shipment must meet demand. A necessary condition in the coverage of customer requests is that all demand must be met. That is, shipments originating in zone *i* in period *t*, for every zone and every period, must be covered.

$$(2) \qquad \Sigma_j \Sigma_r X_{ijrt} = d_{it} \qquad \forall i, t$$

Capacity of depot cannot be exceeded. The assignment of shipments (from different demand zones) to a depot must be accompanied by the decision to install the depot along with the corresponding capacity decision. The shipment assigned must not exceed the accumulated capacity of a depot operating at maximum utilization level. It must also be ensured that no more than one capacity type can be installed at any given period.

$$(3) \qquad \Sigma_i \Sigma_r X_{ijrt} \leq \mu \{ \Sigma_k C_k Y_{jk1} + ... + \Sigma_k C_k Y_{jkt} \} \forall j, t$$

$$(4) \qquad \Sigma_k Y_{jkt} \leq 1 \qquad \forall j, t$$

Capacity of service center must not be exceeded. Similarly, the amount of shipments (from different depots) allocated to a service center must be accompanied by simultaneous decisions of installation and capacity choice with an utilization limit. Only one capacity-type installation is permitted if the installation is to be implemented at all.

$$(5) \qquad \Sigma_i \Sigma_j X_{ijrt} \leq \mu \{ \Sigma_m S_m Z_{rm1} + ... + \Sigma_m S_m Z_{rmt} \} \qquad \forall r, t$$

$$(6) \qquad \Sigma_m Z_{rmt} \leq 1 \qquad \forall r, t$$

Maximum flow-time. A crucial requirement is that the flow time of a delivery (elapsed time from pickup to airport arrival) cannot exceed the *time window*, the duration between cut-off time and latest arrival time to the airport. That is all "*ijrt*" links must satisfy the condition $\{td_{ijt} + ts_{jrt} + tp_{rt} + ta_{rat}\} \leq T_{max,t}$. Here, we prescreen all the links that violate the time window by setting the corresponding decision variable X_{ijrt} to zero.

Maximum response time. To ensure speedy pickup of customer packages, the response time for a pickup must not exceed the maximum response time, which is a predetermined limit. Similarly, we prescreen all *"ijrt"* links which violate the condition $td_{ijt} \leq TZ_{i,max}$ by setting the corresponding decision variable X_{ijrt} to zero.

Operation Simulation Model

The simulation model tries to validate the performance of the developed network at the operation level. As macro planning model develops the network using parameters in aggregate and average manner, it does not take into account of the daily variations and random behavior. A SIMAN-based simulation software - ARENA[2] is used to model the air express courier daily operation. The simulation model here simulates the daily operations of the developed model and investigates the performance of the network in different measures, e.g. the operating cost of the system, the utilization of each facilities, the service coverage and reliability of the system.

The Simulation Environment

The simulation experiment considers the dynamics of courier pickups, delivery to depot and service centers, and delivery to the airport. A schematic depiction of the simulation environment is shown in Figure 5-4. The locations of the depots and service centers, as well as shipment assignments of zone-depots-service centers, are in accordance with the results of the planning model. Three types of vehicles - van, truck, and lorry - are used to transport shipments from zone to depot, depot to service center, and from service

---

[2] ARENA is a registered trademark of System Modeling Corporation.

center to the airport respectively. Consolidation of shipments from vans to trucks, and from trucks to lorries are simulated. The entire production process will be applied to two major product types, *Document* and *Package*, which collectively represent almost 90% of the shipments.

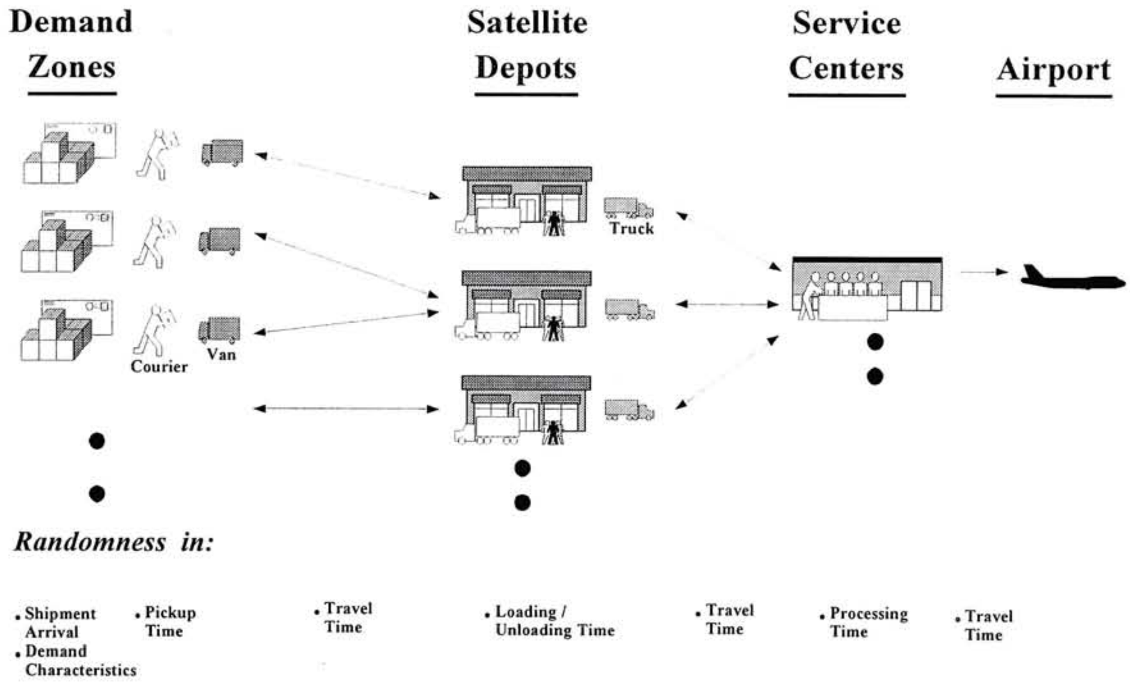


Figure 5-4 The Simulation Environment

The workforce includes couriers and data-processing workers. A typical working day starts at 8:00 am when couriers leave depots by vans to pick up shipments in zones. The lunch break is from 12:30 to 2:00 pm and the cutoff time is 5:15 pm in each zone. Each week has five and a half workdays with a Saturday finish at 12:30 pm and no activity on Sunday. The quantity of vans, trucks, lorries, couriers, and data-processing workers are based on the corresponding cost estimates used in the macro model. Probabilistic behaviors exhibited in three categories of events - shipment arrivals and characteristics, travel time, and processing time - are incorporated.

The Simulation Model

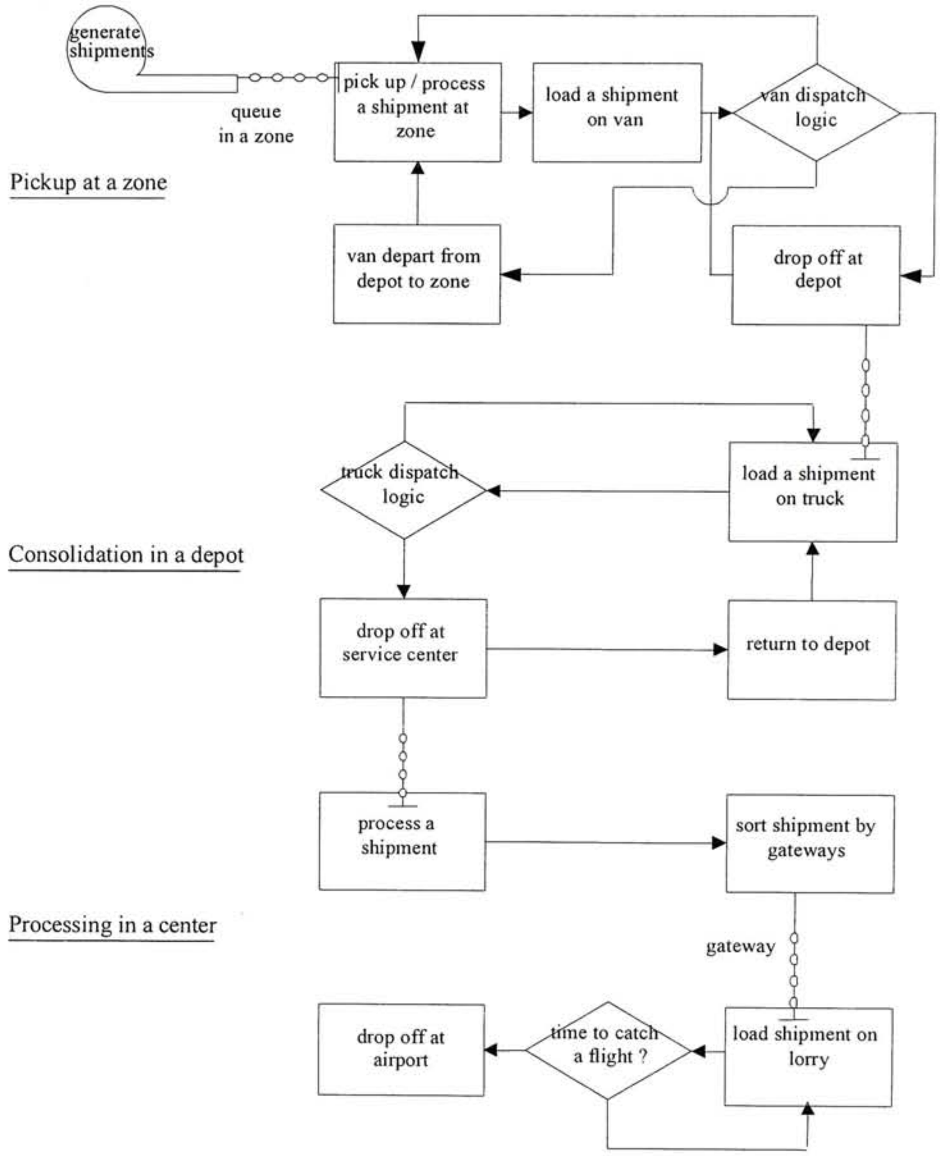


Figure 5-5 The Simulation Model

The logic of the simulation model is shown in Figure 5-5. The distribution network along with the assignments recommended by the planning model are first initiated. Two types of entities representing the two product types are generated according to their arrival patterns and characteristics (i.e., weight and destination) for each demand zone. The shipment arrival rate depends on the zone, the time of the day, and the day of the week. Here, shipments are generated for each zone according to a Poisson process with a mean rate (estimated from historical data) that changes every hour. Shipment type and destination are generated from predetermined distributions which are also estimated from historical data. Weight distributions for *Document* and *Packages* are determined via the distribution fitting of historical data. The former is approximated by a bounded normal distribution and the latter by an exponential distribution.

Both zones and service centers are modeled as one-line multiple-server queueing resources. The capacity of a resource represents either the number of couriers in a zone or the number of processing workers in a service center. At the end of each day, entities which remain in the queue of each zone are lost-sale shipments. Entities which remain in the queue of each service center are undelivered shipments and are to be delivered on the following day. Since there is no processing needed in the depots except consolidating shipments, a depot is modeled as a simple storage with a queue, where a time delay is incorporated for the unloading and loading of vehicles.

Three types of transporters (representing van, truck, and lorry) are created to handle entity movements from zone to depot, depot to service center, and from service center to the airport respectively. Travel time for all routes were measured under differing traffic conditions. An average travel time T' and a standard deviation s were obtained for each route. Assuming that the travel time ranges from a low value of $T' - 2s$ to some very large value (due to traffic congestion), we use the following Gamma distribution for generating travel time: $T = (T' - 2s) + Gamma(\alpha, \beta)$, with $\alpha = 4$ and $\beta = s$. Note that the mean and standard deviation of T are respectively T' and s. The processing times for both couriers at zones and workers in service centers are described by bounded normal distribution. Each simulation run covers one week with no activities on Saturday afternoon and Sunday.

5.2 Iterative Network Planning

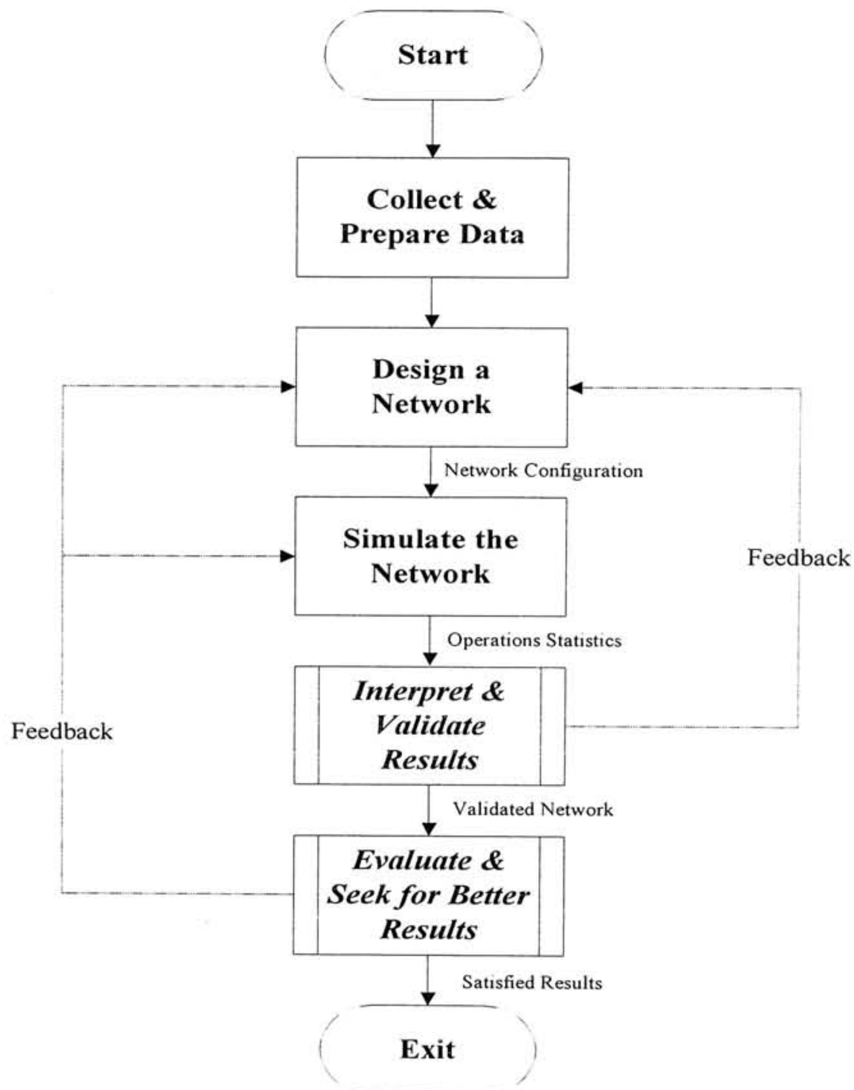


Figure 5-6 Decision flow in network design

Throughout the network design process, the decision-maker has to manipulate the two models iteratively to gradually reach a planning solution. Operation characteristics collected from the simulation are feedbacked to the optimization model during validation and evaluation. Besides, the decision-maker needs to re-run the models periodically with revised data in future. Figure 5-6 illustrates the flow of the network design. The decision-maker first develops a ten-year horizon network using a mixed integer programming model. Then, the developed network is simulated for its daily operation with a SIMAN based simulation software. According to the simulated results, the decision-maker validates the feasibility of the network and cross-checks the input used in the planning stage. Finally, the decision-maker evaluates the performance of network and tries to fine tune the network until the decision-maker is satisfied with the concerned performance measures.

5.2.1 Multi-period Network Planning Feedback

The final design of a multi-period network planning counts on the viability and the performance of the configurations in each period. To validate and evaluate the network planning, simulation should be run for each year in the planning horizon. However, as data and parameters estimated for the models become less accurate in the distance planning period, validation and evaluation is only sensible to be done within couples of future planning period. More importantly, as facilities planned in the first year are accumulated throughout the whole planning periods, network configurations for the following periods rely heavily on the configurations in the first year. As a result, the very first step of validating and evaluating a multi-period network planning goes to the simulation of the first planning year. In practice, the decision-maker starts with the simulation of the first year and then to the second and third, depending on the network configurations resulted in the macro model. In some scenarios, the decision-maker may be satisfied and stop with the simulation in the first year, as the developed configurations in the following periods are similar with no facility expansion recommended. Simulations in these planning periods are likely to give comparable result as first year, which may be skipped.

As the multi-period network design is sensitive to the modification of the model inputs, revision of any data/parameters during the validation and evaluation for each planning period may result in a totally different network design. Such consequences may cause decision-maker to repeat the whole feedback process in each data/parameter modification.

As a result, the interactions between different measures in both validation and evaluation for each period needs to be addressed throughout the feedback process.

5.2.2 Feedback in Validation and Evaluation

During the network planning, a ten-year horizon network is developed based on a series of data sets and parameters, including the estimated unit transportation cost of each year, facility utilization factor to control the facility utilization, and a time window to control the delivery time. Developed network is resulted with configurations for each year with the respective facility locations, capacity type and the shipment assignments. To test if the developed result is feasible and with satisfying performance, simulation is done to examine the network viability and its efficacy based on four issues: (1) unit transportation cost verification, (2) facility utilization validation, (3) service coverage evaluation, and (4) service reliability evaluation.

Network Validation

Network validation aims at testing the viability of the developed network. Two of the measures are critical and needs to be verified and validated if the network is operational feasible. (1) To check if the cost estimated in the macro model is close as simulated. (2) To validate if any of the facilities is over-utilized during daily operations. If any of the validation fails, specific parameters are revised to rectify the problem.

Unit transportation cost verification. The verification of the unit transportation cost is important because the network configurations resulted in the macro model are dependent to both variable and fixed cost of the facilities. As facility setup costs are not concerned during simulation, the verification of operating and transportation cost plays an important role during cost validation. In the simulation, operating cost is computed according to the daily shipment pickup activities, workforce incurred, and transporter trips utilized. Occasionally such simulated cost may not be the same as the cost estimated in the macro models, which are implicitly based on certain simulation environment. As the choice of facility site locations, capacity and installation year are sensitive to the operating cost, an accurate estimation of the costs is essential to give an appropriate network configuration. As a result, the verification of the unit transportation cost becomes the first step of the network validation. It should be noted that the interaction between the resulted network configuration and the simulated cost is so complex that a small modification of the unit transportation cost in the MIP model may give a very different configuration. Hence, the control of the numbers of iteration in the cost validation needs to be addressed.

The algorithm for the unit transportation cost verification is illustrated as Figure 5-7. The decision-maker first computes if the overall cost deviation is significant and determines the member of the routes that are relevant for validation. Then each of the simulated cost in all relevant routes is compared with the cost estimated in the macro model. In case of any substantial deviations, respective cost is updated according the number of iterations gone through. Hence, the updated cost is prepared for model re-run in the next iteration.

Step 0:	Initialize parameters: tolerance limit = $\eta\%$; simulation iterations = n.
Step 1:	Select major routes. A route r_{ij} is major if its assignment $\geq$ average volume.
Step 2:	Validate estimated costs ec with simulated costs sc for major routes. If $\lvert sc_{ij} - ec_{ij}\rvert / ec_{ij} \geq \eta\%$, validation for r_{ij} fails. Else, go to step 4.
Step 3:	Modify ec_{ij} for r_{ij} in macro data set. If number of iterations $\leq n$, set $ec_{ij} = sc_{ij}$; else, set $ec_{ij} = (sc_{ij} + ec_{ij})/2$.
Step 4:	Conclude cost validation. If all major routes are validated, validation is completed; else re-run macro model with updated cost.

Figure 5-7 Algorithm of Unit Transportation Cost Validation

Facility utilization validation. Another concern during the network validation is to validate if the planned facilities are feasible in the actual operating environment. As the utilization of the facilities are important facets of the network, it is important that the facilities in the network are neither over- nor under-utilized. The simulation experiments provide clearer picture of the utilization for the planned facilities. More importantly, the macro model ensures capacity feasibility in an aggregate fashion, but the daily utilization is not concerned. As a result, another exercise for the facility utilization validation is to give information for the decision-maker to examine the facility utilization in a micro level. These information are important for planning a smooth network operations as these statistics enable the decision-maker to identify certain peak utilization in specific facilities and week days, or even during certain period. As a result, the decision-maker can determine the respective rectification to the planning.

The revision of the utilization problem is done by either revising the utilization limit μ in the MIP model or by minor route re-assignment to alleviate the over-utilized facilities. The utilization limit is a parameter in the MIP model, which control the utilization of the facilities. In a classical scenario, the utilization of the facilities is expected to be operated under 85% of its total capacity. The utilization limit here controls the maximum capacity of the facilities within this planned limitation. However, as simulation runs may give very different results, the utilization limit may need to be revised to provide feedback to the changes. On the other hand, over-utilization may be alleviated by minor route re-assignment depending on the overall network situation. Based on the simulation statistics, the decision-maker tries to determine the appropriate modification.

The algorithm for the facility utilization validation is presented in Figure 5-8. The decision-maker verifies the daily utilization of two types of facilities, (1) satellite depots and (2) service centers. The simulated average utilization for each facility is first computed and compared with the estimated average utilization. The two figures should match, which verify the simulation model. To ensure smooth daily operations, daily utilization is examined. If the simulated result indicates an over-utilization during daily operation, the decision-maker may try to either make manual re-assignment to some shipments or decreased the respective facility utilization factor in the macro model according to the overall network situation.

Step 0:	Initialize parameters. Acceptable utilization level = $\alpha\%$; acceptable percentage of over-utilized facilities = β.
Step 1:	Check daily utilization for each j. If daily utilization $su_{dj} \geq \alpha\%$, j is over-utilized on day d. Repeat for all d. If no over-utilization takes place, go to step 4.
Step 2:	Check overall facility capacity tightness. If % of over-utilized facilities $\geq \beta$, overall facility capacity is tight; go to step 4.
Step 3:	Check re-assignment for over-utilized facilities. Re-assignment is infeasible if (1) no spare capacity in adjacent facilities, and (2) no alternate route meets time constraint. Else, reassign and proceed.
Step 4:	Conclude utilization validation. Rule 1: if no daily over-utilization, utilization validation is completed. Rule 2: if the overall utilization is tight, re-run macro model with reduced μ. Rule 3: if route re-assignment is feasible, re-run simulation with routes reassigned; else re-run macro model with reduced μ.

