A Knowledge-Based Environment for the Validation of Simulation Models

by

F. Nur Özmızrak

THESIS
SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES AND RESEARCH OF
THE UNIVERSITY OF OTTAWA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
PH. D. DEGREE IN COMPUTER SCIENCE

The Ph. D. program in Computer Science
is jointly administrated with Carleton University by
the Ottawa-Carleton Institute for Computer Science.

ISBN 0-315-95943-6

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Abstract

A new approach to the validation of simulation models is developed in this thesis. The underlying goal of the approach is to utilize both domain specific knowledge about expected behaviour of the system (typically acquired from a domain expert), and where available, data gathered from an observable system. The approach is knowledge, rather than statistically, based.

Behavioural knowledge about the simulation model is formulated in terms of an external specification which is comprised of three disjoint sets of relationships; namely, formal specifications, qualitative specifications and observational specifications. These sets of relationships are developed in a formal context and together they form a knowledge base of validation reference information.

A key element in the implementation of a software environment for simulation model validation is the development of an effective means for utilizing the validation reference information. This, in effect, becomes an experiment design problem. Its solution is undertaken in a constraint set framework that evolves from a transformation on the validation reference information. The experiment design problem is thus transformed into a "constraint covering" problem which we refer to as the C^3 problem (Consistent Constraint Covering).

The complexity of the C^3 problem is analyzed using graph-theoretic concepts. This analysis is based on a new concept that is associated with a set of constraints; namely, a constraint set graph. We explore the relationships between a maximum independent set of a constraint set graph and a simplified version of the C^3 problem and show its relationship to both the edge cover problem and a set-covering problem. A relationship between the C^3 problem and the clique cover problem, which is known to be NP-complete, is also shown. An effective approximation algorithm to solve the C^3 problem is then developed and some of its key properties are explicitly demonstrated. The performance of the algorithm on a set of example problems is included.

The thesis establishes a formal general framework for studying the validation problem for simulation models. The C^3 problem is shown to be a key aspect of the validation problem and the solution to the C^3 problem which is developed therefore contributes significantly to the solution of the simulation model validation problem.

Acknowledgement

I am deeply grateful to my supervisor Professor Louis G. Birta for his guidance, unwavering support, constant encouragement and kindness. I am also thankful to him for giving me an opportunity to do my graduate studies under his supervision.

I am especially grateful to Professor Tuncer I. Ören for his kind advice and invaluable support throughout my graduate studies.

I am indebted to Professor John Chinneck and Professor Jeffrey Sidney who spent considerable amount of their precious time, discussing several points related to this thesis.

I wish to thank Professor Art Warburton whose comments and suggestions were extremely helpful. I also thank Professor Sylvia Boyd for her useful suggestions.

I would also like to express my thanks to the examiners of this thesis (Professor John Chinneck, Professor Jeffrey Sidney, Professor Tuncer I. Ören and Professor Brian Unger) for their constructive suggestions for improving the quality of the presentation in this thesis.

My financial support during the course of my graduate studies from both OGS and NSERC (through my supervisor's grant) is gratefully acknowledged. I also thank the Computer Science department for providing the necessary facilities through the course of my graduate study.

I thank my colleague Dr. G. H. Masapati for his constant encouragement during my graduate studies. His friendship is greatly valued.

I would like to express my deepest gratitude to my dearest parents İhsan and Jale, for their constant support and for their kind help in taking care of my wonderful daughter Pınar, with whom I plan to spend much more time. I also thank my brother Yavuz for his encouragement.

I am immeasureably grateful to my husband Murat for his incredible support, tolerance and help without which this work would not have been possible.

Finally, I know my cat PonPon will be relieved to see that he does not have to stay awake late watching me study.

Contents

Chapter 1

Introduction

The development and use of simulation models has become an integral part of contemporary problem solving strategies. A simulation study is often the only scientific approach for the evaluation of decision alternatives or for gaining insight into the dynamic behaviour of complex systems.

As expressed in Birta et.al.[9], a simulation study typically focuses on some aspect of reality, either as it exists now, may exist in the future or perhaps as it existed in the past. The particular portion of reality that is of interest is referred to as the system under study. A main ingredient in a simulation study is a conceptual model of the system under study. This model captures, in some appropriate format, the essential behaviour generating aspects of the system under study. A simulation model expresses a conceptual model in a computer processable form; i.e., a computer program. This program provides the basis for the generation of model behaviour.

Computer simulation is the activity of using the simulation model to carry out goal-directed experiments. The results and/or insights obtained from these experiments provide the basis for the problem solving activity. Unless the simulation model

is validated, the information derived from it can be challenged since it is, at best, simply a correctly executing computer program. Validation of a simulation model is the problem of establishing whether or not the behaviour of the simulation model is consistent with some set of specifications. We refer to it as the simulation model validation problem or simply the validation problem.

The goal of this research project is to develop a knowledge-based approach to the validation problem. The approach is based on a formal framework which provides a context within which the validation problem can be formally studied. Our approach both complements and for the most part, subsumes, existing simulation model validation methods.