Figure 5-8 Algorithm of Facility Utilization Validation

Performance Evaluation

During performance evaluation, the decision-maker intends to evaluate the network performance and to test for any potential rooms for improvement in the current configuration. Two of the major performance measures which are critical to the success of the company operations are examined. (1) Service coverage – which reflects the coverage of the customer request and (2) service reliability – which represents the percentage of the delivery promise made. In the macro model, both coverage and reliability are implicitly required to be 100%. This is due to the requirements that all demand must be met and that the network allows sufficient time for all shipments to be delivered to the airport. The former requirement assumes deterministic behavior in demand and the latter in travel and processing times. However, in the simulation, the model captures detail dynamics of the operational characteristics on a daily basis, thus producing a more realistic picture of the network performance. As a result, the performance evaluation is done for evaluating the two critical measures in the daily operations. In case of unsatisfactory performance, the decision-maker needs to either modify the operations parameters or simply redesign a more efficient network with revised parameter.

Service coverage evaluation. When taking into account of the daily shipment arrival pattern, simulation experiment usually results a lower coverage than the estimated in the macro model. During daily operations, percentage of response to the customer demand depends on the operational cut-off time defined. Obviously, the later the cut-off, the higher the coverage attained. As a result, the decision-maker can attain the desired coverage by adjusting the operational cut-off.

Step 0:	Set required service coverage level = δ%;
Step 1:	Check simulated coverage. If simulated coverage is ≥ δ%, coverage is satisfied; go to step 3.
Step 2:	Estimate and modify the operational cut-off time to a later time, t_c.
Step 3:	Conclude coverage evaluation. If service coverage is satisfied, coverage evaluation is completed. Else re-run simulation with t_c.

Figure 5-9 Algorithm of Coverage Evaluation

Figure 5-9 illustrates the algorithm for the coverage evaluation and how improvements are made during performance evaluation. Based on the simulated coverage, the decision-maker identifies if the existing network give a desirable coverage. In case of revision required, the decision-maker studies on the simulated shipment arrival pattern and hence estimated the respective cut-off. The decision-maker then revises the operational cut-off parameter and re-run simulation model.

Service reliability evaluation. Similar as the measures of service coverage, the simulated reliability may be lower than the estimation due to the dynamics of the operational characteristics, including the traffic congestion, randomness of the travel time and processing time, etc. On the other hand, the trade-off in delaying the operational cut-off to accomplish a higher coverage will also decrease the service reliability. In both cases, modification is needed to restore the reliability to an acceptable level. One of the ways to improve the reliability is to increase the workforce level. Obviously, increased the workforce can share the workload and hence reduce the shipment pickup time. Another way to improve the reliability is to redesign a more time efficient network. As modification can be done by applying a more stringent time-window and re-run the MIP model.

Step 0:	Set required service reliability level = σ%.
Step 1:	Check overall simulated reliability. If overall reliability $R < \sigma$%, reliability check fails; go to step 4.
Step 2:	Check daily reliability. If daily reliability $R_d < \sigma$%, there is minor reliability problem on day d. Else go to step 5.
Step 3:	Estimate and modify workforce to a higher level for zone i on day d
Step 4:	Determine new time window. Set time window $T = T - t_s$ (t_s is the smallest slack travel time among all zones).
Step 5:	Conclude reliability evaluation. Rule 1: if there is a minor reliability problem on day d, re-run simulation with increased workforce. Rule 2: if overall reliability fails, re-run macro model with the new T. Else reliability evaluation is completed.

Figure 5-10 Algorithm for Reliability Evaluation

Figure 5-10 illustrates the algorithm of the reliability evaluation. The essence of the reliability evaluation goes to the determination of the approach of improving the reliability. Decisions are made by investigating both the overall and specific weekdays and zones tha specific network reliability. When the overall reliability is below the required level, the macro model is re-run with a narrower time window to obtain a more efficient network for improvement. On the other hand, specific reliability problem may be examined in particular weekdays and zones. In such cases, couriers may be added to those weekdays or zones with reliability below the required level to improve the performance.

5.3 The System Prototype

Distribution network design requires the decision-maker to repetitively use both models and reapply the knowledge of how to conduct each iterations. Throughout the design process, we summarize that a system can facilitate the decision-maker in two levels. The first level focuses on manipulating data and models. The steps of data gathering, storage, retrieval, and routine analysis in using both models will be facilitated. The second level aims at providing intelligence guidance during network design. The interpretation of the result, validation and evaluation are assisted with the help of the system. According to the network planning process described, we summarize the incurred activities into two categories: (1) data and model management, and (2) intelligence feedback guidance. The functions of these two categories are implemented and responsible by two subsystems, Decision Support System (DSS) and Expert System (ES) respectively.

5.3.1 Data Management and Model Manipulation

Throughout the planning design, the decision-maker first manipulates with data and models to generate preliminary network configurations and simulation results. DSS subsystem which focuses on providing data and models manipulation functions assist the decision-maker in these process.

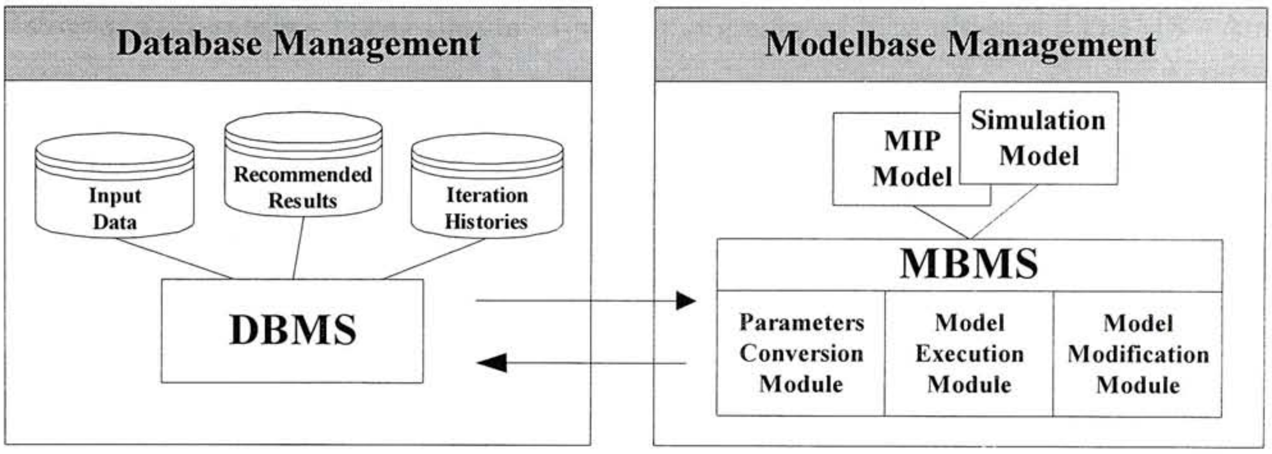


Figure 5-11 Data and Model Manipulation

Data management functions during network planning process includes three major tasks: (1) managing all data inputs require for both MIP and simulation model, (2) storeing up results generated by two models, including network configurations and simulation statistics, and (3) recording the iteration histories throughout the planning process. On the other hand, model manipulation focuses on three functions: (1) initiating the execution of MIP and simulation model upon request, (2) getting and putting data/results by communicating with the database management system (DMBS) with file format conversions, and (3) performing simple model modification by specific function calls.

Data Management

In the network planning, the decision-maker collects and verifies all data required for analysis and stores in forms of Excel spreadsheet files. Three types of data are categorized and stored in the database system. (1) Data inputs, which refers to all the data sets required for the MIP and simulation models in the networking planning. (2) Intermediate results which refers to developed network configurations, simulation logs and statistics generated from two models. (3) Iteration histories which refers to the parameters used and interpreted results for each iteration. These files are linked and managed by a PC-based DBMS - Access, which enable both direct manipulation of data and communication with the MBMS.

Input Data. Data sets for both models are stored in forms of spreadsheet files and managed by the DBMS. Aggregate data set for the macro model includes ten files stored in comma separate values (CSV) format. Data files include:

(1) *Data1.csv - aggregate annual demand of each zone*
(2) *Data2-3.csv - facility capacity type specification*
(3) *Data4-5.csv - fixed cost for different facility specification*
(4) *Data6.csv - unit transportation cost from zones to depots*
(5) *Data7.csv - unit transportation cost from depots to centers*
(6) *Data8.csv - average transportation time from zones to depots*
(7) *Data9.csv - average transportation time from depots to centers*
(8) *Data10-12.csv - average processing time in different centers, average transportation time from centers to airport, and planning time window*
(9) *Data 13.csv - unit transportation cost from centers to airport*
(10) *Spec.csv – list of possible candidates of depot and center*

On the other hand, operations details for the simulation model are stored in the form of tab delimited format in seven sets of files. Data files include:

(1) *Flights.prn - Schedule of the flight in each weekday for major destination*
(2) *Gateway.prn - Distribution of shipment demand for different destination*
(3) *Satdata.prn - Satellite depots operation details, including route assignment and travel time to service centers*
(4) *Srvdata.prn - Service center operation details, including route assignment and travel time to airport gateways*
(5) *Zonesdata.prn - Demand zones operation details, including shipment*

assignment, shipment characteristics, number of couriers, cutoff time policies, travel time depots

(6) *ZoneXXdata.prn - Hourly demand of each zone for Sunday to Saturday*

(7) *Mis.prn - All miscellaneous operations details, including shipment process time, transporter capacity, service center process time, etc.*

Intermediate results. The results developed in the network planning project includes the suggested network configuration from the MIP model and simulation statistics from the simulation model. These files are stored in the form of dBase files and managed by the DBMS. These results are used for validation and evaluation in the feedback process and for reference during the planning project. Outputs from the macro model includes three sets of output files:

(1) *Network configurations – Size and year of installation of each facilities throughout the planning horizon together with the shipment assignment.*

(2) *Estimated cost settings – Total cost of the network configuration and cost breakdown of each facilities for each planning period with transportation and fixed cost.*

(3) *Estimated network performance – Estimated facility utilization in each planning period together with the slack time of each of the routing assignment.*

For the simulation model, output of log files describes the operations characteristics of the developed network. Statistics includes:

(1) *Courier.log – Statistics of shipment pickup activities for each courier*

(2) *Depot.log – Statistics of shipment process, consolidation and arrival pattern for each satellite depot*

(3) *Center.log – Statistics of shipment process, consolidation and arrival pattern for each satellite center*

(4) *Undeliver.log – Statistics of undelivered shipments including shipment not pick from the customer and shipment not delivered within the same day flight*

(5) *Transporter.log – Statistics of each transporter recording the miles and trips*

travel

Iteration histories. Throughout the network planning process, such planning and simulation activities are repeated with various feedback. In order to keep track on the iterations for decision-maker during network planning, the iterative histories are recorded and kept in the database via the DBMS. Histories of the network planning iterations are stored in one dBase file which includes the information of:

(1) number of iterations,
(2) parameters used in each run,
(3) modification made to data and models

Model Manipulation

During the network planning, the modelbase management system receives commands initiated from decision-maker or the expert subsystem and hence performs various model manipulations. Two major functions, model execution and model manipulation, are supported by three categories of program modules in the MBMS. Each module is composed of a set of C programs and batch files responsible for specific functions and manipulations.

Parameter conversion module. During the network planning, the decision-maker initiates a command in DSS subsystem, Then, the MBMS communicates with DBMS to request relevant data sets for MIP and simulation models, and hence converts into specific input format for execution. This program module is responsible for cooperating MBMS with the DBMS to get the specified data sets, parameters and hence generate the input files in specific formats for MIP and simulation model execution. Two C programs named '*convert_mip.c*' and '*covert_sim.c*' are responsible for the input files generation for MPSIII and ARENA respectively.

- *convert_mip (scenario, time window, utilization limit)*

  This program generates input files in specific format for the MIP model execution. According to the scenario specified, it retrieves the respective sets of macro model data sets via the DBMS. Together with the parameters of planning time window t_{max}, and utilization limit μ, this program generates the objective functions and constraints of the MIP model into two specific format files, a mps file (the model file in MPSIII format) and a rtb file (the control file for MPSIII execution).

- *convert_sim (scenario, cutoff, workforce level)*

  This program generates input files for simulation model execution. Based on the scenario specified, it retrieves the respective developed network configurations, including the locations and size of the facilities with the routing assignments, and also the operation details via the DBMS. Together with the parameters of operation cutoff time *co* and workforce level *wf*, a sets of input files is generated for simulation execution.

Model Execution Module. This program module is responsible for utilizing the input files generated from the parameter conversion module and calling upon MPSIII and ARENA programs for respective model execution. Two batch files of '*mip.bat*' and '*sim.bat*' together with the optional execution parameters initiate the respective programs and performs the model execution.

- *mip ([nodes])*

  This batch file calls upon the execution of MPSIII. By using the input files (mps and rtb files) generated by '*convert_mip*', the batch file initiates the mixed integer optimizer, MPSIII for execution. With an optional MPSIII execution parameters, the number of nodes *n* can be specified according to the decision-maker preference. Resulted files are then extracted and stored in the DBMS.

- *sim ([replication])*

  This batch file calls upon the execution of ARENA simulation. By using the sets of input files generated by *'convert_sim'*, the batch file initiate the SIMAN-based simulation software, ARENA for execution. With an optional simulation execution parameters, the number of replication *r* can be specified according to the experiment run design. Simulation outputs are stored via the DBMS for later reference.

Model Modification Module. During the feedback and re-evaluation of the developed network configurations, the decision-maker may require to perform modification in data sets and/or parameters, add or release some of the constraints in the MIP model or make policy changes in the simulation model. This module is responsible how providing various model modifications. A series of C programs are written for different predefined modifications and stored in this module.

- *cost_update (scenario, period, zone, depot, cost)*

  This program enables the modification of the unit transportation cost in the MIP model. According to the cost validation result given by the expert system, this program parses the recommend action into a series of SQL statements and performs data updates via the DBMS. The generated SQL statements are based on a parameter file with the list of unit cost modification. Parameters includes the scenario specified, together with planning period of each unit cost *p*, origination zone *i*, destination depot *j*, and new value of cost, *c*.

- *time-window_update (scenario, time window)*

  This program modifies the network planning time window in the MIP model. According to the feedback recommendation from the expert system, this program modifies the time window parameter and calls upon the program *'mip_convert'* to generate a new sets of MPSIII input files. Modification is carried out based on parameters including the scenario specified, and the target time window t_{max} used in the simulation.

- *utilization-limit_update (scenario, utilization limit)*

  This program modifies the network planning utilization limit in the MIP model. According to the feedback recommendation from the expert system, this program modifies the utilization limit parameter and calls upon the program *'mip_convert'* to generate a new sets of MPSIII input files. Modification is carried out based on parameters including the scenario specified, and the new utilization limit μ used in the simulation.

- *facilities_selection (scenario, facility, location, type, period, action)*

  This program enables the decision-maker to force/ban specific facilities selection in the MIP model during network re-evaluation or with particular strategic reason. It bounds the facility selection in the MIP model by the modifying the constraints in the MPSIII input files. This modification is based on a file containing a list of all specified parameters. Parameters includes facility x (*depot or center*), location l, capacity type t, planning period p, and respective action *a (either ban or force)*.

- *workforce_change (scenario, zone, depot, change)*

  This program enables the modification of the courier workforce in specific zone(s) or weekday(s) in the simulation model. According to the performance evaluation result given by the expert system, this program parses the recommend modification into a series of SQL statements and performs updates in the simulation input via the DBMS. The generated SQL statements are based on a parameter file with the list of courier workforce modification. Parameters include the scenario specified, together with modification of each demand zone i, weekday(s) d, and workforce change Δ.

- *cutoff_change (scenario, cutoff)*

  This program enables the modification of the operational cutoff time in the simulation model. According to the suggested result from the expert system, this program modifies the cutoff policy in the simulation by modifying the cutoff parameters in the simulation input file. Modification is carried out based on parameters included, the scenario specified, and the target operational cutoff time *co* used in the simulation.

5.3.2 Intelligent Guidance for the Iterations

The second level of support during the network planning process is to provide the intelligence guidance throughout each of the iterations. According to the network planning flow, the system concentrates on three functions. (1) Interpreting and validating the results developed by two models. (2) Evaluating the performance of the network based on specific performance measures. (3) Determining the feedback in each of the iterative process, including the modification approach, parameters, data and model. To provide these functions, accumulated experience, knowledge and expertise are elicited and encoded in a knowledge base. These encoded knowledge is hence implemented in a backward chaining expert system shell, M.4[3] in forms of "If-Then" rules. The intention is to model the function of network validation and performance evaluation, together with the feedbacked recommendations.

Knowledge Engineering

The core of the EDSS goes to the part of providing intelligence feedback guidance during planning design. As a result, the success of this system heavily depends on how the accumulated expertise and knowledge is acquired and reapplied in the EDSS. Knowledge

[3] M.4 is a registered trademark of Cimflex Teknowledge Corporation.

engineering in our EDSS goes into four steps: (1) identifying task and domain, (2) preparing plan and knowledge acquisition, (3) representing and encoding knowledge, and (4) execute and test knowledge base.

Domain and Task Identification. To provide guidance during the feedback iterations, we first identify the procedures and tasks which the decision-maker need to achieve throughout the planning design. Figure 5-12 illustrates the overall feedback logic of the network planning process. According to the decision flow, four major issues are concerned during the feedback process: validation of (1) unit transportation cost, (2) facilities utilization, and evaluation of two performance measures, (3) service coverage, (4) service reliability.

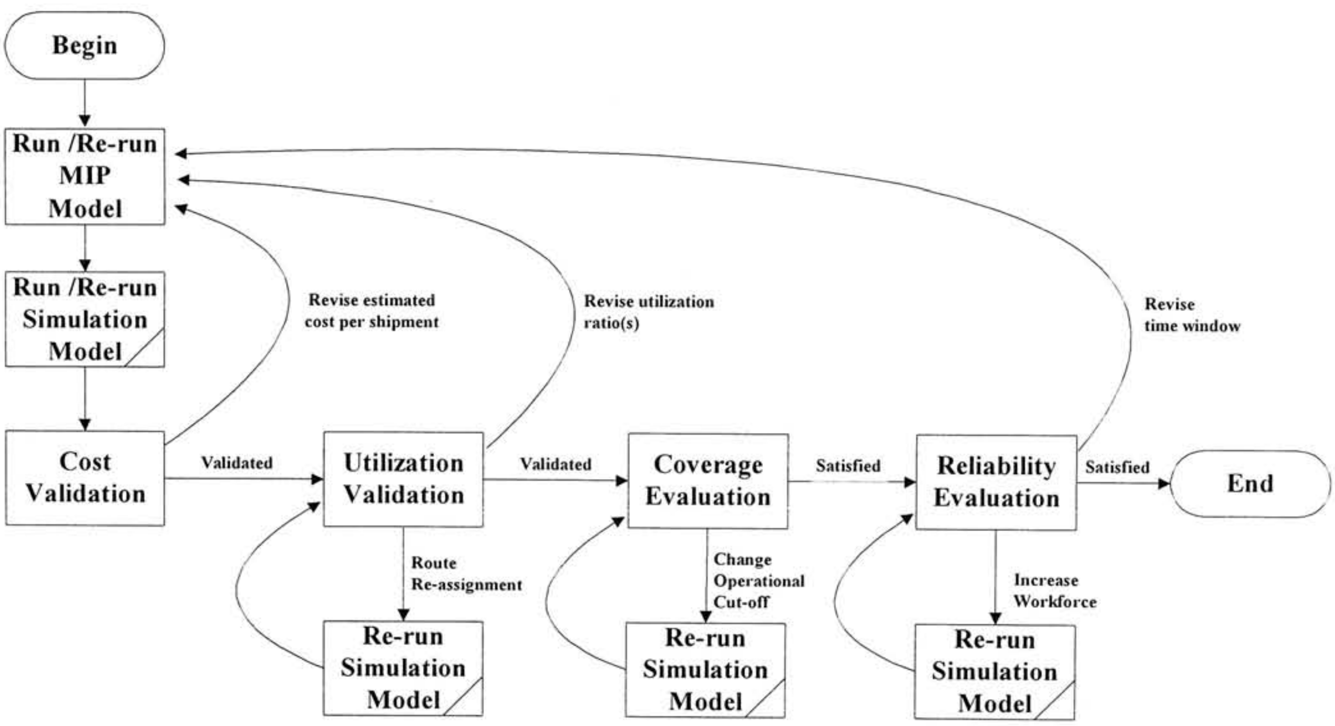


Figure 5-12 Network Planning Overall Feedback Logic

In general, each of the feedback tasks in the network planing can be divided into three steps. (1) Problem identification – nature of the problem is analyzed based on specific measures and simulation results. (2) Seeking for respective modification – modification is made by determining what and how specific parameters are revised according to problem specified as well as other related measures. (3) Concluding next action – with the revised parameters, proceed the next iteration by re-running either the simulation model or the MIP model.

Each of the task during the feedback process requires the decision-maker to interpret the results from both models and hence to make appropriate feedback actions according to a set of predefined rules and procedures. In short, to automate the feedback process, goals in our expert system are defined as:

(1) Unit transportation cost validation

- Verify estimated unit transportation cost with simulated cost
- Determine and update unit transportation cost

(2) Facility utilization validation

- Validate the facility utilization during daily operations
- Determine necessary rectification and parameter changes

(3) Service coverage evaluation

- Evaluate performance based on coverage
- Determine appropriate cut-off time to achieve target coverage

(4) Service reliability evaluation

- Evaluate performance based on reliability
- Determine necessary modification approach and parameter changes

Knowledge Acquisition. Obviously, the steps of how knowledge is acquired for these goals play an important role in the knowledge engineering. The task involves extracting the domain-specific expertise and problem-solving wisdom to the goals specified. In this application, it seeks to capture the knowledge, heuristics, and rules employed in the decision-maker/modeler during the validation and evaluation process. Major sources of the knowledge is gathered from three areas, (1) field expert opinion, (2) experience gain in pass scenarios, and (3) sensitivity tests and analysis.