1.1 Modeling, Simulation and Validation

We establish a basis for our discussion of the validation problem by first examining some of the important aspects of the modeling and simulation activity. An overview is provided in Figure 1.1. A *system*, within the context of Figure 1.1, is some part of the real world which is of interest. Generally, a system is studied in order to acquire insights into relevant aspects of its behaviour for purposes of decision making, design or analysis. What constitutes the "relevant aspects" depends entirely on the goals that have been prescribed for the study. However it may not be possible to investigate a system's behaviour simply by experimenting with it because the system itself may not exist, or it may be infeasible (e.g., too costly) to deal with the system directly. A cost effective alternative to the direct study of a system's behaviour is computer modeling and simulation.

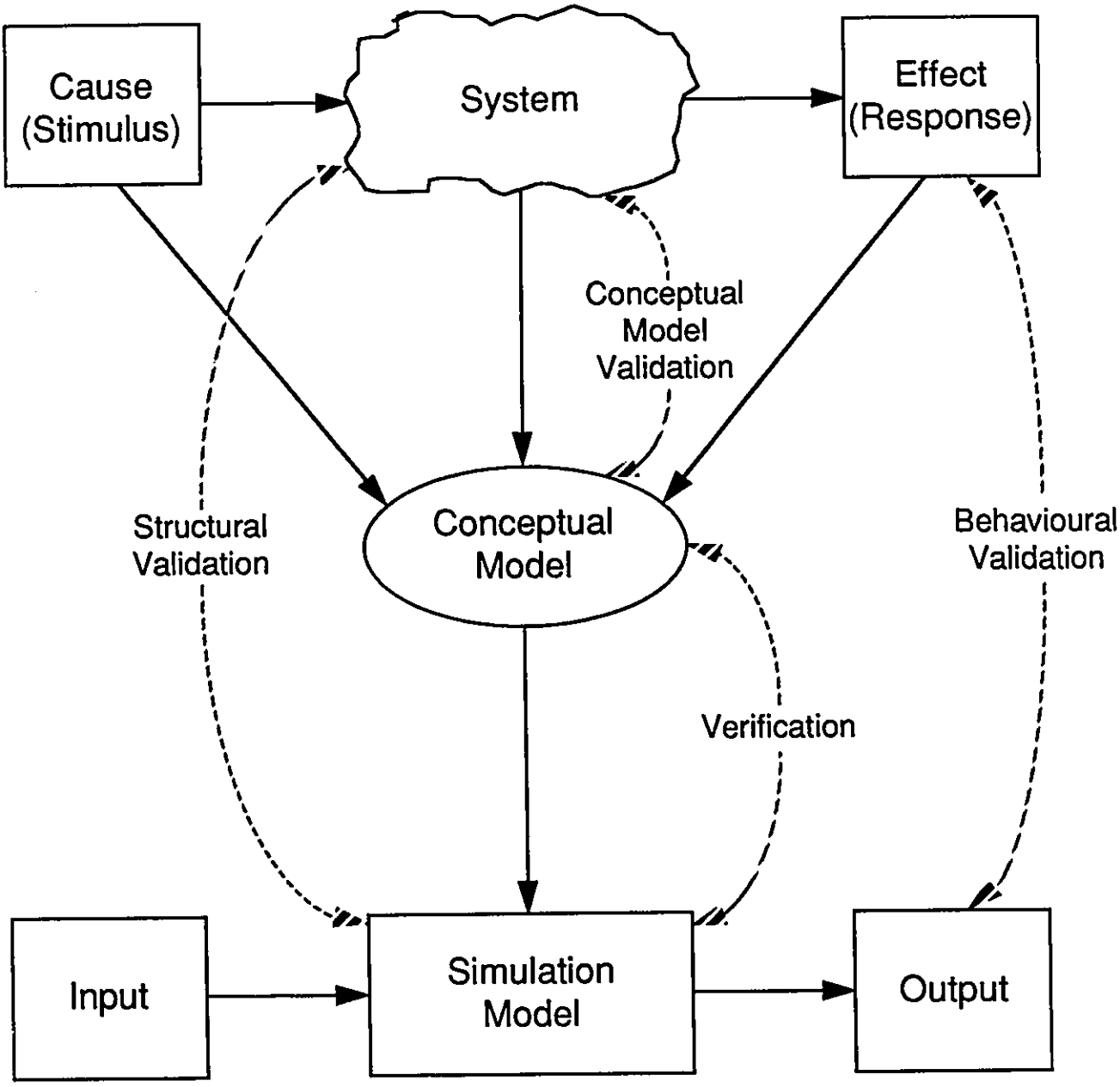

Figure 1.1. Conceptualization of the Modeling and Simulation Activity & Validation/Verification

Modeling and *Simulation* is a collection of activities which includes:

- designing a *conceptual model* which is a formal description of a system being studied (it may have a natural language, pictorial/graphical, mathematical and/or logic format),
- designing a *computer simulation model* or simply *simulation model* which is a computer program representation of the conceptual model. In a sense, the simulation model is an input-output transformation device,
- experimentation with the simulation model under prescribed conditions; i.e., *simulation*, and
- analysis and evaluation of the results obtained from the simulation experiments.

Both in the context of a system and its model, *inputs* are defined as those entities whose values are determined externally to the system (model). On the other hand, *outputs* are those entities whose values characterize the way in which the system (model) responds to its inputs. In this respect, the inputs can be considered as *causes* and the outputs as *effects*.

The set of all possible pairs of input-output values that can be produced by a system (over some observation period) is generally referred to as input-output (I-O) behaviour of the system. However it is generally not feasible to observe all possible pairs and therefore some "interesting" subset of these pairs must suffice to represent the system behaviour. These same remarks apply to simulation model behaviour. Typically, each simulation experiment (i.e., execution of the simulation model) is carried out to explore some particular behavioural aspect of the system.

Since a simulation model is an abstraction of the system under study, it is necessary to have some assurance that inferences drawn from it can be accepted with confidence. The establishment of this confidence is associated with two distinct activities; namely, verification and validation. *Verification* is concerned with the correctness of the transformation from the intermediate abstract representation (the conceptual model) to the program code (the simulation model); i.e., with ensuring that the program code faithfully reflects the behaviour that is implicit in the specifications of the conceptual model. *Validation*, on the other hand, is the process of determining whether the model, conceptual or simulation, is an "adequate" representation of the system. Three kinds of validation can be identified: conceptual model validation, structural validation and behavioural validation. *Conceptual model validation* is the evaluation of a conceptual model with respect to the system, where the objective is primarily to evaluate the realism of the conceptual model with respect to the goaals of the study. *Structural validation* is the evaluation of the structure of a simulation model with respect to perceived structure of the system. *Behavioural validation* is the evaluation of the simulation model behaviour and the system behaviour observed or generated under the same conditions. A comprehensive evaluation of validation in simulation studies can be found in Õren et. al. [86].

Our focus in this study is on behavioural validation. We regard this activity as being concerned with ensuring that the behaviour exhibited by the simulation model is consistent with expected and/or measured behaviour of the system. It, can therefore be undertaken only if "something" is known about the system since the known information provides the reference against which the simulation model behaviour is compared. There is no general "proof" procedure; it is fundamentally

an experimental process. In the sequel we generally refer to behavioural validation simply as validation.

Generally, one cannot establish the "absolute" validity of a model; consequently the goal in the validation process is basically to gain a reasonable level of confidence so that inferences drawn from the model can be accepted with confidence; i.e., they are applicable to the real phenomenon that is being studied [47, 102, 109].

It has been observed [5] that a standard terminology for simulation validation does not yet exist (many terms are used interchangeably). This contributes to the challenge presented by the validation problem. The stochastic nature of the behaviour of the system further contributes to the difficulty of the problem.

The validation problem has traditionally been approached in an ad hoc manner with tools and techniques that are often very specific to the model that is being studied. The formal techniques that do exist typically have a statistical orientation and they offer very little opportunity for the incorporation of domain specific knowledge. Furthermore the acqusition of "valid" data to carry out these statistical techniques can be a substantial problem. Although adequate for special cases, such approaches do not provide a satisfactory basis for automating the validation process.

In this thesis, a knowledge-based approach to the simulation model validation problem is developed. The field of knowledge-based systems is a branch of artificial intelligence. "Three fundamental concepts of knowledge-based systems distinguish them from conventional algorithmic programs and from general search-based programs: 1) The separation of the knowledge from how it is used. 2) The use of highly specific domain knowledge. 3) The heuristic rather than algorithmic nature of the knowledge employed." (Gonzales and Dankel [44, p.22]). Most commonly used

knowledge-based systems are rule-based systems.

Our approach allows for the incorporation of both domain specific knowledge provided by an expert which describes the intended behaviour of the system and data acquired from an alternate observable system; i.e., the real (target) system if it exists and/or another existing (and validated) simulation model.

1.2 Review of Validation Methods

In this section we review existing validation techniques that have appeared in the literature [3, 4, 6, 7, 36, 54, 75, 90, 103, 104, 106, 108, 109, 114]. A tutorial on verification and validation of simulation models, which gives a general introduction and defines various validation techniques, can be found in [103, 104, 106]. A recent bibliography can be found in [120].

Basically, simulation model validation is the process of comparing some subset of the I-O behaviour of a simulation model with a corresponding subset of the I-O behaviour of the system being represented. Two general approaches for the comparison of these behaviours are an objective approach and a subjective approach.

In the objective approach, this comparison is usually done using statistical tests and procedures; whereas in the subjective approach, graphical displays, intuition, opinions or past experience about how the system of interest operates, are employed. The objective approach is most suitable when the system is completely observable whereas the subjective approach is applicable when the system does not yet exist, or is completely or partially unobservable.

There are many statistical test techniques which are applicable under specific

conditions. A bibliography of most of these techniques can be found in [3, 5] and with ongoing research new techniques with enhanced features will likely be developed [8]. However the selection of an appropriate test technique itself can be a considerable problem. This has led to new research areas such as advisory expert systems for the selection of suitable test techniques; examples can be found in [27, 93].

An interesting discussion of simulation validation from the point of view of Popper's falsificationism can be found in [47, p.65]. This view is based on a view of science which "holds that a theory is scientific only if it is falsifiable; that is if it makes specific predictions whose nonoccurance will discredit it".

A summary of some subjective validation methods is given below:

Historical methods are quite philosophical in nature and classified as rationalism, empiricism and pragmatism. The underlying approach in both rationalism and empiricism is to formulate hypotheses devised to fit the observed facts, and then apply the rules of formal logic to deduce various consequences. With rationalism all the basic assumptions and premises may remain open to an empirical verification. Empiricism on the other hand refuses to admit any assumptions and premises that cannot be verified independently by experiment or analysis of empirical data. Pragmatism views the simulation model as a black box which transforms the inputs into a set of outputs, and looks only at the input-output relationships.