Acquired knowledge about the algorithm of each validation and evaluation task is documented in the form of flow logic diagrams. These diagrams help to formalize the process of performing each validation and evaluation and hence raise information to the coding of rules in the expert system. In short, we summarize the validation and evaluation tasks into three logic flow diagrams.

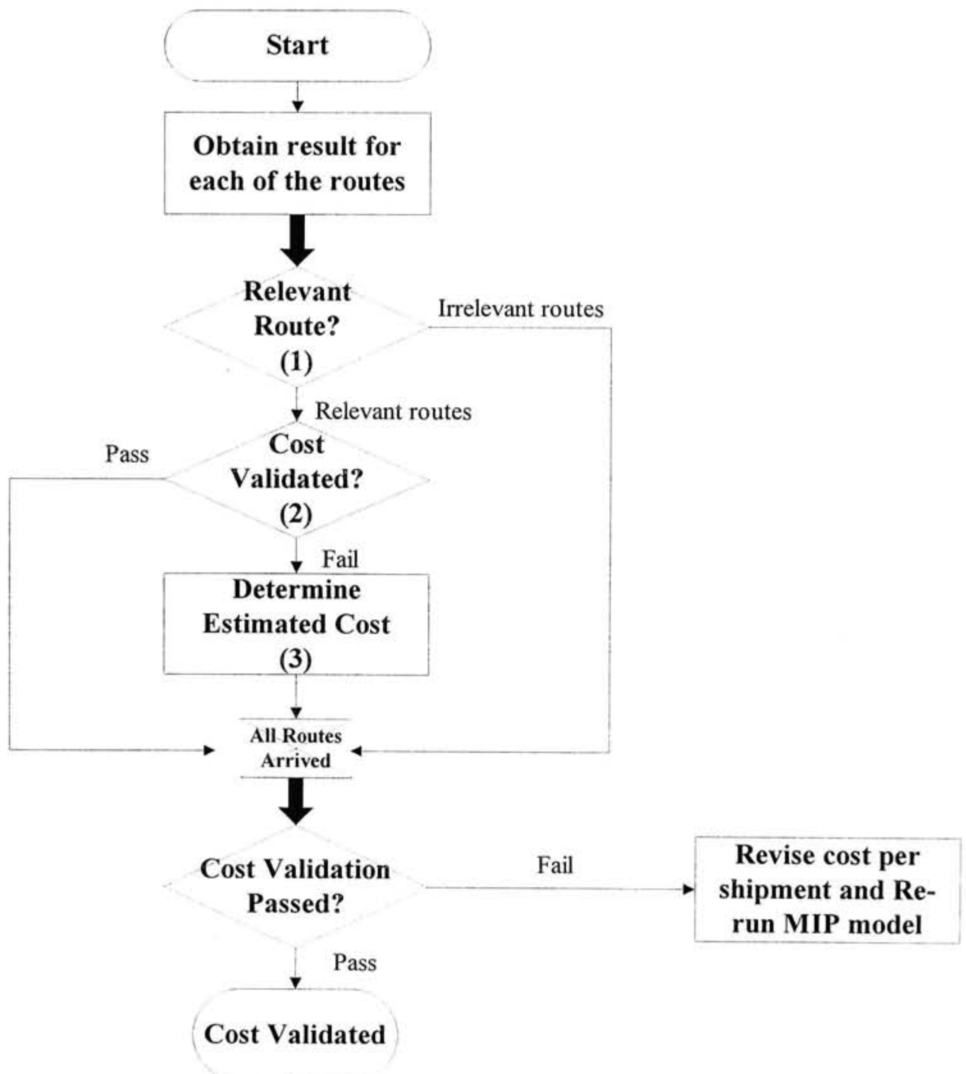


Figure 5-13 Unit Transportation Cost Validation Logic Flow

Figure 5-13 shows the flow logic for cost validation. The decision-maker tries to collect the simulated costs and compare with the cost inputs used in the MIP model. Throughout the cost validation, the decision-maker goes through each of the routings, and determines the major routes for validation based on the volume of the respective route and the zone from which it is originated. Accuracy of these transportation costs is then validated by comparing the cost used in the macro model, with those obtained in the simulation. The accuracy level is determined by means of a predefined tolerance limit at a reasonable confidence level. In case of validation fail, the decision-maker determines the new values of the routing cost and hence performs updates. By means of gathering results from all routings, conclusion and recommendation are drawn in deciding whether to re-run the MIP model with updated costs or proceed to the performance evaluation.

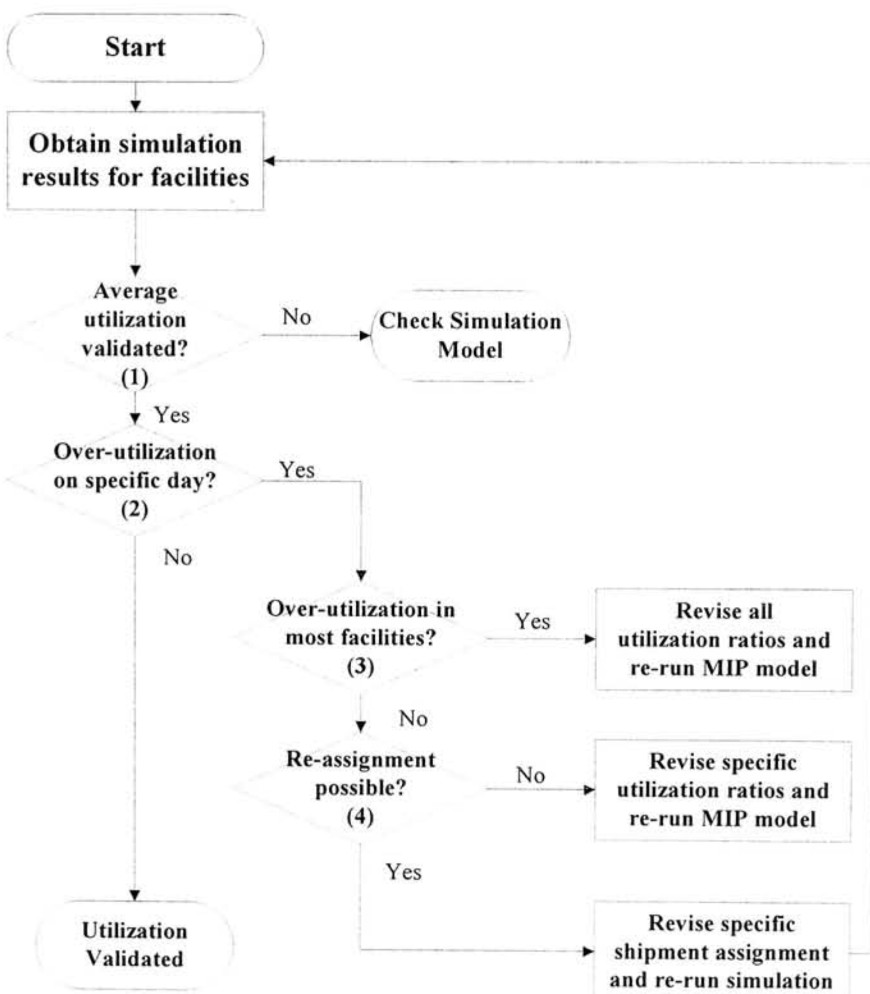


Figure 5-14 Utilization Validation Logic Flow

The flow logic of the utilization validation is illustrated in Figure 5-14. The decision-maker checks on both weekly and daily utilization for two types of facilities, satellite depots and service centers. In case over-utilization is examined, specific over-utilization is examined for appropriate modification either by making re-assignment for respective routes or decreasing the utilization factor and re-designign the network.

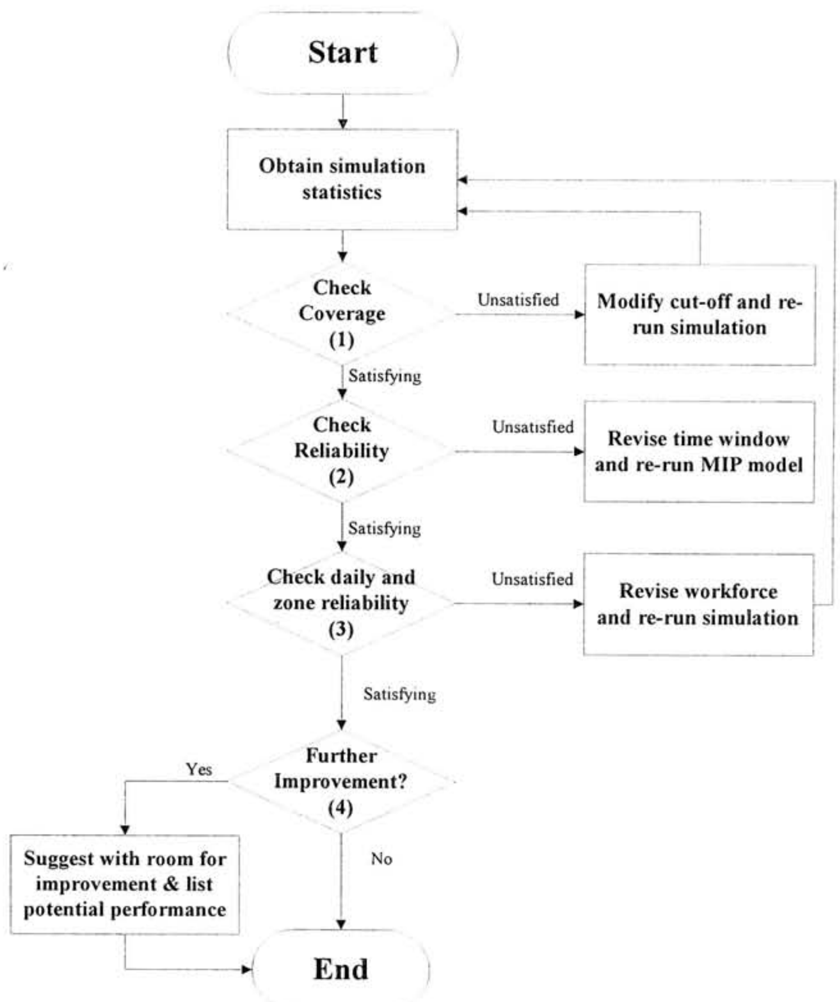


Figure 5-15 Performance Evaluation Logic Flow

The flow logic for performance evaluation is illustrated in Figure 5-15. The decision-maker first collects the statistics from simulation runs and evaluates the performance of the developed network in two manners: (1) service coverage, and (2) service reliability. Here the decision-maker determines the modification/improvement needed for the developed network.

In case of an unsatisfactory performance, the decision-maker seeks to determine the approach and/or parameters for feedback depending on the modification needed. Other than these two measures, other performance like operation slack time, facilities and courier activities are studied to determine if the network has rooms for improvement in performance.

Knowledge Representation and Coding. To implement the acquired knowledge in our Expert System, two types of diagrams and charts are used to represent these knowledge, namely, dependency diagram and decision chart.

According to the procedures and rules documented from the flow logic diagrams, dependency diagrams are drawn to help to encode the knowledge base. Dependency diagram is a type of diagram used for structural, backward chaining decision determination purpose. These diagrams indicate the basic reasoning process by what knowledge is required and how they are manipulated during the feedback tasks. Besides, the diagrams serves as the graphical model of the knowledge base system, showing the conditions required in concluding each action/goal. Figure 5-16 illustrates the dependency diagram in validating the unit transportation cost and determination of the new values of cost.

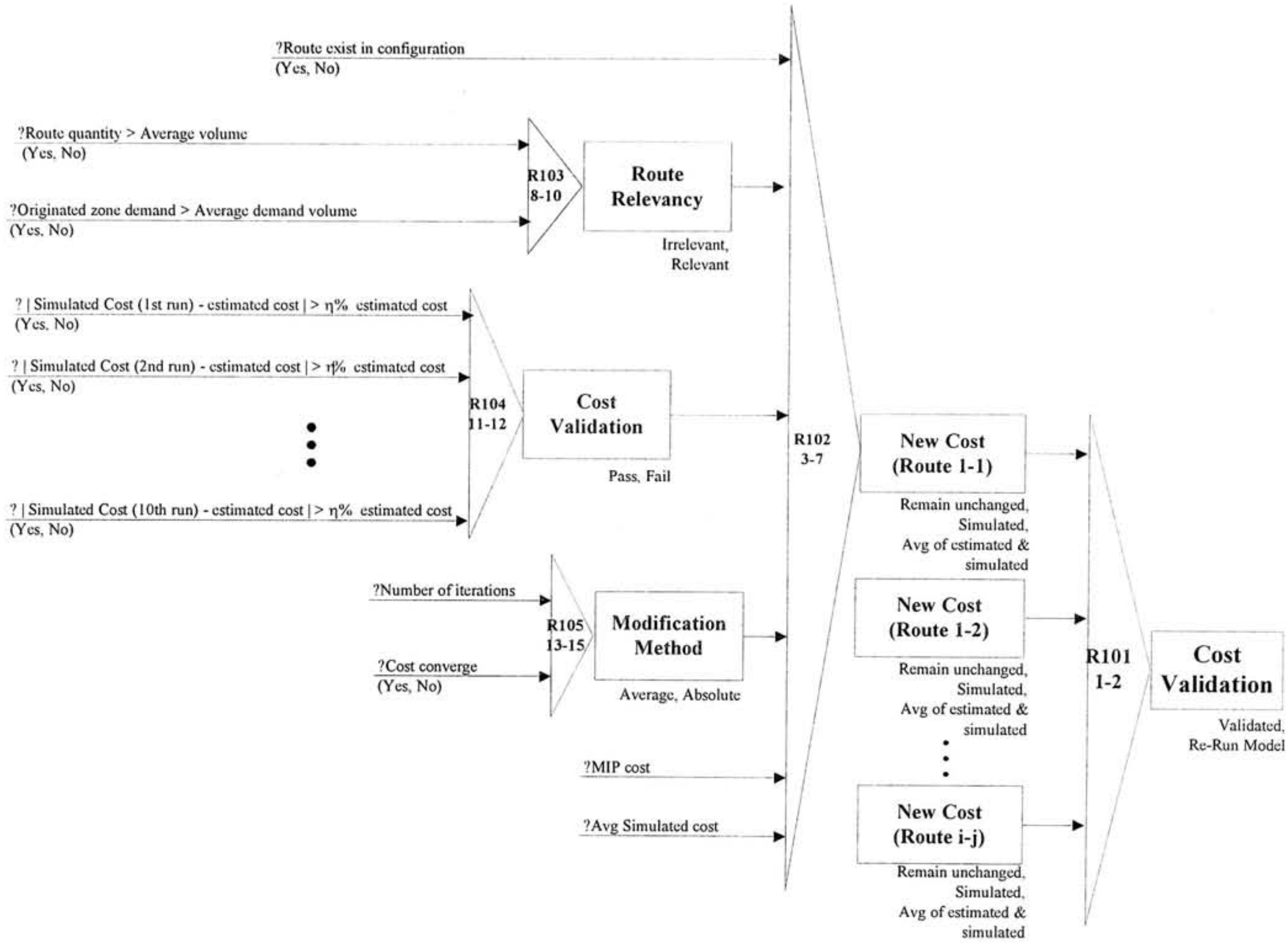


Figure 5-16 Dependency Diagram – Unit transportation cost validation

The triangles contain numbers that refer to the rules (rule set number and numbers of rules) in the knowledge base, which manipulate the conditions adjacent to the triangles – each triangle is attached with one decision chart. The boxes and arrows adjacent to a triangle show the name of the conditions to which they are related and to which they are manipulated. The question marks indicate questions asked by the system (values either seeks from the database reference from DBMS in DSS subsystem or by prompting the decision-maker). Acceptable values for each phase in the process are given under the input arrows and boxes. The recommendation to be made by the system during a consultation is named under the final decision box in the diagram. Referring to the above diagram, the results of the unit transportation cost validation is determined by the new cost of each route, which concluded by different conditions.

As a companion for the dependency diagram, decision chart is prepared for each of the rule-set triangle. Figure 5-17 illustrates one of the decision charts for the rule sets of determining the unit transportation cost modification action. Here the decision chart gives two information in drawing the conclusion. First, it provides a list of all possible combinations of condition values and hence the respective conclusion. For example, in case of route existence equals to "no", the unit transportation cost modification is concluded as "remain unchanged". Second, it states the sequence of how the rules are fired during the value seeking process, i.e. the system will try to conclude the action/goal with the first rule stated and then the second and third until it comes up with a value.

Decision Chart - R102 (New Cost of Route i - j)

Conditions/Rules	3	4	5	6	7
Route Existence	N	Y	Y	Y	Y
Route Relevancy		N	Y	Y	Y
Cost Validation Result			P	F	F
Modification Method				ABS	AVG
MIP Cost					MIP
Average Simulated Cost				ASIM	ASIM
New Cost of Route i - j	RM	RM	RM	AS	AT

Key:
NA - Not Applicable
RM - Remain Unchange
AS - Set as Average Simulated Cost
AT - Average of Two Costs
MIP - MIP Cost
ASIM - Average Simulated Cost

Figure 5-17 Decision Chart –Unit transportation cost Modification

For the full sets of the dependency diagrams and decision charts for the system prototype, refer to the appendix B and appendix C respectively.

Knowledge Base Execution. With the help of the dependency diagrams and decision charts, knowledge are coded in forms of "If-Then" rules. Figure 5-18 illustrates the rules for transformed rule set for the cost update action. For the full sets of rules, refer to appendix D.

These rule sets are then implemented in a backward chaining PC-based expert system shell, M.4 for the system prototype and testing. Implemented system prototype is divided into three functions, unit transportation cost validation, facility utilization validation and performance evaluation. These two system modules cope with the DSS subsystem to perform the validation and evaluation for the feedback throughout the network planning.

```
/* -------------------------------------*/
/* Rule Set 102 - New Cost */
/* -------------------------------------*/

kb3:
if route_existence-I-J = no
then new_cost-I-J = remain unchanged.

kb4:
if route_existence-I-J = yes
and route_relevancy-I-J = irrelevant
then new_cost-I-J = remain unchanged.

kb5:
if route_existence-I-J = yes
and route_relevancy-I-J = relevant
and cost_validation-I-J = pass
then new_cost-I-J = remain unchanged.

kb6:
if route_existence-I-J = yes
and route_relevancy-I-J = relevant
and cost_validation-I-J = fail
and modification_method-I-J = abs
and average_simulated_cost-I-J = ASIM
then new_cost-I-J = simulated cost.

kb7:
if route_existence-I-J = yes
and route_relevancy-I-J = relevant
and cost_validation-I-J = fail
and modification_method-I-J = avg
and mip_cost-I-J = MIP
and average_simulated_cost-I-J = ASIM
then new_cost-I-J = Average of estimated and simulated.
```

Figure 5-18 Rule Sets in Expert System M.4

CHAPTER 6

6. EVALUATION AND ANALYSIS

In this chapter, we try to demonstrate the functions of the prototype of the Expert Decision Support System (EDSS), and intend to evaluate and analyze our EDSS in a network planning scenario for prototype test. Then, we generalize the advantages and limitations of the proposed EDSS for two stage operations planning.

6.1 Test Scenario for Network Planning

The scenario is considering a design of distribution network having 33 demand zones, 15 candidates for depots and 9 of which are potential sites for service centers. Inputs of macro planning model includes: demand profile, fixed and variable cost estimates, travel and processing times, capacity alternatives of installations, together with policy of no more than 85% facility utilization and a cut-off time of 5:15 p.m. While simulation inputs includes: hourly-based customer request, shipment patterns & characteristics, process and transportation time distributions, workforce level, and operation policy including cut-off time for each zone.

6.1.1 Consultation Process

All data inputs for both MIP and simulation model are stored in the form of spreadsheet files managed by MS Access. To begin with, the decision-maker initiates the network planning by issuing a command in the Decision Support System (DSS) subsystem. Hence, the DSS subsystem generates the MPSIII model files and initiate model execution.

Developed network are displayed to the decision-maker and used, together with the operations details and parameters, to convert into a set of input files for simulation. To validate and evaluate the developed network, these intermediate results, including the network configuration and simulation statistics are passed to the Expert System (ES) subsystem for interpretation and hence determining the appropriate feedback actions. During the feedback process, the ES subsystem performs validation and evaluation in three steps: (1) unit transportation cost validation, (2) facility utilization validation, and (3) network performance evaluation. For each step, the ES draws conclusion to each task and displays the recommendation to the decision-maker. Such recommendations are used to launch respective programs in the DSS subsystem for necessary modifications and model evaluation in the next iteration.

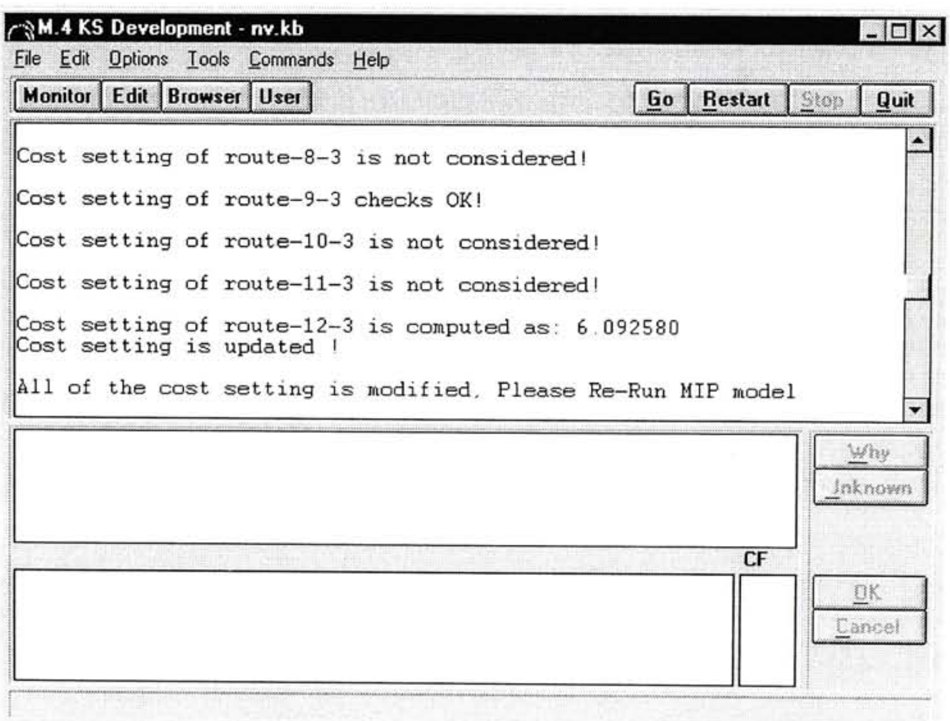


Figure 6-1 Network Validation Consultation – Unit Transportation Cost Validation

In the process of network validation, the system validates the cost and facility utilization according to the predefined rules and simulated results (Figure 6-1). The system first checks with the relevancy of each route, performs necessary unit cost update and concludes action for cost validation. Hence, the system checks with the utilization of each facility and suggests the appropriate action in case over-utilization is examined. Iterative process is repeated until both cost and facility utilization are validated.

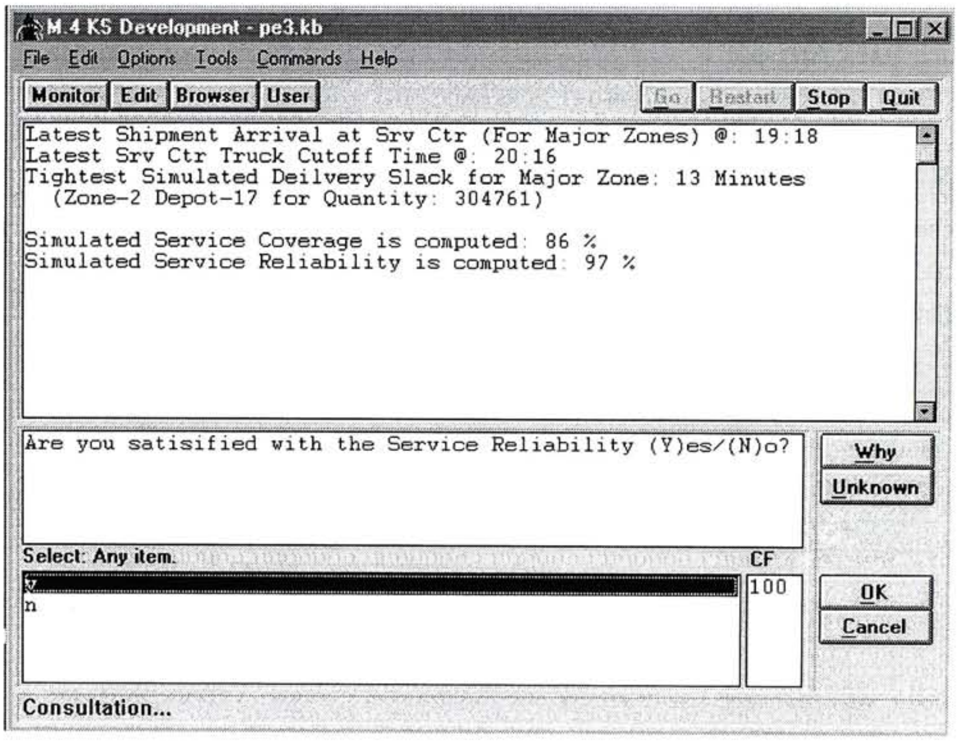


Figure 6-2 Performance Evaluation Consultation

With the validation completed, the system starts to examine the performance measures to determine if modification of the network is recommended. Two major measures, service coverage and service reliability, are computed from the simulation statistics and displayed to the decision-maker and prompts for satisfaction (Figure 6-2). Together with the simulated statistics, the ES determines the appropriate feedback modification and initiates the next iteration.

6.1.2 Consultation Results

During the test scenario, fourteen iterations were gone through in coming with the final design. Figure 6-3 summarizes the iteration histories in the planning process.

Itns	Problems	Suggested Rectification	Actions
1-5	Failed in unit cost validation	Modify unit transportation cost	Re-run MIP model with modified unit cost
6	Over-utilization examined in specific facilities	Route re-assignment to alleviate over-utilization	Re-run simulation model with new route re-assignment
7	Unsatisfactory service coverage	Delay operational cutoff	Re-run simulation model with new cutoff
8	Unsatisfactory overall service reliability	Re-design a more time efficient network	Re-run MIP model with next binding time window
9-12	Failed in unit cost validation	Modify unit transportation cost	Re-run MIP model with modified unit cost
13	Unsatisfactory service reliability in specific zone	Increase workforce for respective zone	Re-run simulation model with new workforce
14	Nil	N/A	Finalize network design

Figure 6-3 Summary of testing scenario iteration history

Throughout the test scenario, many of the iterations (Itns 1-5 and 9-11) went to unit cost validation. As the resulted network configurations were sensitive to the unit transportation costs, modification of the unit cost during the iterations gave new configurations, which needed to be re-validated. The cost validation required considerable numbers of iteration for a cost validated network. Figure 6-4 gives the cost validation iteration summary.

Itn	Total Cost Difference	No. of Total Routes	No. of Relevant routes	No. of unit cost need update
1	21%	38	12	6
2	22%	38	10	7
3	18%	40	11	5
4	12%	39	12	3
5	10%	41	10	3
6	3%	40	11	0

Figure 6-4 Cost Validation Summary

Validation throughout iteration 1 to 6 showed a converging trend and finally gave a cost validated network in the 6th iteration. However, the convergence of the unit cost is not guaranteed. In some scenario, unit cost modification may result in divergence and looped iterations, especially if modification is made arbitrarily. Hence, the control of the numbers of iteration in the cost validation is addressed which may require decision-maker intervention.

With the cost validated network in the 6th iteration, the ES system proceeded to check if the facilities were operating within the planned utilization. This was done in two folds by checking both weekly and daily utilization. Figure 6-5 illustrates the summary of the facilities utilization validation. The weekly facility utilization were within the limit, while two facilities were showing over-utilization during the Wednesday and Friday. To rectify the problem, the system seeked to find any feasible alternate route for re-assignment to alleviate the situation. All possible routes were listed and prompted for decision-maker's selection. Modifications were then made via the DSS and re-run simulation.

Facility	Weekly Utilization	Daily Over-utilization	Alternate route re-assignment
6	77%	Nil	N/A
9	81%	Day3 – 102% Day5 – 105%	Day3: 18-18, 20-18, 5-22, 30-22, 7-24, 24-24 Day5: 18-18, 20-18
11	73%	Nil	N/A
17	75%	Day3 – 101% Day5 – 108%	Day3: 11-11, 18-18, 20-18, 5-22, 30-22, 7-24 Day5: 5-22, 7-24

18	66%	Nil	N/A
22	74%	Nil	N/A
24	70%	Nil	N/A
29	67%	Nil	N/A
32	4%	Nil	N/A

Figure 6-5 Utilization Validation Summary

After the route re-assignment, both cost and utilization were validated in 7th iteration. Hence, the system proceeded to the performance evaluation for evaluating the service coverage and reliability respectively. During the 7th iteration, the coverage evaluation was found unsatisfactory (*Simulated-87% Vs Target-90%*). To enhance the coverage, ES recommended delaying the operation cut-off. Based on the shipment arrival pattern, the system determined the modification of cutoff (5:15 p.m. $\Rightarrow$ 5:35 p.m.), and initiated the next iteration by re-running the simulation model.

In the 8th iteration, with the new operation cutoff, the coverage was enhanced to 91% while the overall reliability was dropped to an unsatisfactory level (*Simulated-93% Vs Required-95%*). To restore the reliability, the system finally recommended to redesign a more time efficient network with a more stringent time window. Here the system seeked for the next binding time window and re-runs the MIP model for a new network configuration.

For any re-run in the MIP model, the new network configuration needed to be re-validated for the unit cost and facility utilization. Iteration 9-12 went to the cost validation of the new configuration. Fortunately, the unit cost validation showed a convergence trend and gave a cost validated network in the 13th iteration. With the cost validated configuration, the system proceeded to facility utilization and coverage evaluation with satisfactory results. However, minor reliability problem was examined in specific demand zone during performance evaluation. To restore the reliability, the system computed the respective workforce increment based on the shipment arrival pattern, and existing workforce level, and initiated the modification via the DSS for re-running the simulation. In the 14^{th} iteration, the developed network finally passed with all validations and evaluations and the system concluded with the final network design.

6.2 Effectiveness of EDSS in Network Planning

The exercise of the PC-based prototype offered a chance to test the feasibility and effectiveness of our proposed system design. In this practice, the EDSS was proved to release the loading from decision-makers in several ways: (1) the system provides an easy-to-use media to manage data, result and iteration histories. Decision-makers are hence release considerable time and effort from heavy data management activities, like data set preparations, updates, and retrievals, etc. Besides, as EDSS keeps track of each of the results and iteration histories in its database, decision-maker can easily trace back to the pass designs, which enable rapid reference during the network planing process. (2) The EDSS offers simple but useful model manipulation functions for model execution and modification. These programmed modules unload decision-maker from the complex syntax of both MIP model and simulation model. The parameter conversion module releases decision-makers from trivial but prone-to-error model input preparation. (3) The ES subsystem assists decision-makers during the feedback process by mimicking a field expert. With the accumulated knowledge and expertise during the network planning process, decision-makers can rapidly perform the network validation, determine and update the parameter/data. Besides, the intelligence of conducting the performance evaluation assists both experienced and inexperienced decision-maker in deciding the approach and/or parameters for feedback iterations.

All these activities require decision-makers for considerable time, effort, specific expertise and knowledge accumulated in the absence of the EDSS support. We believed this system is claimed to be effective by providing both data & model manipulation function together with an intelligence guidance during the feedback process.

6.3 Generalized Advancement and Limitation

From the prototype implemented for the network planning design, we conclude the major improvements comparing with the existing manual approach. On the other hand, we also try to point out the potential limitations which demand future efforts and improvement.

Better Data and Model Management

The use of Database Management System (DBMS) and Modelbase Management System (MBMS) provide a better means of data management and model manipulation respectively. As most of the two stage operations planning designs require decision-makers in dealing with comprehensive data inputs and to perform frequent model manipulation, the introduction of the data and model manipulation function releases decision-makers from such time consuming activities. As a result, decision-makers can focus on other critical and prioritized decisions.

Iterative Activities Guidance Support

The EDSS supports the iterative use of both models in two ways. The system database keeps track of the results and iteration histories of the planning design to, which decision-maker can easily trace back and hence make reference during the feedback process. Besides, with the accumulated knowledge and expertise in the ES subsystem, decision-makers are assisted throughout the validation and evaluation process. With such guidance, inexperienced decision-maker is supervised along with the recommendations while the experienced decision-makers / modelers uptake the suggestion as advice for feedback considerations.

Flexible and Cumulative Knowledge for Planning Design

As the knowledge and heuristics for the strategic planning design are modeled in the form of knowledge base entries, decision-makers can easily change or add new criteria or rules for the validation or evaluation process. Such characteristics enable modeler to change their planning strategies easily and leave the potential for the system self-learning ability.

Limited Model Modification Capability

On the other hand, unlike most of the Decision Support System, the model modification capability is limited. As most of the operations planing design problem, the simulation model is developed on specific simulation software and the model is comparatively sophisticated. Modification to such model requires particular simulation software manipulation with considerable knowledge and effort. As a result, detailed modification on similar simulation model can hardly be carried out by simple programming modules and hence be incorporated in our MBMS.

Required Decision-maker Intervention

During the process of the validation, there exists some cases that the validation cannot be achieved even after many iterations. The EDSS here can only prompt for decision-maker to solve such problem by manual. Besides, in case of evaluation, the system may not always suggest definite solution to decision-makers even after thorough considerations in its knowledge base. As a result, decision-makers may need to intervene the planning design and update with newly acquired knowledge frequently.

CHAPTER 7

7. CONCLUSION

While the two stage approach has been successfully used in many operations planning, the planning exercise requires a team of modeler(s) and decision-maker(s) who are (1) familiar with the two models, (2) able to manipulate data and the models, and (3) domain experts. In this paper, we have proposed an Expert Decision Support System (EDSS), which combines a Decision Support System (DSS) with an Expert System (ES) to facilitate the two stage approach for operations planning. In the joint use of optimization and simulation planning models, a modeler/decision-maker often has to manipulate the two models iteratively to gradually search for a planning solution. Furthermore, since many of the operations planning are for a long-term purpose, the recommended plan will need to be periodically/repeatedly evaluated or modified during the planning horizon. The other dimension of the difficulty is that the knowledge gained and lessons learnt for the planning exercise need to be documented and passed on to the next round. The proposed EDSS is able to document the knowledge in rules and further operationalize the knowledge by the expert system. On the other hand, the data and model manipulation is done by the DSS the decision-maker can focus on the planning domain issues.

An architecture which characterizes the conceptual design of the integration of the EDSS and the two stage planning approach is constructed. The DSS subsystem is responsible for data management and model manipulation, while the ES subsystem functions to elicit the heuristics, and experience accumulated to assist the decision-maker throughout the planning design.

To illustrate the functionality and effectiveness of the proposed EDSS in the two stage planning approach, a PC-based prototype EDSS is built based on a real-life project for an express service network design. During the exercises, the system was able to manage data sets, proposed planning results and iteration histories which saved considerable time and effort from the decision-maker. More importantly, the ES subsystem has demonstrated its ability to capture the knowledge and provide intelligence guidance in conducting the feedback iterations. We believed that our EDSS has provide significant improvement in the use of the two stage approach for operations planning.

As the essence of the EDSS design goes to the intelligence guidance throughout the feedback process, the knowledge that guides for such activities should be cumulative and updated from time to time. Future studies could be conducted to focus on a self-learning mechanism for the ES subsystem.

BIBLIOGRAPHY

Aykin Turgut, "Networking polices for hub-and-spoke systems with application to the air transportation system", *Transportation Science*, Vol.29, pp.201-221 (1995).

Barnhart, C., and R. Schneur, "Air network design for express shipment service", *Operations Research*, Vol.44, pp.852-863 (1996).

Bidgoli, H., "Integration of technologies: An ultimate decision-making aid", *Industrial Management & Data Systems*, Vol.93, pp.10-17 (1993).

Butler, T.W., K.R, Karwan, J.R. Sweigart, and G.R. Reeves, "An Integrative Model-Based Approach to Hospital Layout", *IIE Transactions*, Vol.24, pp.144-152 (1992).

Carlson, R.C., J.C. Hershey, and D.H. Kropp, "Use of optimization and simulation models to analyze outpatient health care settings", *Decision Sciences*, Vol.10, pp.412-433 (1979).

Chang, Y., and R. Ponder, "The small package air freight industry in the united states: A review of the federal express experience", *Transportation Research*, Vol.13A, pp.221-229 (1979).

Cook, T., and P. Rao, "Airline terminal design using computer simulation", *IIE Transactions*, Vol.17, pp.314-319 (1985).

Cook, T., and R. Russell, "A simulation and statistical analysis of stochastic vehicle routing with timing constraints", *Decision Sciences*, Vol.9, pp.673-687 (1978).

Cheung, W., L.C. Leung, and Y.M. Wong, "Strategic Service Network Design for DHL Hong Kong", *Interfaces*, (forthcoming)

El-Najdawi, M.K., and A.C. Stylianou, "Expert support systems: Integrating AI technologies", *Communications of the ACM*, Vol.36, pp.55-65+ (1993).

Hueter J., and W. Swart, "An integrated labor-management system for Taco bell", *Interfaces*, Vol.28, pp.75-91 (1998).

Jungthirapanich, H., and C.O. Benjamin, "A knowledge-based decision support systems for locating a manufacturing facility", *IIE Transactions*, Vol.27, pp.789-799 (1995).

Kamoun, M., and R. Hall, "Design of express mail services for metropolitan regions", *Journal of Business Logistics*, Vol.17, pp.265-302 (1996).

Kanafani, A., and A. Ghobrial, "Airline hubbing - some implications for airport economics", Vol.19A, pp.15-27 (1985).

Keim, R.T., and S. Jacobs, "Expert Systems: The DSS of the Future?", *Journal of Systems Management*, Vol.37, pp.6-14 (1986).

Leung, L.C., S. Maheshwari, and W.A. Miller, "Concurrent Part Assignment and Tool Allocation in FMS with Materials Handling Considerations", *International Journal of Porduction Research*, V.31, pp. 117-138, (1993).

Leung, L.C., and W. Cheung, "An Integrative Decision Methodology For Designing And Operating An Air-Express Courier's Distribution Network", *Decision Science*, (Forthcoming)

Magnanti, T., and R. Wong, "Network design and transportation planning: Models and algorithms", *Transportation Science*, Vol.18, pp.1-55 (1984).

Moore, E.W., J.M. Warmke, and L.R. Gorban, "The Indispensable Role of Management Science in Centralizing Freight Operations at Reynolds Metals Company", *Interfaces*, Vol.21, pp.107-129 (1991).

Moore, J.S., "A Prototype Expert Decision Support System for the Market Appraisal of the Single Family Residence", *Decision Science*, Vol.23, pp.1408-1421 (1992).

Nolan, R.L., and M.G. Sovereign, "A Recursive Optimization and Simulation Approach to Analysis with an Application to Transportation Systems", *Management Science*, Vol.18, pp.B676-B690 (1972).

O'Kelly, M., "The location of interacting hub facilities," *Transportation Science*, Vol.20, pp.92-106 (1986).

Rich, E., *Artificial Intelligence*, McGraw Hill, (1983).

Scott Morton, M.S., *Management Decision Systems*: Computer Based Support for Decision Making, Division of Research, Harvard University, Cambridge, Mass., (1971).

Sengupta, S., and R. Combes., "Optimizing and simulating Kraft general foods' manufacturing environment", *IIE Solutions*, Vol.27, pp.30-35. (1995).

Turban, E., *Decision Support and Expert System: Management Support System*, Macmillan, (1988).

Wellons, M.C., A.V. Sapre, A.I. Chang, and T.L. Laird, "On-line power plant optimization improves Texas refiner's bottom line", *Oil and Gas Journal*, Vol.92, pp.53-58. (1994).

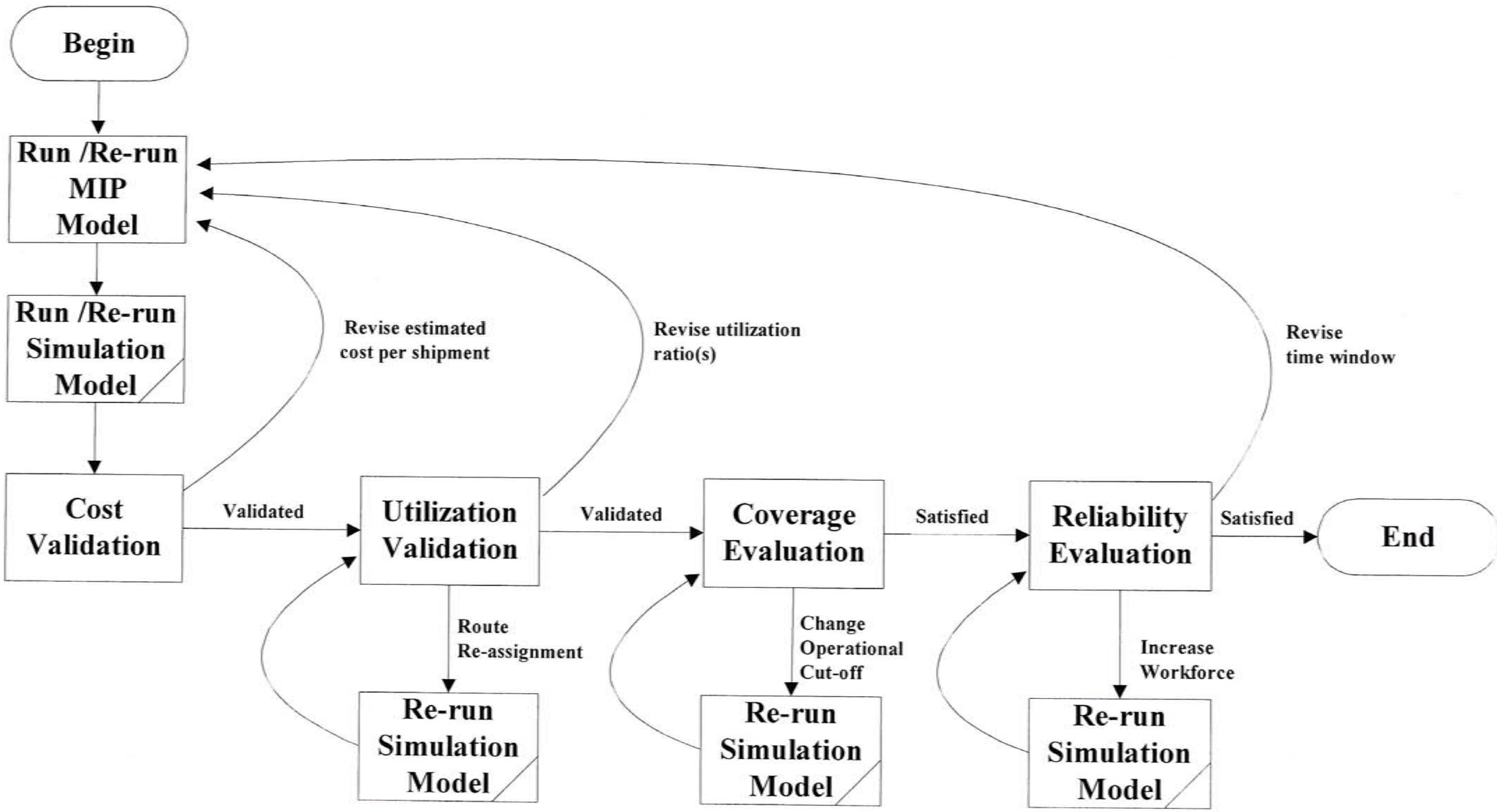
Yaghmai, N.S., and J.A. Maxin, "Expert system: A tutorial", *Journal of the American Society for Information Science*, Vol.35, pp.297-305 (1984).