Multistage validation combines the three historical methods into a three-stage approach as suggested in [75]. The first stage is a modified rationalist approach based upon previous research, relevant theory and observations; the second stage consists of empirical testing of the internal structure of the model whenever possible by statistical testing; the third stage requires comparing input-output transformations generated

by the model with those generated by the real world system.

Face validation is carried out by asking individuals knowledgeable about the system whether or not the model's behaviour is reasonable.

Turing test [114] and its extensions with statistical procedures [1, 16, 108] have been formulated as an attempt to deal with the question of whether or not a computer program can exhibit intelligence. In the simulation context, this test corresponds to asking people who are knowledgeable about the system if they can discriminate between system and model outputs.

Field tests are performed on the simulation model to establish whether or not the model can successfully replicate system input-output behaviour which has not yet been previously observed.

Extreme condition tests are performed to evaluate the model's behaviour with extreme and unlikely inputs.

Animation is a useful validation tool which gives the ability to observe the model's operational behaviour graphically as the model moves through time.

Traces refer to the activity of following the behaviour of entity types as they evolve over time and confirming their correct logical relationships.

Sensitivity analysis usually consists of systematically varying the values of the inputs over some range of interest and observing the effect upon the model's behaviour. Such a study can help to build confidence in the model if the same I-O relationships occur in the model as in the system.

Although there is a large variety of techniques for model validation, Sargent [106] correctly observes that there is no set of specific procedures that can be easily applied to determine the validity of the model, and furthermore no procedure exists to

determine what techniques/procedures to use.

A comprehensive view of the validation problem is developed in [111]. This work views the validation problem in a far broader context than that taken in our investigation.

Artificial intelligence techniques have been incorporated in many aspects of modeling and simulation. This has, in particular, included the validation problem [26, 33, 70, 80, 105, 122]. Its application in model development [31, 55, 71, 74, 81, 82, 85, 94] and in the development of new simulation languages and/or environments [37, 79, 83, 84] have also been extensive.

The work of Findler [33] is particularly relevant since his approach to incorporate artificial intelligence methodology into the model validation problem shares some common features with our own approach. His approach consists of three phases of activity. First, certain properties of the simulation model are collected via a menu-driven dialog system. Next, a "meta-model" of the simulation model, which has the characteristics amenable to the processes of verification and validation is constructed through a process of experimenting with the simulation model. Finally, structural and statistical analyses are carried out on the meta-model to reveal defects that may exist in the simulation model. These are then eliminated through an iterative process.

1.3 Review of Relevant Software Testing Concepts

Like simulation model validation, software testing is concerned with assessing the compliance of a program to its intended specifications. Because of this close inter-

relationship, we review in this section some important software testing concepts. These concepts have significantly influenced the evolution of the work described in this thesis.

An external specification describes the intended (expected) behaviour of a program from its user's point of view. To determine whether or not discrepancies between the program and its (external) specification are present, a process called function testing [49, 50] is used. Function testing is generally a black-box oriented (input-output driven) activity in which the tester is not concerned about the internal code of the program but rather is interested in detecting situations where the program does not behave according to its specification.

Function testing based on exhaustive input testing; i.e., use of every possible input condition as a test-case, is usually impossible. Instead, the specification is analyzed to derive a set of test-cases by techniques such as equivalence partitioning, boundary-value analysis (extreme condition testing) and cause-effect graphing.

The equivalence partitioning method is largely a heuristic process [78]. Its underlying idea is that each test-case should involve as many different input conditions as possible to minimize the total number of test-cases necessary. The approach is based on partitioning the input domain into a finite number of equivalence classes. A test which uses a representative value of any particular class is deemed to be equivalent to a test using any other value within the class.

In some cases boundary-value analysis can be more effective than equivalence partitioning because it explores boundary (extreme) conditions which are directly on, above, or below the boundaries of the equivalence classes.

Cause-effect graphing is a method of generating test-cases representing combi-

nations of conditions. In using this technique, the natural-language specification is analyzed and causes and effects in the specification are identified and represented in a graphical manner. Each representative member of an input class is a cause, and an effect is the output condition which results from the input condition(s). The resulting graph is annotated with constraints describing the relationships among causes and effects; the graph is actually a Boolean logic network. By tracing state conditions in the graph, the graph is normally converted into a limited-entry decision table which has a beneficial side effect in pointing out incompleteness and ambiguities in the specification [64]. The columns in the decision table are converted into test-cases. The process also allows boundary conditions to be blended into the test-cases derived from the cause-effect graph.

The test-case design techniques discussed above are described in considerable detail in the software testing literature [72, 73]. These techniques can be combined into an overall strategy since each contributes a particular set of useful test-cases. However, even a combined strategy will not guarantee that all errors will be found i.e., that the program is error-free. Test-case design is a difficult problem and in fact the general problem is unsolvable [15, 67, 68]. Examples of approximation methods can be found in [15, 25, 77, 92].

Test-case development, execution and evaluation are generally very time consuming and labor-intensive tasks, and the tester must be satisfied with examining the results of a finite number of test-cases which are likely to have adequate coverage of the problem. Test data selection is a key component in software testing and has received considerable attention in the literature [24, 45]. Considerable on-going research effort is directed towards automating the software testing process [22, 116].