APPENDICES

APPENDIX A

LOGIC FLOW DIAGRAMS

Feedback Logic Flow



Cost Validation Logic

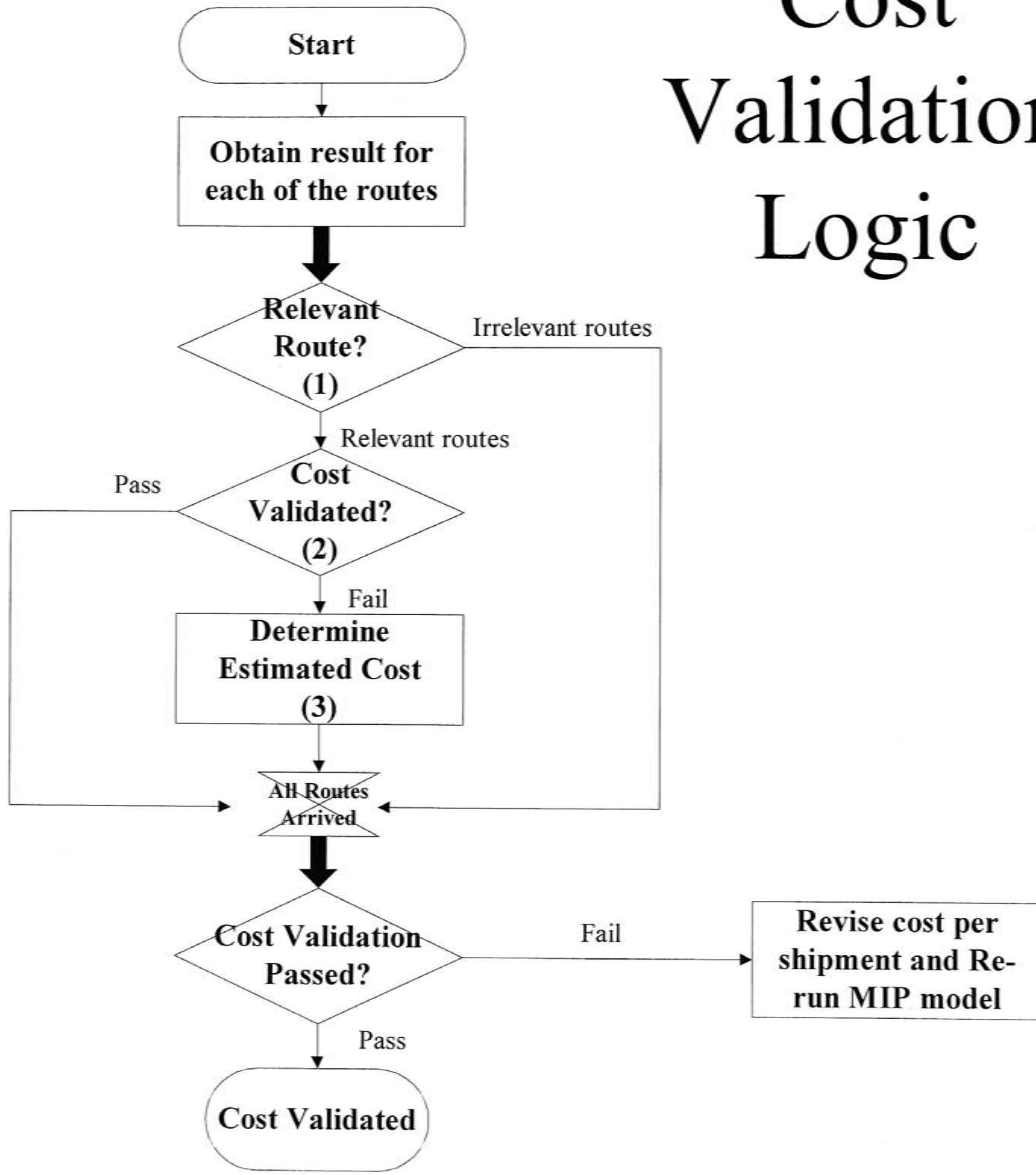
Step 0: Initialize parameters
- tolerance limit = η%
- simulation iterations = *n*

Step 1: Select major routes
- If a route rij is major if its assignment ≥ average volume, select rij as major route

Step 2: Validate estimated costs ec with simulated costs sc for major routes
- If |scij - ecij|/ecij ≥ η%, validation for rij fails

Step 3: Modify ecij for rij in macro data set
- If number of iterations ≤ *n*, set ecij = scij; else, set ecij = (scij + ecij) /2

Step 4: Conclude cost validation
- If all major routes are validated, validation is completed; else re-run macro model with updated cost



Utilization Validation Logic

Step 0: Initialize parameters

- Acceptable utilization level = α%;
- acceptable percentage of over-utilized facilities = β.

Step 1: Check average utilization

- Simulated utilization Vs calculated estimated

Step 2: Check daily utilization

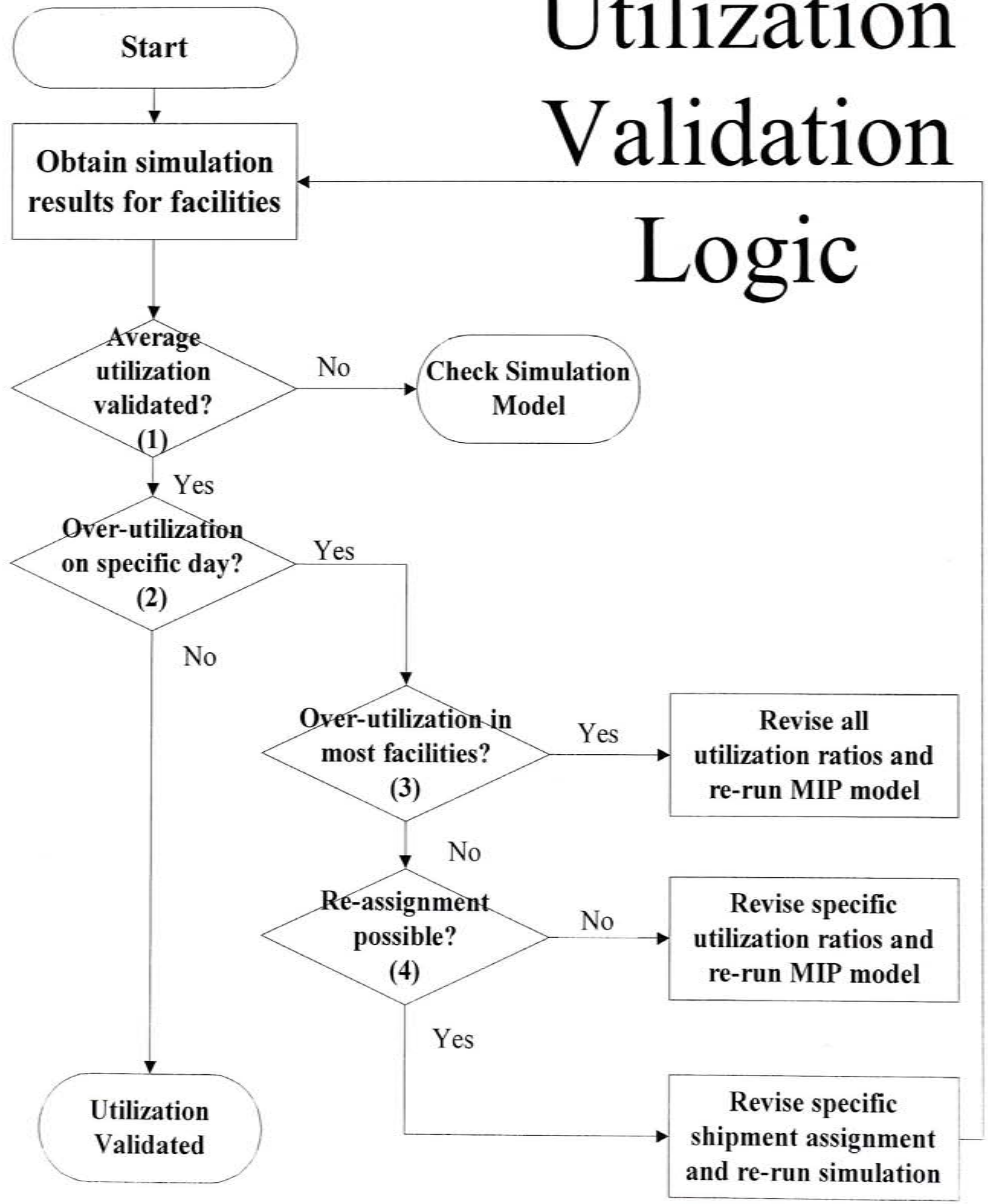
- If daily utilization SUdj $\geq \alpha\%$, j is over-utilized on day d.

Step 3: Check overall facility capacity tightness

- If % of over-utilized facilities $\geq \beta$, overall facility capacity is tight

Step 4: Check re-assignment for over-utilized facilities

- Re-assignment is infeasible if (1) no spare capacity in adjacent facilities, and (2) no alternate route meets time constraint.



Performance Evaluation Logic

Step 0: Initialize parameters.
- Set required service coverage level = δ %
- Set required service reliability level = σ %

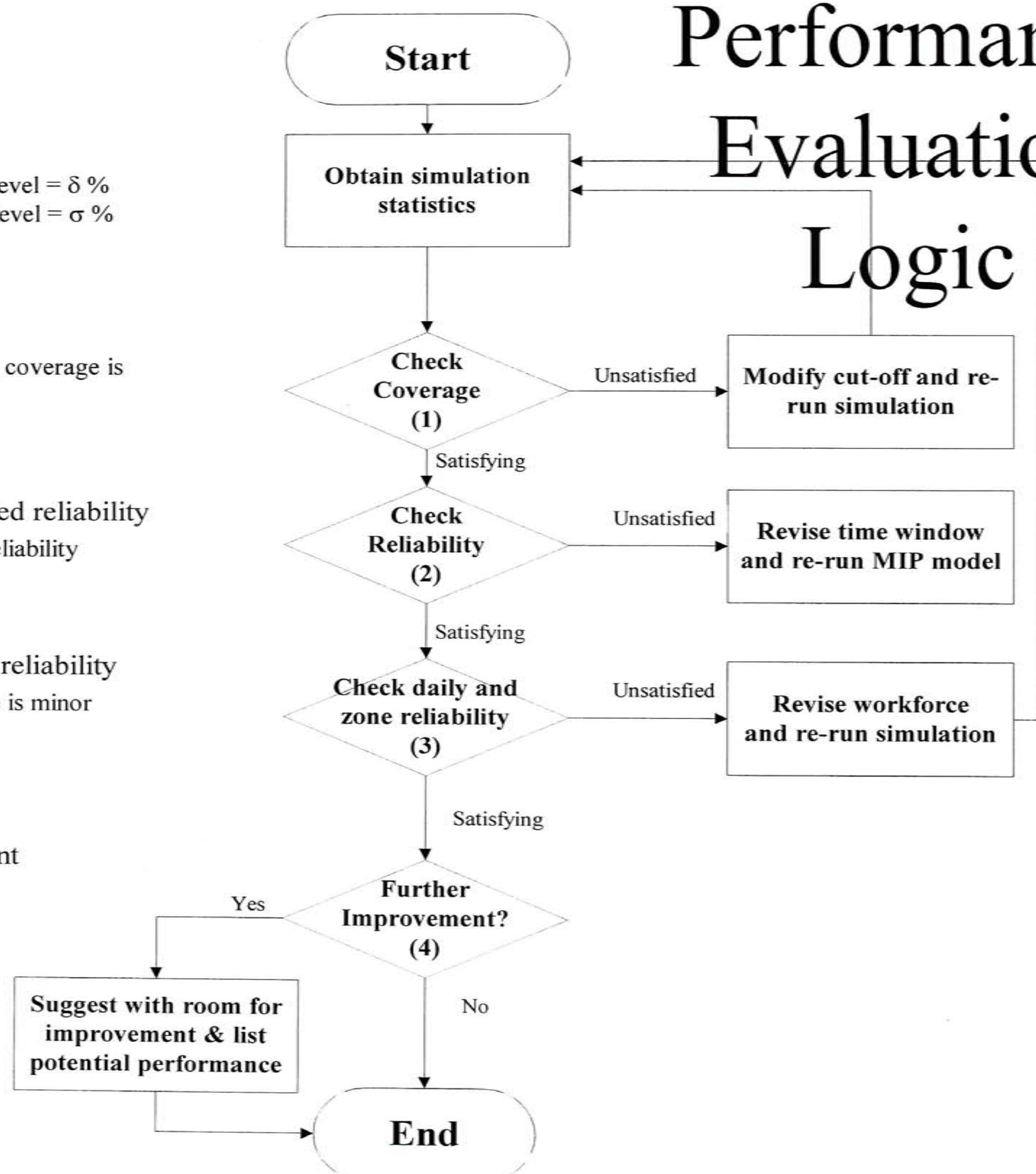
Step 1: Check Coverage
- If simulated coverage is ≥ δ %, coverage is satisfied

Step 2: Check overall simulated reliability
- If overall reliability R < σ %, reliability check fails

Step 3: Check daily and zone reliability
- daily reliability Rd < σ %, there is minor reliability problem on day d

Step 4: Check for Improvement

Suggestions for improvement:
- Slack time in service center
- Slack time for pickup

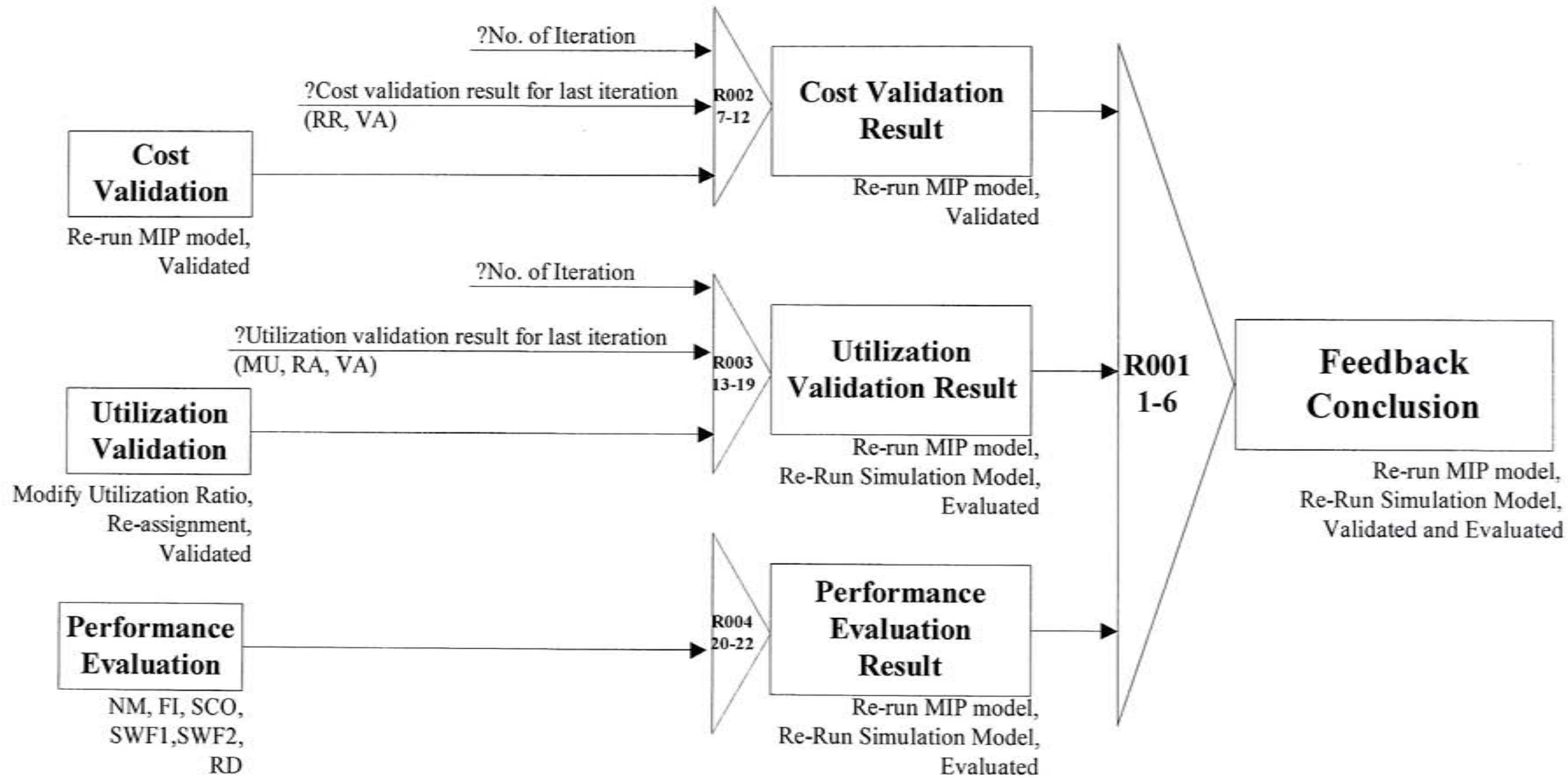


APPENDIX B

DEPENDENCY DIAGRAMS

Dependency Diagram

Overall Feedback

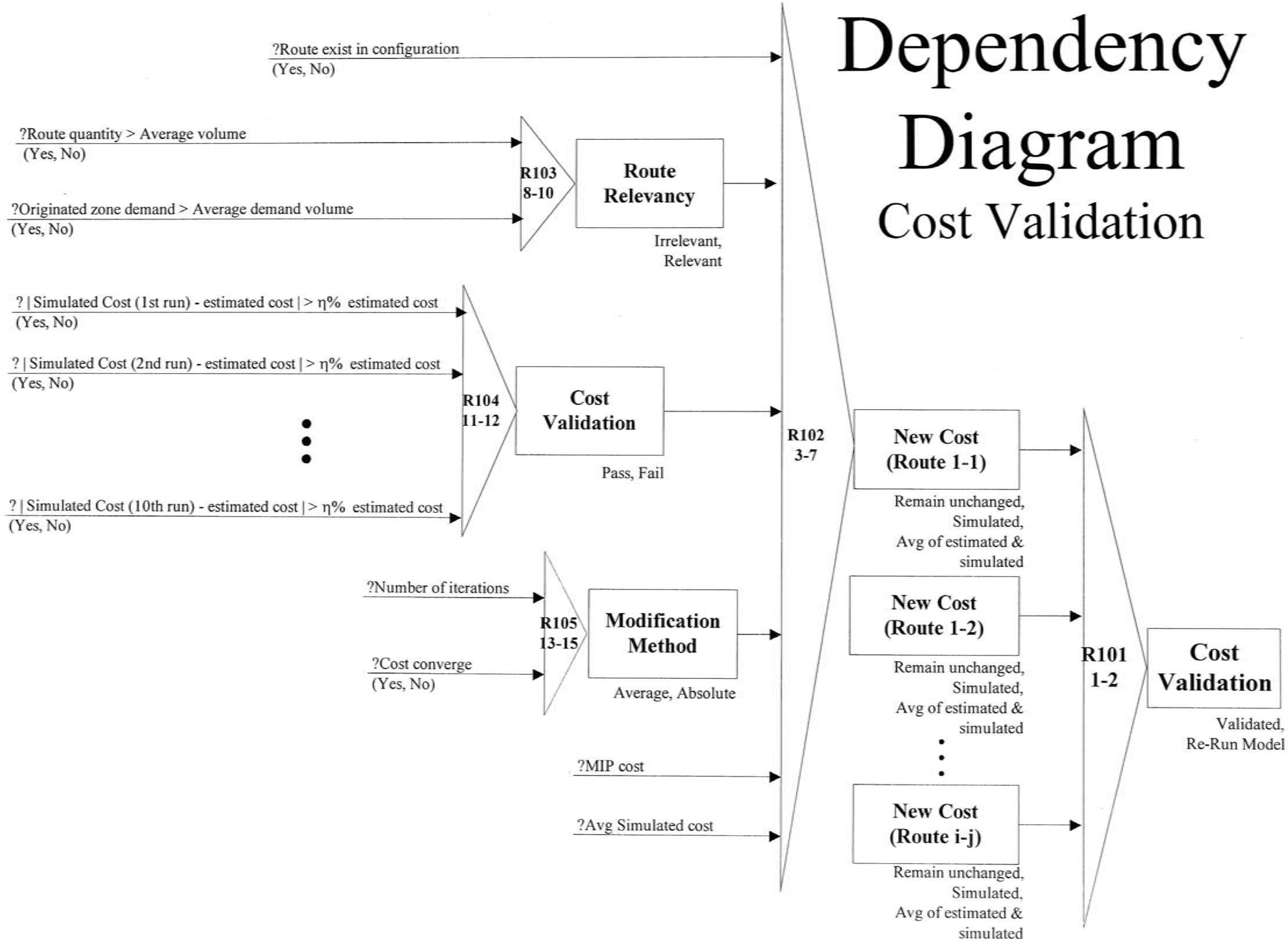


KEY:

VA - Validated
RR - Re-Run MIP Model
MU - Modify Utilization Ratio
RA - Re-assignment
NM - No modification needed
FI - Prompt for potential improvement
SCO - Re-run simulation with new operational cut-off
SWF1 - Re-run simulation with increase workforce in
SWF2 - Re-run simulation with overall increase workforce specific zones
RD - Re-run MIP model with next binding time window

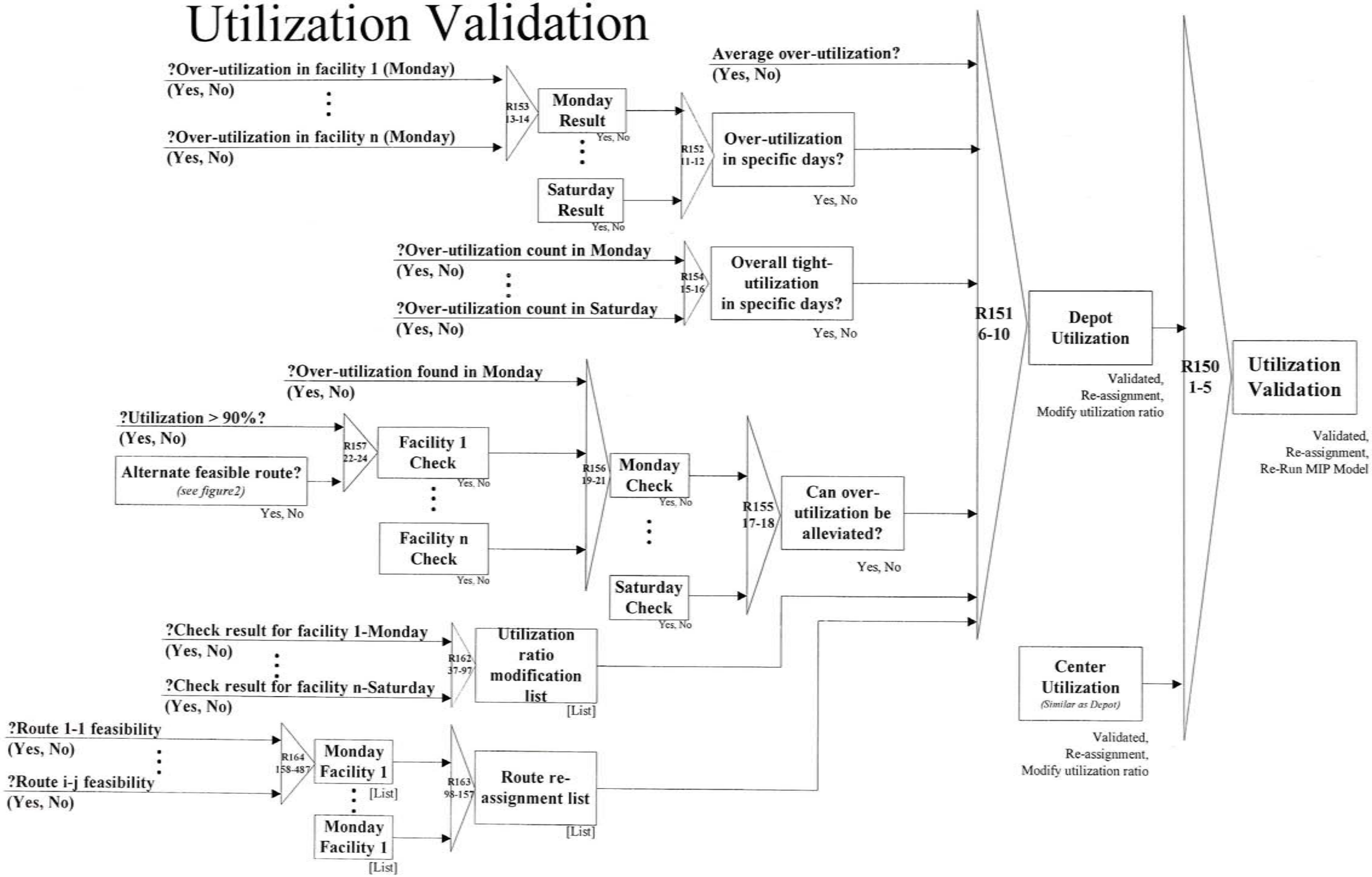
Dependency Diagram

Cost Validation



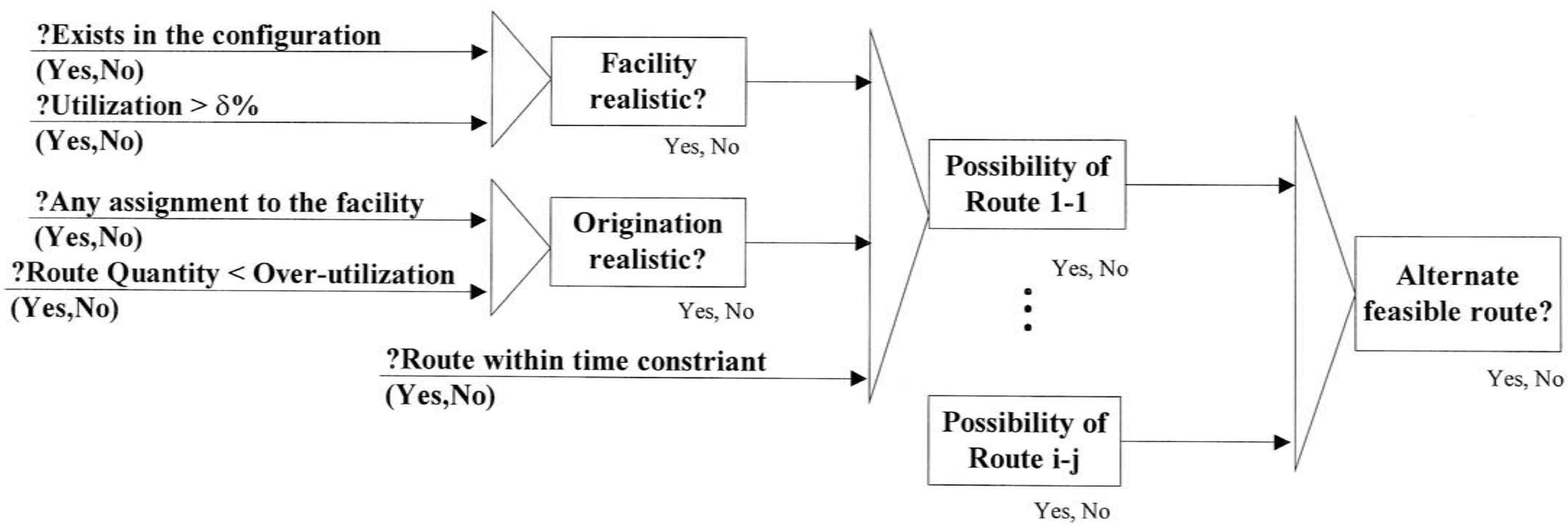
Dependency Diagram

Utilization Validation



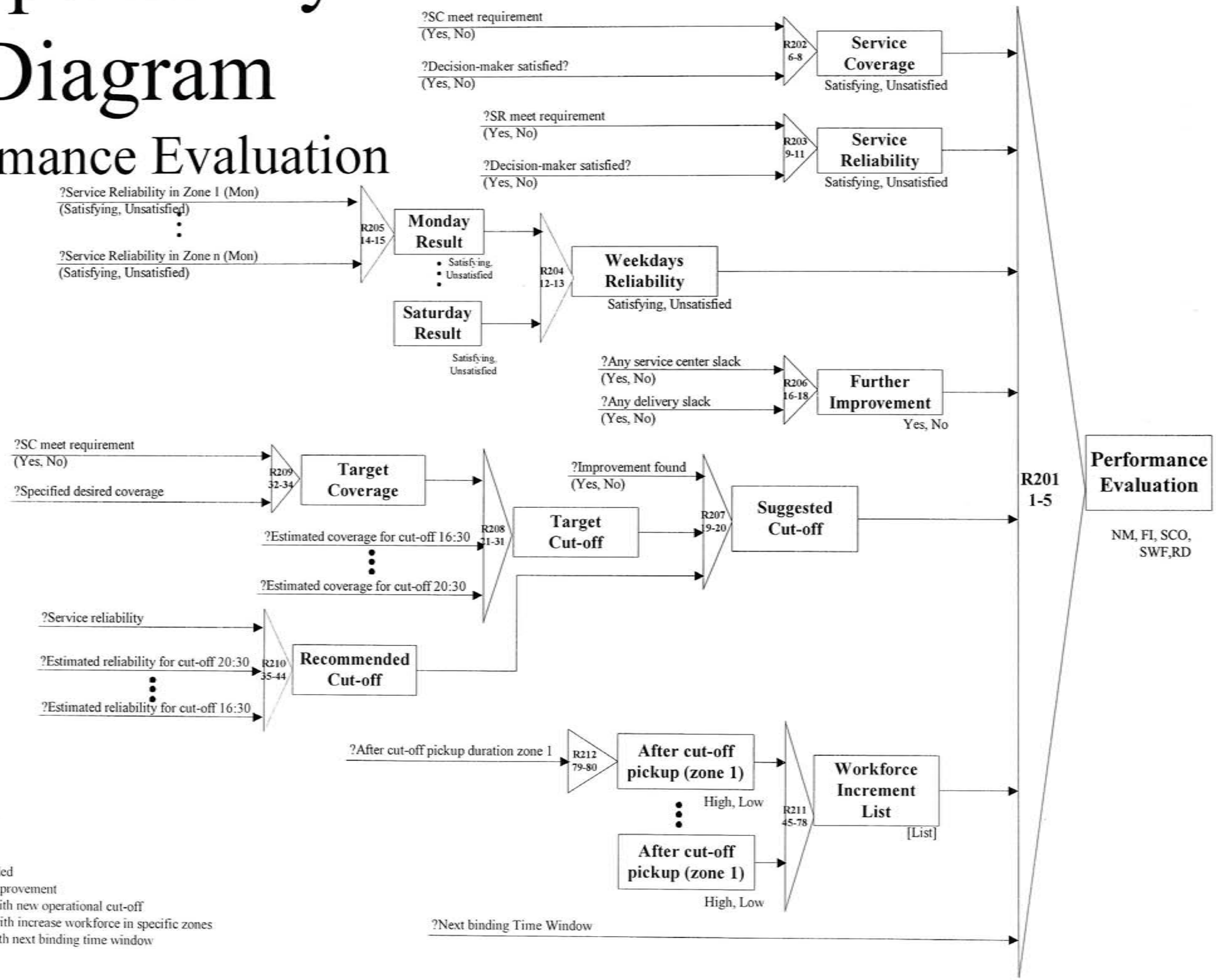
Dependency Diagram

Utilization Validation (*Con't*)



Dependency Diagram

Performance Evaluation



KEY:

NM - No modification needed
FI - Prompt for potential improvement
SCO - Re-run simulation with new operational cut-off
SWF - Re-run simulation with increase workforce in specific zones
RD - Re-run MIP model with next binding time window

APPENDIX C

DECISION CHARTS

Decision Charts

Decision Chart - R001 (Feedback Conclusion)

Conditions/Rules	1	2	3	4	5	6
Cost Validation Result	RR	VA	VA	VA	VA	VA
Utilization Validation Result		RR	RA	VA	VA	VA
Performance Evaluation Result				RR	RS	VA
Feedback Conclusion	RR	RR	RS	RR	RS	VA

Key:
VA - Validated
RS - Re-run Simulation Model
RR - Re Run MIP Model

Decision Chart - R002 (Cost Validation Result)

Conditions/Rules	7	8	9	10	11	12
No. of Iteration	1	1	>5			
Cost Validation Result for Last Iteration				VA	RR	RR
Cost Validation	VA	RR			VA	RR
Cost Validation Result	VA	RR	VA	VA	VA	RR

Key:
VA - Validated
RS - Re-run Simulation Model
RR - Re Run MIP Model

Decision Chart - R003 (Utilization Validation Result)

Conditions/Rules	13	14	15	16	17	18
No. of Iteration	1	1	1			
Utilization Validation Result for Last Iteration				VA	RR/RS	RR/RS
Utilization Validation	VA	RA	MU		VA	RA
Utilization Validation Result	VA	RS	RR	VA	VA	RS

Conditions/Rules	19
No. of Iteration	
Utilization Validation Result for Last Iteration	RR/RS
Utilization Validation	MU
Utilization Validation Result	RR

Key:
VA - Validated
RA - Re-assignment
MU- Modify Utilization Ratio
RS - Re-run Simulation Model
RR - Re Run MIP Model

Decision Chart - R004 (Performance Evaluation Result)

Conditions/Rules	20	21	22
Performance Evaluation	MN/FI	SCO/ SWF1/ SWF2	RN
Performance Evaluation Result	VA	RS	RR

Key:
VA - Validated
RS - Re-run Simulation Model
RR - Re Run MIP Model
NM - No modification Needed
FI - Further improvement
SCO - Re run simulation with new operational cut-off time
SWF1 - Re-run Simulation with workforce increased in specific zones
SWF2 - Re-run Simulation with general workforce increase

Decision Charts

Decision Chart - R150 (Utilization Validation)

Conditions/Rules	1	2	3	4	5
Validation for Depot	VA	MU	MU	*	RA/VA
Validation for Center	VA	MU	*	MU	RA/VA
Utilization Ratio Modification List		*	*	*	
Route Re-assignment List					*
Utilization Validation	VA	RR	RR	RR	RA

Key:
VA - Validated
RA - Re-assignment
MU - Modify Utilization Ratio
RR - Re-run MIP Model
* - ANY VALUE

Decision Chart - R151 (Depot Utilization)

Conditions/Rules	6	7	8	9	10
Average Over-utilization	Y	N	N	N	N
Over-utilization in Specific Weekdays		N	Y	Y	Y
Over-utilization in Most Facilities			Y	N	N
Route Re-assignment Possibility				N	Y
Validation for Depot	ERR	VA	MU	MU	RA

Key:
ERR - Prompt for error, and check simulation model
Y - Yes
N - No
VA - Validated
RA - Re-assignment
MU - Modify Utilization Ratio
RR - Re-run MIP Model
* - ANY VALUE

Decision Chart - R152 (Over-utilization in Specific Day)

Conditions/Rules	11	12
Monday Result	N	*
Tuesday Result	N	*
......	N	*
Saturday Result	N	*
Over-utilization in Specific Day	N	Y

Key:
Y- Yes
N - No
* - ANY VALUE

Decision Chart - R153 (Specific Day Result)

Conditions/Rules	13	14
Over-utilization in facility 1	N	*
Over-utilization in facility 2	N	*
......	N	*
Over-utilization in facility n	N	*
Specific Day Result	N	Y

Key:
Y- Yes
N - No
* - ANY VALUE

Decision Charts

Decision Chart - R154 (Over-utilization in Most Facilities)

Conditions/Rules	**15**	**16**
Over-utilization Count in Monday	< (3/4 n)	*
Over-utilization Count in Tuesday	< (3/4 n)	*
……	< (3/4 n)	*
Over-utilization Count in Saturday	< (3/4 n)	*
Over-utilization in Most Facilities	N	Y

Key:
n - Numbers of total facility
Y- Yes
N - No
* - ANY VALUE

Decision Chart - R155 (Route Re-assignment Possibility)

Conditions/Rules	**17**	**18**
Check for Monday	Y/NA	*
Check for Tuesday	Y/NA	*
……	Y/NA	*
Check for Saturday	Y/NA	*
Route Re-assignment Possibility	Y	N

Key:
NA - Not Applicable
Y- Yes
N - No
* - ANY VALUE

Decision Chart - R156 (Check for Specific Day)

Conditions/Rules	**19**	**20**	**21**
Over-utilization in Specific Day	N	Y	Y
Check for facility 1		Y	*
……		Y	*
Check for facility n		Y	*
Check for Specific Day	NA	Y	N

Key:
NA - Not Applicable
Y- Yes
N - No
* - ANY VALUE

Decision Chart - R157 (Check for Specific Facility)

Conditions/Rules	**22**	**23**	**24**
Utilization > 90%	Y	N	N
Alternate Feasible Route		Y	N
Check for Specific Facility	N	Y	N

Key:
Y- Yes
N - No

Decision Chart - R158 (Alternate Feasible Route)

Conditions/Rules	25	26
Possibility of Route 1-1	N	*
Possibility of Route 1-2	N	*
......	N	*
Possibility of Route i-j	N	*
Alternate Feasible Route	N	Y

Key:
Y- Yes
N - No
* - ANY VALUE

Decision Chart - R159 (Possibility for Specific Route)

Conditions/Rules	27	28	29	30
Facility Realistic	N		Y	Y
Origination Realistic		N	Y	Y
Route within Time Constraint			N	Y
Possibility for Specific Route	N	N	N	Y

Key:
Y- Yes
N - No

Decision Chart - R160 (Facility Realistic)

Conditions/Rules	31	32	33
Exists in Configuration	N	Y	Y
Utilization >85%		N	Y
Facility Realistic	N	N	Y

Key:
Y- Yes
N - No

Decision Chart - R161 (Origination Realistic)

Conditions/Rules	34	35	36
Assignment to facility	N	Y	Y
Route Quantity < Over-utilization		Y	N
Origination Realistic	N	N	Y

Key:
Y- Yes
N - No

Decision Charts

Decision Chart - R162 (Utilization Ratio Modification List) [Multi-value Conclusion]

Conditions/Rules	37	38	...	97
Check result for facility 1 (Monday)	N		...	
Check result for facility 2 (Monday)		N	...	
....			...	
Check result for facility n (Saturday)			...	N
			...	
Utilization Ratio Modification List	Facility 1	Facility 2	...	Facility n

Key:
Y - Yes
N - No
* - ANY VALUE

Decision Chart - R163 (Route Re-assignment List) [Multi-value Conclusion]

Conditions/Rules	98	99	...	157
List of facility 1 (Monday)	Not Null		...	
List of facility 2 (Monday)		Not Null	...	
....			...	
List of facility n (Saturday)			...	Not Null
			...	
Utilization Ratio Modification List	Facility 1 (Mon) [List]	Facility 2 (Tue) [List]	...	Facility n (Sat) [List]

Key:
Not Null - (Contain a list of value)

Decision Chart - R164 (Feasible Route List of Specific Facility and Day) [Multi-value Conclusion]

Conditions/Rules	158	159	...	487
Route 1-1 Feasibility	Y		...	
Route 1-2 Feasibility		Y	...	
....			...	
Route i-j Feasibility			...	Y
			...	
Feasible Route List of Specific Facility and Day	Route 1-1	Route 1-2	...	Route i-j

Key:
Y - Yes

Decision Charts

Decision Chart - R101 (Cost Validation)

Conditions/Rules	1	2
New Cost (Route 1-1)	RM	*
New Cost (Route 1-2)	RM	*
……	RM	*
New Cost (Route i-j)	RM	*
Cost Validation	VA	RR

Key:
RM - Remain Unchanged
VA - Validated
RR - Re Run MIP Model
* - ANY VALUE

Decision Chart - R102 (New Cost of Route i - j)

Conditions/Rules	3	4	5	6	7
Route Existence	N	Y	Y	Y	Y
Route Relevancy		N	Y	Y	Y
Cost Validation Result			P	F	F
Modification Method				ABS	AVG
MIP Cost					MIP
Average Simulated Cost				ASIM	ASIM
New Cost of Route i - j	RM	RM	RM	AS	AT

Key:
NA - Not Applicable
RM - Remain Unchange
AS - Set as Average Simulated Cost
AT - Average of Two Costs
MIP - MIP Cost
ASIM - Average Simulated Cost

Decision Chart - R103 (Route Relevancy)

Conditions/Rules	8	9	10
Belong to Major Route	Y	N	N
Originated from Major Zone		Y	N
Route Relevancy	RE	RE	IR

Key:
Y - Yes
N - No
IR - Irrelevant
RE - Relevant

Decision Chart - R104 (Cost Validation Result)

Conditions/Rules	11	12
Simulated Cost - 1st run passed?	Y	*
Simulated Cost - 2nd run passed?	Y	*
……		
Simulated Cost - 10th run passed?	Y	*
Cost Validation Result	P	F

Key:
Y - Yes
N - No
P - Pass
F - Fail
* - ANY VALUE

Decision Chart - R105 (Modification Method)

Conditions/Rules	13	14	15
Numbers of Iteration	<3	>=3	>=3
Sign of Convergence		Y	N
Modification Method	ABS	ABS	AVG

Key:
Y - Yes
N - No
ABS - Absolute
AVG - Average

Decision Charts

Decision Chart - R201 (Performance Evaluation)

Conditions/Rules	1	2	3	4	5
Service Coverage	U	S	S	S	S
Service Reliability		U	S	S	S
Weekdays Reliability			N	S	S
Further Improvement				N	Y
Suggested Cutoff	*				
Workforce Increment List			*		
Next Binding Time Window		*			
Performance Evaluation	SCO	RN	SWF	NM	FI

Key:
U - Unsatisfied
S - Satisfying
Y - Yes
N - No
RN - Redesign Network
NM - No modification Needed
FI - Further improvement
SCO - Re run simulation with new operational cut-off time
SWF - Re-run Simulation with workforce increased in specific zones
* - ANY VALUE

Decision Chart - R202 (Service Coverage)

Conditions/Rules	6	7	8
SC Meet Requirement?	N	Y	Y
Decision-maker Satisfied?		N	Y
Service Coverage	U	U	S

Key:
U - Unsatisfied
S - Satisfying
Y - Yes
N - No
* - ANY VALUE

Decision Chart - R203 (Service Reliability)

Conditions/Rules	9	10	11
SR Meet Requirement?	N	Y	Y
Decision-maker Satisfied?		N	Y
Service Reliability	U	U	S

Key:
U - Unsatisfied
S - Satisfying
Y - Yes
N - No

Decision Chart - R204 (Reliability in Specific Day)

Conditions/Rules	12	13
Monday Result	N	*
Tuesday Result	N	*
……	N	*
Saturday Result	N	*
Reliability in Specific Day	N	Y

Key:
Y- Yes
N - No
* - ANY VALUE

Decision Charts

Decision Chart - R205 (Specific Day Result)

Conditions/Rules	14	15
Over-utilization in facility 1	N	*
Over-utilization in facility 2	N	*
……	N	*
Over-utilization in facility n	N	*
Specific Day Result	N	Y

Key:
Y- Yes
N - No
* - ANY VALUE

Decision Chart - R206 (Further Improvement)

Conditions/Rules	16	17	18
Any Srv Ctr Slack?	Y	N	N
Any Delivery Slack?		Y	N
Further Improvement	Y	Y	N

Key:
Y - Yes
N - No

Decision Chart - R207 (Suggested Cut-off)

Conditions/Rules	19	20
Improvement	N	Y
Target Cut-off	*	
Recommended Cut-off		*
Suggest Cut-off	TC	RC

Key:
Y - Yes
N - No
TC - Target Cut-off
RC - Recommended Cut-off
* - ANY VALUE

Decision Chart - R208 (Target Cut-off)

Conditions/Rules	21	22	23	24	25	26
Target coverage	NA	<=EC1	<=EC2	<=EC3	<=EC4	<=EC5
Target Cut-off	NA	CT1	CT2	CT3	CT4	CT5

Conditions/Rules	27	28	29	30	31
Target coverage	<=EC6	<=EC7	<=EC8	<=EC9	<=EC10
Target Cut-off	CT6	CT7	CT8	CT9	CT10

Key:
EC - Estimated Coverage
CT - Cut-off Time

Decision Charts

Decision Chart - R209 (Target Coverage)