1.4 Review of Relevant Qualitative Modeling Concepts

The expected behaviour of a simulation model can be characterized using a qualitative model. A qualitative (structural) model is a collection of elements of a system and relationships among their states. When the relationships are causal in nature, the resulting model is called a causal model [62]. Although causal models focus primarily on qualitative features of the system, some quantitative information can also be incorporated. Three conditions must normally be met in order to infer the existence of a causal relationship between the states of two elements X and Y of a system [2]:

1. There must be covariation between the states of X and Y (a change in X is expected to result in a change in Y),
2. There must be a temporal asymmetry or time ordering between the states of X and Y (cause must occur before its effect, i.e., causal ordering),
3. The covariation between the states of X and Y should not vanish when those elements causally prior to both X and Y are removed.

Two basic characteristics of a causal model [51] are:

- it reveals the causal relationships among the states of the system, and
- it contains domain knowledge that permits reasoning about the behaviour of the system to be carried out.

These characteristics of causal models make them valuable tools for knowledge representation.

Structuring a knowledge base in the form of a causal model has certain advantages:

- There is a close correspondence between the mental model of the expert and the knowledge representation of the system which leads to easier knowledge acquisition.
- Concern is entirely with domain knowledge representation and there is no need for consideration of control knowledge which results in case of knowledge base maintenance.
- Causal models represent domain knowledge in a concise form which consequently provides an appreciation for the system as a "whole".
- The analysis of the patterns formed by the relationships can often yield valuable insights into likely behavioural properties of the system.
- A reasonably deep understanding of the problem domain is required in order that its structure and/or behaviour can be expressed in the causal model format. This therefore obliges a potential user to undertake more extensive study before a modeling activity is initiated.

Extensive work has been done with causal models in variety of disciplines; e.g., in social sciences [2], systems science [35], chemistry [115], and artificial intelligence [21, 23, 34, 59, 60, 96, 112, 117]. Presently, much research in artificial intelligence is focusing on the problems of causal reasoning (reasoning with causal models) in knowledge acquisition, explanation, trouble-shooting and diagnostics; e.g., problems in physics, electronics and medicine.

1.5 Proposed Approach for the Knowledge-Based Validation of Simulation Models

Our knowledge-based approach to the validation of simulation models focuses on the characterization of anticipated behaviour of the system being studied in order to specify experiments with the model which will provide confidence in its correct behaviour. This approach to validating simulation models is similar in principle to that in software testing. It must be established that the target system (i.e. simulation model) meets its requirements, in other words it must be shown that the model behaves as expected as implied by some set of specifications. A correspondence with the function testing in the software testing context is clearly evident here.

In our approach we capture the specification of the simulation model in a qualitative model which strives to characterize some potential interactions among the elements of the system and provides a means for studying the expected behaviour of the system. This expected behaviour constitutes a knowledge base for our approach which, in turn, establishes the basis for dealing with two fundamental issues of simulation model validation; namely, designing experiments and evaluating the results of experiments.

The underlying idea of the approach is to automatically design from the knowledge base the experiments (test-cases) to be carried out with the simulation model. The results of the execution of the simulation model (i.e., observed behaviour) for these experiments are checked against the expected behaviour in the knowledge base. If the observed behaviour of the model is in agreement with the expected behaviour then the model is accepted as being valid.

In this thesis we primarily focus our efforts on designing experiments. An important practical requirement for the experiment design is to obtain a minimum number of experiments which encompasses the expected behaviour. This requirement arises because each experiment with the simulation model may consume substantial computing resources. An appropriate example of this point can be found in [69, p.39] : "a telephone switch generates about 100 internal messages in completing a local call. Large telephone switches can handle 100 or more calls per second. Thus, simulation of a telephone switch for 15 minutes of real time requires the simulation of nearly 10 million messages, which will require several hours on a very fast uniprocessor". Experiment design is a difficult problem because of this minimization objective. It is interesting to note that the major task in software testing is also test-case design which is known to be a difficult problem [15, 67, 68].

Our knowledge-based approach has the following noteworthy features which, to some extent, it shares with software testing techniques and with qualitative modeling:

- It has the potential to provide an efficient validation process by generating, in a systematic way, a non-redundant high-yield set of test-cases.
- It has the potential to support clear explanations during the validation study thereby providing a more user friendly system.
- It can be effective for both observable and non-observable systems because the problem originator's knowledge provides the major portion of the validation reference information.
- It is practical and natural since the underlying process duplicates to a large extent the most direct approach that would be undertaken by a "human val-

idator".

- It is independent of the underlying simulation formalism; i.e., it is equally applicable to continuous and discrete event models.

1.6 Contributions of the Thesis

The results obtained in this research study relate to the validation problem for simulation models. Our main contribution is the formulation of a knowledge-based approach to this problem and the development of algorithms which solve a key subproblem that emerges in the approach.

Our knowledge-based approach is proposed as a means for incorporating domain specific knowledge into the simulation model validation process. A formal framework is first developed within which a precise formulation of our approach can be presented. This framework is developed around the concept of a "dynamic object" (see section 2.1.1) which has a simple characterization but nevertheless provides a sufficiently powerful means for representing the behaviour of the class of entities that are relevant to our study.

Our knowledge-based approach relies on the specification of a set of properties of the simulation model's behaviour. This set of properties, which is called the simulation model's external specification, can be precisely expressed using our formalism developed for dynamic objects. The external specification (see section 2.1.2) is, furthermore, shown to be a union of three disjoint sets of specifications, each of which represents a particular class of domain knowledge about the simulation model under investigation.

A key aspect of the validation process is the problem of designing experiments which properly "exercise" the external specification. A crucial requirement here is the efficient design of these experiments because of the potentially high cost associated with their execution. Consequently a second phase of our work focusses on the development of efficient solutions to the experiment design problem.

This problem is formulated as a constraint satisfaction problem (see section 3.1) that has an additional optimization aspect. We refer to this problem as the C^3 problem and we show the relationship between this problem and two well-known problems in graph theory; namely, the edge cover problem and the clique cover problem. Our justification for heuristic approaches for solving the C^3 problem is based on the results from this analysis.

A greedy approximation algorithm for obtaining an approximate solution to the C^3 problem is first presented for a special case where one particular category of behavioural specifications is absent. The effectiveness of this algorithm is formally demonstrated and some properties of its performance are also provided. A general version of the algorithm is then developed, based on a transformation technique that makes it possible to accomodate the special class of behaviour specifications in the previously developed procedure.

In summary, the thesis proposes a new, knowledge-based, approach for handling the simulation model validation problem and provides a formal framework for its characterization and investigation. An important underlying subproblem within the approach is that of efficiently designing experiments from a knowledge base of expected behaviour. Our proposed format for the knowledge base facilitates the development of an effective solution algorithm for this subproblem. This algorithm provides the

basis for the realization of a key element of our proposed global architecture for a knowledge-based software environment for the validation of simulation models.

1.7 Outline of the Thesis

The remainder of this thesis is structured as follows:

In chapter 2, we develop a knowledge-based approach to the simulation model validation problem. Specifically, a formal context for the validation problem is developed. Based on this formal context, we present a software environment for handling the validation problem. We define the experiment design problem and point out that its solution is the central issue for an effective and efficient validation process.

In chapter 3, we formulate the experiment design problem in a constraint framework which is referred to as the C^3 problem. We analyze the complexity of the C^3 problem in chapter 4. We show that a simplified version of the C^3 problem is related to two well known problems in graph theory; namely, the clique cover problem and the edge cover problem. Because of its relation to the clique cover problem which is known to be a difficult problem, an approximation rather than optimal algorithm for the C^3 problem is developed and its important properties are demonstrated in chapters 5 and 6. We summarize the results of this thesis and discuss the directions for further research in chapter 7.

An example of external specification and generalized consistency definitions for the C^3 problem are given in Appendices A and B respectively. Several examples of the C^3 problem are presented in Appendix C and the results obtained using the proposed algorithm are discussed.

Chapter 2

Generalized Framework for the Validation of Simulation Models

In this chapter we present a knowledge-based approach to the simulation model validation problem which incorporates the concepts of software testing and qualitative modeling as outlined in sections 1.3 and 1.4. We believe that the basic criteria in validating simulation models should be similar in principle to that in software testing. It must be established that the target system meets its requirements, which in the case of a simulation model, correspond to a specification of expected behaviour. In our approach we capture the specifications of the simulation model in a qualitative model which strives to characterize all potential interactions and provides a means for studying the expected behaviour of the simulation model. This characterization of expected behaviour corresponds to a knowledge base. This knowledge base has a two-fold function; it is used to automatically design test-cases (experiments) to be carried out with the simulation model and it provides reference information against which the model's behaviour can be checked. If the behaviour is consistent with the knowledge base then the simulation model can be accepted as being valid.

The approach differs from the existing statistical validation techniques because it utilizes both domain specific knowledge provided by an expert which describes the intended behaviour of the system as well as data acquired from an observable system.

In section 2.1, we develop a formal basis for the simulation model validation problem which is formulated around dynamic objects. We discuss in section 2.2 an interpretation as it relates to the validation of simulation models in practice. In section 2.3, we present the global architecture for a simulation model validation software environment.

2.1 A Formal Context for the Validation Problem

Our approach is formulated around a generalized characterization of both model and system behaviour. This characterization provides a formal basis for formulating the notion of an experiment with a model (or system) as well as various important relationships between experiments that have special relevance to the validation activity. These relationships enable the precise specification of a knowledge base which provides validation reference information.

The presentation below relies on the following notational conventions:

- For the vector v, we use v_j to denote the j^{th} component of v and $\mid v \mid$ to denote the length of v; i.e. the number of components in v.
- The cardinality of a set S is denoted by $\mid S \mid$.
- For n a positive integer, we use I_n to denote the set $\{1, 2, \cdots, n\}$ and $\bar{I}_n$ to denote the set $I_n \bigcup \{0\}$.

2.1.1 Dynamic Objects and Their Behaviour

Our interest throughout this study is with dynamic objects whose fundamental property is their ability to generate (exhibit) behaviour over some prescribed time interval. From our perspective, a dynamic object, O, is an ordered pair of real-valued vectors; i.e., $O = (X, Y)$ where X and Y respectively correspond to the (generalized) input and output of the dynamic object. The implication here is that a dynamic object is indistinguishable from the specifications of its input and output vectors (included in these specifications is the meaning associated with the individual components of X and Y).

For example, in a model of an automobile suspension system, some of the components of X might correspond to the stiffness of the spring and the damping coefficient of the shock absorber while some of the components of Y might correspond to the maximum displacement of the automobile (over the time interval of interest) and the average force applied to the automobile chassis.