Conditions/Rules	32	33	34
Does coverage meet requirement?		N	Y
Decision-maker satisfied with coverag	N		Y
Target Coverage	DC	RC	NA

Key:
Y - Yes
N - No
DC - Desire coverage specified
RC - Required coverage
NA - Not applicable

Decision Chart - R210 (Recommended Cut-off)

Conditions/Rules	35	36	37	38	39
Service Reliability	<ER10	<ER9	<ER8	<ER7	<ER6
Recommended Cut-off	CT10	CT9	CT8	CT7	CT6

Conditions/Rules	40	41	42	43	44
Service Reliability	<ER5	<ER4	<ER3	<ER2	<ER1
Recommended Cut-off	CT5	CT4	CT3	CT2	CT1

Key:
EC - Estimated Reliability
CT - Cut-off Time

Decision Chart - R211 (Workforce Increment List) [Multi-value Conclusion]

Conditions/Rules	45	46	47	…	78
After Cut-off Pickup Activities	N	Y	Y	Y	Y
After Cut-off Pickup(zone 1)		H		…	
After Cut-off Pickup(zone 2)			H	…	
…….				…	
After Cut-off Pickup(zone 33)				…	H
				…	
Workforce Increment List	NA	Zone1	Zone2	…	Zone33

Key:
H - High
NA - Not applicable

Decision Chart - R212 (After Cut-off Pickup in zone n)

Conditions/Rules	79	80
After Cut-off Pickup Duration of zone n	<60	>=60
After Cut-off Pickup Activities	L	H