It should also be noted here that we adopt a very broad point of view with respect to our interpretation of the input vector. In particular, it is taken to include initial state values and parameter values, in addition to the more conventional notion of "inputs".

It is also important to emphasize that an implicit causal relationship is assumed to exist between the input and output vectors of a dynamic object. This, in particular, implies that changes in the values taken by X will generally result in changes in the values acquired by Y. The mechanism that transforms X values into Y values is not of any particular interest in our considerations.

A number of interesting inter-relationships among dynamic objects can be identified. For example, a dynamic object $O = (X, Y)$ can be obtained from a dynamic object $O' = (X, Y')$ by constructing Y as a subset of the components of Y'. In such a case, O is regarded as a "simple restriction" of O'.

Two dynamic objects, $\hat{O} = (\hat{X}, \hat{Y})$ and $\tilde{O} = (\tilde{X}, \tilde{Y})$ are said to be isomorphic if $| \hat{X} | = | \tilde{X} |$ and a meaningful association can be identified between distinct pairs of components of $\hat{X}$ and $\tilde{X}$ and, in addition, there exist non-empty subsets $\hat{Y}' \subseteq \hat{Y}$ and $\tilde{Y}' \subseteq \tilde{Y}$ with $| \hat{Y}' | = | \tilde{Y}' |$ such that a meaningful association can be identified between distinct pairs of components of $\hat{Y}'$ and $\tilde{Y}'$. The notion of a "meaningful association" here is left somewhat vague to allow for a variety of practically useful interpretations.

Isomorphic dynamic objects share a similarity with respect to the meaning of corresponding pairs of components within their respective input and output vectors. Non-meaningful elements may exist within both output vectors (but not within the input vectors). With both the input and the output vectors, a simple re-indexing operation will enable corresponding elements to share common indices. In our subsequent references to isomorphic dynamic objects, we assume that such a re-indexing has taken place. In addition, we assume that, where appropriate, such pairs of isomorphic objects are replaced by the respective "simple restrictions" that result when non-meaningful output elements are removed. This simplifies the treatment of inter-relationships between these dynamic objects.

An important aspect of our interest with dynamic objects is their observable behaviour as reflected in values acquired by their output vectors. This particular aspect of the behaviour of a dynamic object $O = (X, Y)$ is described in terms of

a sequence of values acquired by Y over some finite interval of time. These values correspond to "snapshots" or instantiations of Y at the points contained in some finite non-empty time set that contains real scalar values; i.e. $T = \{t_0, t_1, \cdots, t_N\} = \{t_i : i \in \bar{I}_N\}$[1]. Any particular case can be represented as $\{Y^k(t_i) : t_i \in T\}$ which we denote by $Y^k(T)$. The causality implicit in our notion of a dynamic object, implies that the set of values $Y^k(T)$ is associated with a corresponding sequence of values for the vector X; i.e. $X^k(T) = \{X^k(t_i) : t_i \in T\}$. In effect, the sequence $X^k(T)$ provides a "context" for the observed behaviour represented by $Y^k(T)$.

The perspective outlined above for the output vector of a dynamic object may appear overly restrictive inasmuch as all components of Y are regarded as having an observable (and meaningful) value at each value within the time set. Situations which do not correspond to this perspective can be easily identified; e.g., the case where the j^{th} component of Y represents a performance measure that is defined only at some fixed point in time. Such situations can, however, be easily accomodated by simply assigning a generic value (i.e., the symbol '$*$') to Y_j at those values of the time set that are not relevant. To maintain the relevance of relationships that are introduced in our subsequent discussion, we adopt the convention that $*$ satisfies any binary relation $\mathrm{r} \in \{=, \neq, <, >, \leq, \geq\}$; i.e., $*\mathrm{r}*$ is true and both $*\mathrm{r}v$ and $v\mathrm{r}*$ are true for any explicit (real) value v.

The sets T, $X^k(T)$ and $Y^k(T)$, taken together, provide one instance of the possible behaviour which a dynamic object $O = (X, Y)$ can exhibit. Correspondingly, we regard the triplet $(T, X^k(T), Y^k(T))$ as a behaviour instance of the dynamic object $O = (X, Y)$ which we denote by $B_k(O, T)$.

[1] We assume for convenience that the time set T is ordered in the sense that if $t_i, t_j \in T$ and $i < j$, then $t_i < t_j$.

The notation outlined above provides a sufficiently general framework to describe many useful system properties. For example, the distinction between time variant and time invariant system can be dealt with in the following way.

Let $B_k(O,T) = (T, X^k(T), Y^k(T))$ and $B_l(O,\breve{T}) = (\breve{T}, X^l(\breve{T}), Y^l(\breve{T}))$ with $\breve{T} = \{t_i + \tau : t_i \in T \text{ and } \tau > 0\}$ be two behaviour instances of a dynamic object $O = (X,Y)$. Suppose $X^k(T) = X^l(\breve{T})$. Dynamic object O is <u>time invariant</u> if $Y^k(T) = Y^l(\breve{T})$ otherwise it is <u>time variant</u>.