Key:
L - Low
H - High

APPENDIX D

EXPERT SYSTEM RULES

AND CODING

```
/*=====================================================================*/
/*                                                                     */
/* Unit Transportation Cost Validation Rules                           */
/*                                                                     */
/*=====================================================================*/


kb_goal:
initialdata = [cv].

/* ----------------------------------*/
/* Rule Set 101 - Recommended Action */
/* ----------------------------------*/

kb1-2:
procedure(cv) = {
   display(['Verifiying cost per shipment...',nl,nl]);
   R := 0;
   I := 1;
   do {
     J := 1;
      do {
        if (new_cost-I-J = asim or new_cost-I-J = avgt) {
           R := R+1;
        }
      J := J + 1;
      } while (J <= 34);
     I := I + 1;
   } while (I <= 34);
   if (R>0) {
    display(['Cost per shipment in ',R,' route(s) is modified.',nl,'Please Re-Run MIP
model']);
   }
   else {
    display(['All cost setting of routes is verified!',nl,'Please proceed to evaluate the
performance of this network.']);
   }
   return R;
}.


/* ----------------------------------*/
/* Rule Set 102 - New Cost           */
/* ----------------------------------*/

if route_existence-I-J = no
then new_cost-I-J = na.

if route_relvalency-I-J = irrelvalent
and mip_cost-I-J = MIP
and route_quantity-I-J = Q
and dbinsert(rcij(I,J,Q,MIP,MIP,'ir'))
and display(['Cost setting of route-',I,'-',J,' is not considered!',nl,nl])
then new_cost-I-J = ir.

if cost_validation-I-J = pass
and mip_cost-I-J = MIP
and route_quantity-I-J = Q
and dbinsert(rcij(I,J,Q,MIP,MIP,'rm'))
and display(['Cost setting of route-',I,'-',J,' checks OK!',nl,nl])
then new_cost-I-J = mip.

if cost_validation-I-J = fail
and modificaiton_method-I-J = abs
and mip_cost-I-J = MIP
and average_simulated_cost-I-J = ASIM
and route_quantity-I-J = Q
and dbinsert(rcij(I,J,Q,MIP,ASIM,'asim'))
and display(['Cost setting of route-',I,'-',J,' is computed as: ',ASIM,nl])
and nmvcij(I,J,C)
and dbupdate(nmvcij(I,J,ASIM))
and display(['Cost setting is updated !',nl,nl])
then new_cost-I-J = asim.

if cost_validation-I-J = fail
and modificaiton_method-I-J = avg
and mip_cost-I-J = MIP
and average_simulated_cost-I-J = ASIM
and (MIP+ASIM)/2 = AVGT
and route_quantity-I-J = Q
and dbinsert(rcij(I,J,Q,MIP,AVGT,'avgt'))
and display(['Cost setting of route-',I,'-',J,' is computed as: ',AVGT,nl])
and nmvcij(I,J,C)
and dbupdate(nmvcij(I,J,AVGT))
and display(['Cost setting is updated !',nl,nl])
then new_cost-I-J = avgt.
```

```
/* ----------------------------------*/
/* Rule Set 103 - Route Relvalency   */
/* ----------------------------------*/

if route_existence-I-J = no
then route_relvalency-I-J = irrelvalent.

if route_existence-I-J = yes
and belong_to_major_route-I-J = yes
then route_relvalency-I-J = relvalent.

if route_existence-I-J = yes
and belong_to_major_route-I-J = no
and originated_from_major_zone-I-J = yes
then route_relvalency-I-J = relvalent.

if route_existence-I-J = yes
and belong_to_major_route-I-J = no
and originated_from_major_zone-I-J = no
then route_relvalency-I-J = irrelvalent.


/*----------------------------------*/
/* Rule Set 104 cost validation     */
/*----------------------------------*/

if tolerence_level = E
and mip_cost-I-J = MVCIJ
and simulated_cost-I-J-1 = SIM1
and simulated_cost-I-J-2 = SIM2
and simulated_cost-I-J-3 = SIM3
and simulated_cost-I-J-4 = SIM4
and simulated_cost-I-J-5 = SIM5
and simulated_cost-I-J-6 = SIM6
and simulated_cost-I-J-7 = SIM7
and simulated_cost-I-J-8 = SIM8
and simulated_cost-I-J-9 = SIM9
and simulated_cost-I-J-10 = SIM10
and SIM1 >= MVCIJ * (1-E) and SIM1 <= MVCIJ * (1+E)
and SIM2 >= MVCIJ * (1-E) and SIM2 <= MVCIJ * (1+E)
and SIM3 >= MVCIJ * (1-E) and SIM3 <= MVCIJ * (1+E)
and SIM4 >= MVCIJ * (1-E) and SIM4 <= MVCIJ * (1+E)
and SIM5 >= MVCIJ * (1-E) and SIM5 <= MVCIJ * (1+E)
and SIM6 >= MVCIJ * (1-E) and SIM6 <= MVCIJ * (1+E)
and SIM7 >= MVCIJ * (1-E) and SIM7 <= MVCIJ * (1+E)
and SIM8 >= MVCIJ * (1-E) and SIM8 <= MVCIJ * (1+E)
and SIM9 >= MVCIJ * (1-E) and SIM9 <= MVCIJ * (1+E)
and SIM10 >= MVCIJ * (1-E) and SIM10 <= MVCIJ * (1+E)
then cost_validation-I-J = pass.

if tolerence_level = E
and mip_cost-I-J = MVCIJ
and simulated_cost-I-J-1 = SIM1
and simulated_cost-I-J-2 = SIM2
and simulated_cost-I-J-3 = SIM3
and simulated_cost-I-J-4 = SIM4
and simulated_cost-I-J-5 = SIM5
and simulated_cost-I-J-6 = SIM6
and simulated_cost-I-J-7 = SIM7
and simulated_cost-I-J-8 = SIM8
and simulated_cost-I-J-9 = SIM9
and simulated_cost-I-J-10 = SIM10
or SIM1 < MVCIJ * (1-E) or SIM1 > MVCIJ * (1+E)
or SIM2 < MVCIJ * (1-E) or SIM2 > MVCIJ * (1+E)
or SIM3 < MVCIJ * (1-E) or SIM3 > MVCIJ * (1+E)
or SIM4 < MVCIJ * (1-E) or SIM4 > MVCIJ * (1+E)
or SIM5 < MVCIJ * (1-E) or SIM5 > MVCIJ * (1+E)
or SIM6 < MVCIJ * (1-E) or SIM6 > MVCIJ * (1+E)
or SIM7 < MVCIJ * (1-E) or SIM7 > MVCIJ * (1+E)
or SIM8 < MVCIJ * (1-E) or SIM8 > MVCIJ * (1+E)
or SIM9 < MVCIJ * (1-E) or SIM9 > MVCIJ * (1+E)
or SIM10 < MVCIJ * (1-E) or SIM10 > MVCIJ * (1+E)
then cost_validation-I-J = fail.


/*-------------------------------------*/
/* Rule Set 104 - Modification Method */
/*-------------------------------------*/

if number_of_iteration = IT
and IT < 3
then modificaiton_method-I-J = abs.
```

```
if number_of_iteration = IT
and IT >= 3
and sign_of_convergence-I-J = yes
then modificaiton_method-I-J = abs.

if number_of_iteration = IT
and IT >= 3
and sign_of_convergence-I-J = no
then modificaiton_method-I-J = avg.



/*-----------------------------------------*/
/* Fact set - Predefined Parameters        */
/*-----------------------------------------*/

e_rule:
tolerence_level = 0.05.

mzr_rule:
major_zone_ratio = 1.

mrr_rule:
major_route_ratio = 1.



/*-----------------------*/
/* Fact set - References */
/*-----------------------*/

/* R105b - belong to major route */
if total_route_quan = TQR
and route_count = RC
and major_route_ratio = MRR
and route_quantity-I-J = RQ
and RQ >= TQR*(MRR/RC)
then belong_to_major_route-I-J = yes.

if total_route_quan = TQR
and route_count = RC
and major_route_ratio = MRR
and route_quantity-I-J = RQ
and RQ < TQR*(MRR/RC)
then belong_to_major_route-I-J = no.

procedure(total_route_quan) = {
   SUM := 0;
   forall miprst(I,J,Q,C,E,S,M) {
      SUM := SUM + Q;
   }
   return SUM;
}.

procedure(route_count) = {
   COUNT := 0;
   forall miprst(I,J,Q,C,E,S,M) {
      COUNT := COUNT + 1;
   }
   return COUNT;
}.


/* R105b - originated from major zone */
if total_zone_demand = TZD
and zone_count = ZC
and major_zone_ratio = MZR
and zone_demand-I = ZD
and ZD >= TZD*(MZR/ZC)
then originated_from_major_zone-I-J = yes.

if total_zone_demand = TZD
and zone_count = ZC
and major_zone_ratio = MZR
and zone_demand-I = ZD
and ZD < TZD*(MZR/ZC)
then originated_from_major_zone-I-J = no.

procedure(total_zone_demand) = {
   SUM := 0;
   forall di(Z,D) {
      SUM := SUM + D;
   }
   return SUM;
```

```
}.

procedure(zone_count) = {
   COUNT := 0;
   forall di(Z,D) {
      COUNT := COUNT + 1;
   }
   return COUNT;
}.


/* R107b - sign of convergence */
if previous_mip_cost-I-J = PMIP
and mip_cost-I-J = MIP
and average_simulated_cost-I-J = ASIM
and ASIM <= MIP
and ASIM >= PMIP
then sign_of_convergence-I-J = yes.

if previous_mip_cost-I-J = PMIP
and mip_cost-I-J = MIP
and average_simulated_cost-I-J = ASIM
and ASIM >= MIP
and ASIM <= PMIP
then sign_of_convergence-I-J = yes.

if previous_mip_cost-I-J = PMIP
and mip_cost-I-J = MIP
and average_simulated_cost-I-J = ASIM
then sign_of_convergence-I-J = no.


/* R102c - average_simulated_cost */
if simulated_cost-I-J-1 = SVCIJ1
and simulated_cost-I-J-2 = SVCIJ2
and simulated_cost-I-J-3 = SVCIJ3
and simulated_cost-I-J-4 = SVCIJ4
and simulated_cost-I-J-5 = SVCIJ5
and simulated_cost-I-J-6 = SVCIJ6
and simulated_cost-I-J-7 = SVCIJ7
and simulated_cost-I-J-8 = SVCIJ8
and simulated_cost-I-J-9 = SVCIJ9
and simulated_cost-I-J-10 = SVCIJ10
and (SVCIJ1+SVCIJ2+SVCIJ3+SVCIJ4+SVCIJ5+
      SVCIJ6+SVCIJ7+SVCIJ8+SVCIJ9+SVCIJ10)/10 = AVG
then average_simulated_cost-I-J = AVG.


/*-----------------------------*/
/* Fact set - Extra References */
/*-----------------------------*/

di_ref:
if di(I,D)
then zone_demand-I = D.

miprst_ref:
if miprst(I,J,Q,C,E,S,M)
then route_quantity-I-J = Q.

route_existence_ref-1:
if route_quantity-I-J is sought
and route_quantity-I-J is known
then route_existence-I-J = yes.

route_existence_ref-2:
if route_quantity-I-J is sought
and route_quantity-I-J is unknown
then route_existence-I-J = no.

sim_cost_ref:
if svcij(I,J,R,Q)
then simulated_cost-I-J-R = Q.

mip_cost_ref:
if mvcij(I,J,Q)
then mip_cost-I-J = Q.

new_mip_cost_ref:
if omvcij(I,J,Q)
then new_mip_cost-I-J = Q.

pre_mip_cost_ref:
if omvcij(I,J,Q)
```

```
then previous_mip_cost-I-J = Q.

iteration_ref:
if misc(I)
then number_of_iteration = I.



/*-------------------------------*/
/* Fact set - Database Reference */
/*-------------------------------*/

dbref_zone_demand:
database('c:\edss\db3\di.dbf',
          di(zone:integer,
             demand:integer),[]).

dbref_mip_result:
database('c:\edss\db3\miprst.dbf',
          miprst(zone:integer,
                 depot:integer,
                 quan:integer,
                 cost:real,
                 etime:integer,
                 slack:integer,
                mr:string),[]).

dbref_sim_result:
database('c:\edss\db3\svcij.dbf',
          svcij(zone:integer,
                depot:integer,
                rep:integer,cost:real),[]).

dbref_mip_cost:
database('c:\edss\db3\mvcij.dbf',
          mvcij(zone:integer,
                depot:integer,
                cost:real),[]).

dbref_updated_mip_cost:
database('c:\edss\db3\nmvcij.dbf',
          nmvcij(zone:integer,
                 depot:integer,
                 cost:real),[]).

dbref_previous_mip_cost:
database('c:\edss\db3\omvcij.dbf',
          omvcij(zone:integer,
                 depot:integer,
                 cost:real),[]).

dbref_misc:
database('c:\edss\db3\misc.dbf',
          misc(itn:integer),
          []).

dbref_rcij:
database('c:\edss\db3\rcij.dbf',
          rcij(zone:integer,
               depot:integer,
               quan:integer,
               ec:real,
               sc:real,
               res:string),[]).


nocache(route_existence-I-J).
/*nocache(new_cost-I-J).*/
nocache(route_quantity-I-J).
nocache(belong_to_major_route-I-J).
nocache(zone_demand-I).
nocache(originated_from_major_zone-I-J).
nocache(route_relvalency-I-J).
nocache(mip_cost-I-J).
nocache(simulated_cost-I-J-1).
nocache(simulated_cost-I-J-2).
nocache(simulated_cost-I-J-3).
nocache(simulated_cost-I-J-4).
nocache(simulated_cost-I-J-5).
nocache(simulated_cost-I-J-6).
nocache(simulated_cost-I-J-7).
nocache(simulated_cost-I-J-8).
nocache(simulated_cost-I-J-9).
nocache(simulated_cost-I-J-10).
nocache(cost_validation-I-J).
nocache(validation_result-I-J).
```

```
nocache(previous_mip_cost-I-J).
nocache(average_simulated_cost-I-J).
nocache(sign_of_convergence-I-J).
nocache(modificaiton_method-I-J).
```

```
/*=========================================================================*/
/*                                                                         */
/* Facility Utilization Validation Rules                                   */
/*                                                                         */
/*=========================================================================*/

goal:
initialdata = [uv].

/* ---------------------------------------*/
/* Rule Set 150 -  Utilizaiton Validation */
/* ---------------------------------------*/

kb1:
if vod = va
and voc = va
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'No over-utilization is examined',nl])
then uv = va.

kb2:
if vod = mu
and voc = mu
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'Over-utilization is examined in both Depot and Center',nl])
then uv = rr.

kb3:
if vod = mu
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'Over-utilization is examined in Depot',nl,
'Modify utilization factor and re-run MIP model',nl])
then uv = rr.

kb4:
if vod = ANYVALUE1
and voc = mu
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'Over-utilization is examined in Center',nl,
'Decrease utilization factor and re-run MIP model',nl])
then uv = rr.

kb5:
if vod = ra
or vod = va
and voc = ra
or voc = va
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'Over-utilization is examined',nl,
'Re-assign specific route and re-run simulation model',nl])
then uv = ra.


/* skip voc */
voc_ref:
voc = va.


/* ---------------------------------------*/
/* Rule Set 151 -  Depot Utilization      */
/* ---------------------------------------*/

kb6:
if aou = y
then vod = err.

aou_ref:
procedure(aou) = {
   display([nl,'Checking weekly utilziation...',nl,nl]);
   R :== n;
   J := 1;
   do {
        if (exists_u-J = y and wu-J = WUJ and WUJ > 1) {
               display(['Facility-',J,' is over-utilized : ',WUJ,nl]);
               R :== y;
        }
   J := J + 1;
   } while (J <= 34);
   return R;
}.

exists_wu_ref1:
if u-J-1 is sought
and u-J-1 is known
then exists_u-J = y.

exists_wu_ref2:
```

```
/*=========================================================================*/
/*                                                                         */
/* Facility Utilization Validation Rules                                   */
/*                                                                         */
/*=========================================================================*/

goal:
initialdata = [uv].

/* ---------------------------------------*/
/* Rule Set 150 -  Utilizaiton Validation */
/* ---------------------------------------*/

kb1:
if vod = va
and voc = va
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'No over-utilization is examined',nl])
then uv = va.

kb2:
if vod = mu
and voc = mu
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'Over-utilization is examined in both Depot and Center',nl])
then uv = rr.

kb3:
if vod = mu
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'Over-utilization is examined in Depot',nl,
'Modify utilization factor and re-run MIP model',nl])
then uv = rr.

kb4:
if vod = ANYVALUE1
and voc = mu
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'Over-utilization is examined in Center',nl,
'Decrease utilization factor and re-run MIP model',nl])
then uv = rr.

kb5:
if vod = ra
or vod = va
and voc = ra
or voc = va
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'Over-utilization is examined',nl,
'Re-assign specific route and re-run simulation model',nl])
then uv = ra.


/* skip voc */
voc_ref:
voc = va.


/* ---------------------------------------*/
/* Rule Set 151 -  Depot Utilization      */
/* ---------------------------------------*/

kb6:
if aou = y
then vod = err.

aou_ref:
procedure(aou) = {
   display([nl,'Checking weekly utilziation...',nl,nl]);
   R :== n;
   J := 1;
   do {
        if (exists_u-J = y and wu-J = WUJ and WUJ > 1) {
                display(['Facility-',J,' is over-utilized : ',WUJ,nl]);
                R :== y;
        }
   J := J + 1;
   } while (J <= 34);
   return R;
}.

exists_wu_ref1:
if u-J-1 is sought
and u-J-1 is known
then exists_u-J = y.

exists_wu_ref2:
```

```
/*=======================================================================*/
/*                                                                       */
/* Facility Utilization Validation Rules                                 */
/*                                                                       */
/*=======================================================================*/

goal:
initialdata = [uv].

/* ---------------------------------------*/
/* Rule Set 150 -  Utilizaiton Validation */
/* ---------------------------------------*/

kb1:
if vod = va
and voc = va
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'No over-utilization is examined',nl])
then uv = va.

kb2:
if vod = mu
and voc = mu
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'Over-utilization is examined in both Depot and Center',nl])
then uv = rr.

kb3:
if vod = mu
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'Over-utilization is examined in Depot',nl,
'Modify utilization factor and re-run MIP model',nl])
then uv = rr.

kb4:
if vod = ANYVALUE1
and voc = mu
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'Over-utilization is examined in Center',nl,
'Decrease utilization factor and re-run MIP model',nl])
then uv = rr.

kb5:
if vod = ra
or vod = va
and voc = ra
or voc = va
and display([nl,nl,'Facility Utilization Validation is completed',nl,
'Over-utilization is examined',nl,
'Re-assign specific route and re-run simulation model',nl])
then uv = ra.


/* skip voc */
voc_ref:
voc = va.


/* ---------------------------------------*/
/* Rule Set 151 -  Depot Utilization      */
/* ---------------------------------------*/

kb6:
if aou = y
then vod = err.

aou_ref:
procedure(aou) = {
   display([nl,'Checking weekly utilziation...',nl,nl]);
   R :== n;
   J := 1;
   do {
        if (exists_u-J = y and wu-J = WUJ and WUJ > 1) {
                display(['Facility-',J,' is over-utilized : ',WUJ,nl]);
                R :== y;
        }
   J := J + 1;
   } while (J <= 34);
   return R;
}.

exists_wu_ref1:
if u-J-1 is sought
and u-J-1 is known
then exists_u-J = y.

exists_wu_ref2:
```

```
if u-J-1 is sought
and u-J-1 is unknown
then exists_u-J = n.

u_ref:
if dp(DAY,J,Q,U)
then u-J-DAY = U.


procedure(wu-J) = {
   SUM := 0;
   COUNT := 0;
   forall dp(DAY,J,Q,U) {
      SUM := SUM + U;
      COUNT := COUNT +1;
   }
   AVG := SUM/COUNT;
   P := fix(AVG*100);
   display(['Weekly Utilization for Depot ',J,' : ',P,' %',nl]);
   return AVG;
}.


kb7:
if aou = n
and ousd = n
then vod = va.


ousd_ref:
procedure(ousd) = {
   display([nl,'Checking daily utilziation...',nl,nl]);
   R :== n;
   D := 1;
   do {
    J := 1;
    do {
        if (exists_u-J = y and u-J-D = UJD and UJD > 1 and UJD*100  =P) {
               display(['Facility ',J,' in day ',D,' is over-utilized : ',P,'%',nl]);
               R :== y;
        }
    J := J + 1;
    } while (J <= 34);
   D := D+1;
   } while (D <=6);
   if (R == n) {
     display(['Daily utilization verified!',nl,nl]);
   }
   return R;
}.


kb8:
if aou = n
and ousd = y
and oumf = y
then vod = mu.

kb9:
if aou = n
and ousd = y
and oumf = n
and rrp = n
then vod = mu.

kb10:
if aou = n
and ousd = y
and oumf = n
and rrp = y
then vod = ra.

/* ---------------------------------------------------*/
/* Rule Set 152-4 - Over Utilizaiton in Specific Day  */
/* ---------------------------------------------------*/

kb11-16:
procedure(oumf) = {
   display([nl,'Checking network tightness in specific day...',nl]);
   R :== n;
   FCOUNT := total_depot_count;
   D := 1;
   do {
    J := 1;
    OCOUNT := 0;
    do {
        if (exists_u-J = y and u-J-D = UJD and UJD > 1) {
```

```
              OCOUNT := OCOUNT + 1;
          }
     J := J + 1;
     } while (J <= 34);
        if (OCOUNT / FCOUNT = OR and OR > 0.75) {
        display(['Network in day ',D,' is very tight ',OR,'% of facilities is over-
utilized',nl]);
        R :== y;
        }
   D := D+1;
   } while (D <=6);
   if (R == n) {
     display(['No general over-utilization is examined in specfic days!',nl]);
   }
   return R;
}.

total_depot_count_ref:
procedure(total_depot_count) = {
   COUNT := 0;
   forall dp(1,J,Q,U) {
      COUNT := COUNT +1;
   }
   return COUNT;
}.


/* -----------------------------------------------------*/
/* Rule Set 155-6 -  Rotue Re-assignment Possibility    */
/* -----------------------------------------------------*/

kb17-21:
procedure(rrp) = {
   display([nl,'Checking possibility for route re-assignment...',nl,nl]);
   R :== y;
   D := 1;
   do {
    J := 1;
    do {
        if (exists_u-J = y and u-J-D = UJD and UJD > 1) {
             display([nl,'Inspecting alternate route for Facility ',J,' in day ',D,'...',nl]);
              if (poss-J-D = n) {
                display(['No alternate route found !!!',nl]);
                R :== n;
              }
        }
    J := J + 1;
    } while (J <= 34);
   D := D+1;
   } while (D <=6);
   return R;
}.

/* -----------------------------------------------------*/
/* Rule Set 157 -  RR Check for Specific Facility       */
/* -----------------------------------------------------*/

kb22:
if u-N-D = U
and U > 1.2
then poss-N-D = n.

kb23:
if u-N-D = U
and U < 1.2
and altr-N-D = y
then poss-N-D = y.

kb24:
if u-N-D = U
and U < 1.2
and altr-N-D = n
then poss-N-D = n.

altr_ref:
procedure(altr-N-D) = {
   R :== n;
   J := 1;
   do {
      if (depot_check-J-N-D = y) {
    I := 1;
    do {
        if (zone_check-I-J-N-D = y) {
               display(['Zone ',I,' to Depot ',J,nl]);
               R :== y;
        }
    I := I + 1;
```

```
    } while (I <= 34);
    }
   J := J+1;
   } while (J <=34);
   return R;
}.

depot_check_ref1:
if exists_u-J = n
then depot_check-J-N-D = n.

depot_check_ref2:
if exists_u-J = y
and u-J-D = UJ
and u-N-D = UN
and UJ + UN = TT
and TT >= 2
then depot_check-J-N-D = n.

depot_check_ref3:
if exists_u-J = y
and u-J-D = UJ
and u-N-D = UN
and UJ + UN = TT
and TT < 2
then depot_check-J-N-D = y.


zone_check_ref1:
if exists_r-I-J = n
then zone_check-I-J-N-D = n.

zone_check_ref2:
if exists_r-I-J = y
and quan-I-J = Q
and ovu-N = O
and Q < O
then zone_check-I-J-N-D = n.

zone_check_ref3:
if exists_r-I-J = y
and quan-I-J = Q
and ovu-N = O
and Q >= O
and froute-I-J = n
then zone_check-I-J-N-D = n.

zone_check_ref4:
if exists_r-I-J = y
and quan-I-J = Q
and ovu-N = O
and Q >= O
and froute-I-J = y
then zone_check-I-J-N-D = y.


exists_r_ref1:
if quan-I-J is sought
and quan-I-J is known
then exists_r-I-J = y.

exists_r_ref2:
if quan-I-J is sought
and quan-I-J is unknown
then exists_r-I-J = n.

quan_ref:
if miprst(I,J,Q,C,E,S,M)
and Q//260 = QIJ
then quan-I-J = QIJ.

ovu_ref:
if dp(DAY,J,Q,U)
and (1-U) * Q = O
then ovu-N = O.

froute_ref1:
if tij-I-J = TIJ
and mind-J = MIND
and cutoff = CO
and MIND - CO < TIJ
then froute-I-J = n.

froute_ref2:
if tij-I-J = TIJ
and mind-J = MIND
and cutoff = CO
```

```
and MIND - CO >= TIJ
then froute-I-J = y.

tij_ref:
if tij(I,J,A,S)
then tij-I-J = A.

cutoff_ref:
if simpara(I,C,W,O)
then cutoff = O.

mind-J_ref:
if dptime(D,T)
then mind-J = T.


/*--------------------*/
/* Database Reference */
/*--------------------*/

dbref_dp:
database('c:\edss\db3\dp.dbf',
         dp(day:integer,
            dep:integer,
            tq:integer,
            util:integer),[]).

dbref_mip_result:
database('c:\edss\db3\miprst.dbf',
          miprst(zone:integer,
                 depot:integer,
                 quan:integer,
                 cost:real,
                 etime:integer,
                 slack:integer,
                mr:string),[]).

dbref_tij:
database('c:\edss\db3\tij.dbf',
        tij(zone:integer,
            depot:integer,
            avg:integer,
           std:real),[]).

dbref_simpara:
database('c:\edss\db3\simpara.dbf',
         simpara(itn:integer,
                 cr:real,
                 wr:real,
                 cutoff:integer),[]).

dbref_demandzone_log:
database('c:\edss\db3\zlog.dbf',
            zlog(rep:integer,
                 day:integer,
                 zone:integer,
                 depot:integer,
                 ztime:integer,
                 dtime:integer,
                 ctime:integer,
                 cutil:real,
                mr:string),[]).

dbref_dptime:
database('c:\edss\db3\dptime.dbf',
         dptime(depot:integer,
                mindt:integer),[]).
```

```
/*========================================================================*/
/*                                                                        */
/* Performance Evaluation Rules                                           */
/*                                                                        */
/*========================================================================*/

initialdata = [daysr].


/*-------------------------------------*/
/* Rule set 201 - Concluded Action     */
/*-------------------------------------*/

kb1:
if sc = u
and sco = SCO
and display(['Network Performance Evaluation is completed',nl,nl,
'Modification focus on Improving Service Coverage',nl,
'Suggest delay operational cut-off and re-run simulation',nl,
'Suggested new cut-off : ',SCO,nl])
then pe = sco.

kb2:
if sc = s
and sr = u
and display(['Network Performance Evaluation is completed',nl,nl,
'Modification focus on Improving Service Reliablity',nl,
'Suggest redesign network configuration with next binding time window',nl,nl])
and ptw
then pe = rn.

ptw_ref:
procedure(ptw) = {
   TMAX := tmax;
   COUNT :=0;
   CON :=0;
   while (CON = 0) {
   COUNT := COUNT + 1;
      forall miprst(I,J,Q,C,E,S<COUNT,ALL) {
      if (S >0) {
      CON := CON+S;
      T := TMAX-S;
      display(['Next Binding Route: Zone-',I,' to Depot-',J,nl,
               ' (Route Slack Time : ',S,' Minutes)',nl,nl,
               'Suggested new Tmax parameter : ',T,nl]);
      return S;
      }
      }
   }
}.

tmax_ref:
if mippara(ITN,TMAX,EF)
then tmax = TMAX.


kb3:
if sc = s
and sr = s
and daysr = u
and display(['Network Performance Evaluation is completed',nl,nl,
'Modification focus on Improving Service Reliablity',nl,
'Suggest increase workforce and re-run simulation',nl,
'Recommened workforce increment zones are as follow:',nl,nl])
and wfl
then pe = swf.

kb4:
if sc = s
and sr = s
and daysr = n
and fi = n
and display(['Network Performance Evaluation is completed',nl,
'No modification is recommended',nl])
then pe = nm.

kb5:
if sc = s
and sr = s
and daysr = n
and fi = y
and sco = SCO
and display(['Network Performance Evaluation is completed',nl,nl,
'Room for Improvement Found',nl,
'Suggest delay operational cut-off and re-run simulation',nl,nl,
'Suggested new cut-off : ',SCO,nl])
then pe = fi.
```

```
/*-------------------------------------*/
/* Rule set 202 - Service Coverage    */
/*-------------------------------------*/

kb6:
if scm = n
then sc = u.

kb7:
if scm = y
and scs = n
then sc = u.

kb8:
if scm = y
and scs = y
then sc = s.

scm_ref1:
if asc = V1
and reqsc = V2
and V1 <= V2
then scm = n.

scm_ref2:
if asc = V1
and reqsc = V2
and V1 > V2
then scm = y.

asc_ref:
procedure(asc) = {
   SUM := 0;
   COUNT := 0;
   forall np(R,T,U,L) {
      SC := 1-L/T;
      SUM := SUM + SC;
      COUNT := COUNT +1;
   }
   AVG := SUM/COUNT;
   P := fix(AVG*100);
   display([nl,'Simulated Service Coverage is computed: ',P,' %',nl]);
   return AVG;
}.

reqsc_ref:
reqsc = 0.85.

que_scs_ref:
question(scs) =
'Are you satisified with the Service Coverage (Y)es/(N)o?'.
legalvals(scs) = [y,n].



/*-------------------------------------*/
/* Rule set 203 - Service Reliability */
/*-------------------------------------*/

kb9:
if srm = n
then sr = u.

kb10:
if srm = y
and srs = n
then sr = u.

kb11:
if srm = y
and srs = y
then sr = s.

srm_ref1:
if asr = V1
and reqsr = V2
and V1 <= V2
then srm = n.

srm_ref2:
if asr = V1
and reqsr = V2
and V1 > V2
then srm = y.
```

```
asr_ref:
procedure(asr) = {
   SUM := 0;
   COUNT := 0;
   forall np(R,T,U,L) {
      SR := 1-U/(T-L);
      SUM := SUM + SR;
      COUNT := COUNT +1;
   }
   AVG := SUM/COUNT;
   P := fix(AVG*100);
   display(['Simulated Service Reliability is computed: ',P,' %',nl]);
   return AVG;
}.

que_srs_ref:
question(srs) =
'Are you satisified with the Service Reliability (Y)es/(N)o?'.
legalvals(srs) = [y,n].

reqsr_ref:
reqsr = 0.95.


/*--------------------------------------*/
/* Rule set 204 - Weekdays Reliability */
/*--------------------------------------*/

procedure(daysr) = {
   display([nl,'Checking specific reliability for each weekdays and zone...',nl,nl]);
   R :== s;
   D := 1;
   REQSR := reqsr;
   do {
    I := 1;
    do {
         if (dsr-I-D = DSR and DSR < REQSR) {
              display(['Reliability of Zone ',I,' in day ',D,' is unsatisfying ', DSR, nl]);
               R :== u;
         }
    I := I + 1;
    } while (I <= 34);
   D := D+1;
   } while (D <=6);
   if (R == s) {
    display(['No unsatisfying reliability is examined in specific weekdays and zones.',nl]);
   }
   else {
    display(['Unsatisfying reliability is examined, now seeking for retification.',nl]);
   }
   return R;
 }.


 dsr-I-D_ref1:
 if ud(WEEK,DAY,K,D,I,J,TQ,LIS,LIC)
 and (TQ-LIC-LIS)/(TQ-LIC) = DSR
 then dsr-I-D = DSR.


 /*--------------------------------------*/
 /* Rule set 205 - Room for Improvement */
 /*--------------------------------------*/

 kb16:
 if ss = y
 then ri = y.

 kb17:
 if ss = n
 and ds = y
 then ri = y.

 kb18:
 if ss = n
 and ds = n
 then ri = n.

 ss_ref1:
 if sct = T
 and T > 0
 then ss = y.

 ss_ref2:
 if sct = T
 and T <= 0
```

```
then ss = n.

sct_ref:
procedure(sct) = {
   SUM := 0;
   COUNT := 0;
   forall slog(R,5,S,F,L,U) {
      ST := L-F;
      SUM := SUM + ST;
      COUNT := COUNT +1;
   }
   RST := SUM//COUNT;
   return RST;
}.

ds_ref1:
if dst = T
and T > 0
then ds = y.

ds_ref2:
if dst = T
and T <= 0
then ds = n.

dst_ref:
if mzct = MZCT
and slt = SLT
and SLT - MZCT = D
then dst = D.

mzct_ref:
procedure(mzct) = {
   LC := 0;
   forall zlog(R,5,I,J,Z,D,C,U,Yes) {
      if (C > LC) {
           LC := C;
      }
   }
   HR_RST := LC//60;
   MIN_RST := LC mod 60;
   display(['Latest Shipment Arrival at Srv Ctr (For Major Zones) @: ',HR_RST, ':',
MIN_RST,nl]);
   return LC;
}.


slt_ref:
procedure(slt) = {
   LL := 0;
   forall slog(R,5,S,F,L,U) {
      if (L > LL) {
           LL := L;
      }
   }
   HR_RST := LL//60;
   MIN_RST := LL mod 60;
   display(['Latest Srv Ctr Truck Cutoff Time @: ',HR_RST, ':', MIN_RST,nl]);
   return LL;
}.


/*------------------------------------------*/
/* Rule set 207 - Suggest Cut-off           */
/*------------------------------------------*/

kb19:
if not mn = im
and tc = TC
then sco = TC.

kb20:
if mn = im
and rc = RC
then sco = RC.


/*------------------------------------------*/
/* Rule set 208 - Target Cutoff             */
/*------------------------------------------*/

kb21:
if tcov = na
then tc = na.

kb22-31:
if tcov = DSC and esc-1655 = SSC and DSC < SSC then tc = 1655.
```

```
if tcov = DSC and esc-1705 = SSC and DSC < SSC then tc = 1705.
if tcov = DSC and esc-1715 = SSC and DSC < SSC then tc = 1715.
if tcov = DSC and esc-1725 = SSC and DSC < SSC then tc = 1725.
if tcov = DSC and esc-1735 = SSC and DSC < SSC then tc = 1735.
if tcov = DSC and esc-1745 = SSC and DSC < SSC then tc = 1745.
if tcov = DSC and esc-1755 = SSC and DSC < SSC then tc = 1755.
if tcov = DSC and esc-1805 = SSC and DSC < SSC then tc = 1805.
if tcov = DSC and esc-1815 = SSC and DSC < SSC then tc = 1815.
if tcov = DSC and esc-1825 = SSC and DSC < SSC then tc = 1825.
if tcov = ALLOTHERS then tc = unknown.

esc_ref:
if sst(CO,SC,SR)
then esc-CO = SC.


/*-----------------------------------------*/
/* Rule set 209 - Target Coverage          */
/*-----------------------------------------*/

kb32:
if scs = n
and desired_cov = DC
then tcov = DC.

question(desired_cov) =
'What is your desired service coverage?'.


kb33:
if scm = n
and reqsc = RC
then tcov = RC.

kb34:
if scm = y
and scs = y
then tcov = na.


/*----------------------------------------------*/
/* Rule set 210 -  Recommended Cut-off          */
/*----------------------------------------------*/

kb35-44:
if asr = ASR and esr-1825 = SSR and ASR < SSR + 0.015 then rc = 1825.
if asr = ASR and esr-1815 = SSR and ASR < SSR + 0.015 then rc = 1815.
if asr = ASR and esr-1805 = SSR and ASR < SSR + 0.015 then rc = 1805.
if asr = ASR and esr-1755 = SSR and ASR < SSR + 0.015 then rc = 1755.
if asr = ASR and esr-1745 = SSR and ASR < SSR + 0.015 then rc = 1745.
if asr = ASR and esr-1735 = SSR and ASR < SSR + 0.015 then rc = 1735.
if asr = ASR and esr-1725 = SSR and ASR < SSR + 0.015 then rc = 1725.
if asr = ASR and esr-1715 = SSR and ASR < SSR + 0.015 then rc = 1715.
if asr = ASR and esr-1705 = SSR and ASR < SSR + 0.015 then rc = 1705.
if asr = ASR and esr-1655 = SSR and ASR < SSR + 0.015 then rc = 1655.
if asr = ALLOTHERS then dsst = unknown.

esr_ref:
if sst(CO,SC,SR)
then esr-CO = SR.


/*----------------------------------------------*/
/* Rule set 211 - Workforce Increment List      */
/*----------------------------------------------*/


multivalued(wfl).

kb45:
if aca = n
then wfl = na.

kb46-78:
if acaz-1 = h and display(['zone-1',nl]) then wfl = zone1.
if acaz-2 = h and display(['zone-2',nl]) then wfl = zone2.
if acaz-3 = h and display(['zone-3',nl]) then wfl = zone3.
if acaz-4 = h and display(['zone-4',nl]) then wfl = zone4.
if acaz-5 = h and display(['zone-5',nl]) then wfl = zone5.
if acaz-6 = h and display(['zone-6',nl]) then wfl = zone6.
if acaz-7 = h and display(['zone-7',nl]) then wfl = zone7.
if acaz-8 = h and display(['zone-8',nl]) then wfl = zone8.
if acaz-9 = h and display(['zone-9',nl]) then wfl = zone9.
if acaz-10 = h and display(['zone-10',nl]) then wfl = zone10.
if acaz-11 = h and display(['zone-11',nl]) then wfl = zone11.
if acaz-12 = h and display(['zone-12',nl]) then wfl = zone12.
if acaz-13 = h and display(['zone-13',nl]) then wfl = zone13.
```

```
if acaz-14 = h and display(['zone-14',nl]) then wfl = zone14.
if acaz-15 = h and display(['zone-15',nl]) then wfl = zone15.
if acaz-16 = h and display(['zone-16',nl]) then wfl = zone16.
if acaz-17 = h and display(['zone-17',nl]) then wfl = zone17.
if acaz-18 = h and display(['zone-18',nl]) then wfl = zone18.
if acaz-19 = h and display(['zone-19',nl]) then wfl = zone19.
if acaz-20 = h and display(['zone-20',nl]) then wfl = zone20.
if acaz-21 = h and display(['zone-21',nl]) then wfl = zone21.
if acaz-22 = h and display(['zone-22',nl]) then wfl = zone22.
if acaz-23 = h and display(['zone-23',nl]) then wfl = zone23.
if acaz-24 = h and display(['zone-24',nl]) then wfl = zone24.
if acaz-25 = h and display(['zone-25',nl]) then wfl = zone25.
if acaz-26 = h and display(['zone-26',nl]) then wfl = zone26.
if acaz-27 = h and display(['zone-27',nl]) then wfl = zone27.
if acaz-28 = h and display(['zone-28',nl]) then wfl = zone28.
if acaz-29 = h and display(['zone-29',nl]) then wfl = zone29.
if acaz-30 = h and display(['zone-30',nl]) then wfl = zone30.
if acaz-31 = h and display(['zone-31',nl]) then wfl = zone31.
if acaz-32 = h and display(['zone-32',nl]) then wfl = zone32.
if acaz-33 = h and display(['zone-33',nl]) then wfl = zone33.
if acaz-34 = h and display(['zone-34',nl]) then wfl = zone34.


/*-------------------------------------------------------*/
/* Rule set 212 - After Cut-off Pickup Activities         */
/*-------------------------------------------------------*/

/*---------------------------------------------------------*/
/* Rule set 212 -  After Cut-off Pickup Activities in Zone-N */
/*---------------------------------------------------------*/

kb36:
if pd-N = T
and T >= 60
then acaz-N = h.

kb37:
if pd-N = T
and T < 60
then acaz-N = l.

acaz-N_ref:
nocache(acaz-N).

pd_ref1:
procedure(pd-I) = {
   SUM := 0;
   COUNT := 0;
   CUTOFF := co;
   SLT := slt;
   forall zlog(R,5,I,J,Z,D,C,U,ALL) {
      SUM := SUM + Z - CUTOFF;
      COUNT := COUNT +1;
   }
   if (COUNT = 0) {
   return 0;
   }
   else {
   RST := SUM//COUNT;
   return RST;
   }
}.

pd_ref2:
nocache(pd-I).

co_ref:
if simpara(I,C,W,O)
then co = O.


/*--------------------*/
/* Database Reference */
/*--------------------*/

dbref_pickup_activites:
database('c:\edss\db3\np.dbf',
          np(rep:integer,
             tp:integer,
             ud:integer,
             lic:integer),[]).

dbref_demandzone_log:
database('c:\edss\db3\zlog.dbf',
           zlog(rep:integer,
                day:integer,
                zone:integer,
```

```
                depot:integer,
                ztime:integer,
                dtime:integer,
                ctime:integer,
                cutil:real,
               mr:string),[]).

dbref_network_configurations:
database('c:\edss\db3\miprst.dbf',
          miprst(zone:integer,
                depot:integer,
                quan:integer,
                cost:real,
                etime:integer,
                slack:integer,
               mr:string),[]).

dbref_simpara:
database('c:\edss\db3\simpara.dbf',
         simpara(itn:integer,
                cr:real,
                wr:real,
                cutoff:integer),[]).

dbref_sst:
database('c:\edss\db3\sst.dbf',
         sst(co:integer,
             sc:real,
             sr:real),[]).

dbref_ud:
database('c:\edss\db3\ud.dbf',
         ud(week:integer,
            day:integer,
            service:integer,
            dest:integer,
            zone:integer,
            depot:integer,
            total:integer,
            lis:integer,
            lic:integer),[]).

dbref_srvctr_log:
database('c:\edss\db3\slog.dbf',
            slog(rep:integer,
                 day:integer,
                 stime:integer,
                 ftime:integer,
                 ltime:integer,
                 util:real),[]).

dbref_mippara:
database('c:\edss\db3\mippara.dbf',
         mippara(itn:integer,
                 tmax:integer,
                 ef:real),[]).
```