The set of ordered pairs $\{(t_i, X^k(t_i)) : t_i \in T\}$ associated with a behaviour instance $B_k(O,T)$ is referred to as the <u>input trajectory</u> of the behaviour instance $B_k(O,T)$. This set is denoted by $\theta(B_k(O,T))$. The set $\{(t_i, Y^k(t_i)) : t_i \in T\}$ is referred to as the <u>output trajectory</u> of the behaviour instance $B_k(O,T)$. This set is denoted by $\phi(B_k(O,T))$. Similarly the set $\{(t_i, X^k_j(t_i)) : t_i \in T\}$ which is called the jth input trajectory of $B_k(O,T)$, is denoted by $\theta_j(B_k(O,T))$ and the set $\{(t_i, Y^k_j(t_i)) : t_i \in T\}$ which is called the jth output trajectory of $B_k(O,T)$, is denoted by $\phi_j(B_k(O,T))$. The first and last values in T; i.e., t_0 and t_N are referred to respectively as the <u>initial</u> and the <u>terminal</u> times of the behaviour instance.

Let $B_k(\hat{O},T_a)$ and $B_l(\tilde{O},T_b)$ be two behaviour instances associated with isomorphic dynamic objects $\hat{O}$ and $\tilde{O}$ respectively. $B_k(\hat{O},T_a)$ is <u>time compatible</u> with $B_l(\tilde{O},T_b)$ if $T_a \bigcap T_b = T_a$.

A basic requirement for our considerations is a formal means for characterizing how behaviour instances can be inter-related. This characterization is in terms of relationships. Two fundamental types can be identified; namely those where a change in value exists and those where it does not.

In order to expand on these notions, we consider two time compatible behaviour

instances, $B_k(\hat{O}, T_a)$ and $B_l(\tilde{O}, T_b)$ of $\hat{O} = (\hat{X}, \hat{Y})$ and $\tilde{O} = (\tilde{X}, \tilde{Y})$ respectively, and suppose that $T_a \bigcap T_b = T_a$. (Note that in the event that $\hat{O}$ and $\tilde{O}$ are identical, we assume that $T_a = T_b$.)

The absence of change corresponds to an equality relationship. We use the predicate $\delta_j(\hat{X}^k, \tilde{X}^l, =)$ to indicate the equality relationship that corresponds to $\hat{X}_j^k(t_i) = \tilde{X}_j^l(t_i)$ for all $t_i \in T_a$. If $\delta_j(\hat{X}^k, \tilde{X}^l, =)$ holds for all $j \in I_{|\hat{X}|}$, then we write $\delta(\hat{X}^k, \tilde{X}^l, =)$. Similarly we use the predicate $\delta_j(\hat{Y}^k, \tilde{Y}^l, =)$ to indicate the equality relationship that corresponds to $\hat{Y}_j(t_i) = \tilde{Y}_j(t_i)$ for all $t_i \in T_a$. If $\delta_j(\hat{Y}^k, \tilde{Y}^l, =)$ holds for all $j \in I_{|\hat{Y}|}$, then we write $\delta(\hat{Y}^k, \tilde{Y}^l, =)$.

Change can occur in a wide variety of forms. We outline below a particular group of two change relationships that can exist between the jth input trajectories. We call these change-in-value relationships (CVR's).

The jth input trajectory $\theta_j(B_k(\hat{O}, T_a))$ and the jth input trajectory $\theta_j(B_l(\tilde{O}, T_b))$ are related by:

i) an ϵ - CVR if either

a) $\tilde{X}_j^l(t_i) - \hat{X}_j^k(t_i) \geq 0 \qquad \forall t_i \in T_a$ and
there exists at least one $t_i \in T_a$ such that $\tilde{X}_j^l(t_i) - \hat{X}_j^k(t_i) > \epsilon$ where $\epsilon \geq 0$,
or

b) $\hat{X}_j^k(t_i) - \tilde{X}_j^l(t_i) \geq 0 \qquad \forall t_i \in T_a$ and
there exists at least one $t_i \in T_a$ such that $\hat{X}_j^k(t_i) - \tilde{X}_j^l(t_i) > \epsilon$ where $\epsilon \geq 0$.

We denote these two cases with the predicates $\delta_j(\hat{X}^k, \tilde{X}^l, \epsilon)$ and $\delta_j(\tilde{X}^l, \hat{X}^k, \epsilon)$ respectively.

ii) an $|\epsilon|$ - CVR if

$|\hat{X}_j^k(t_i) - \tilde{X}_j^l(t_i)| \leq \epsilon \qquad \forall t_i \in T_a$ where $\epsilon \geq 0$.

We denote such an $|\epsilon|$ - CVR by the predicate $\bar{\delta}_j(\hat{X}^k, \tilde{X}^l, \epsilon)$.

In a similar way the input trajectories $\theta(B_k(\hat{O}, T_a))$ and $\theta(B_l(\tilde{O}, T_b))$ are related by:

i) an ϵ^* - CVR if either

a) $\delta_j(\hat{X}^k, \tilde{X}^l, \epsilon) \wedge [\delta_i(\hat{X}^k, \tilde{X}^l, =)$ for all $i \neq j]$ or

b) $\delta_j(\tilde{X}^l, \hat{X}^k, \epsilon) \wedge [\delta_i(\tilde{X}^l, \hat{X}^k, =)$ for all $i \neq j]$.

We denote these two cases with the predicates $\delta_j^*(\hat{X}^k, \tilde{X}^l, \epsilon)$ and $\delta_j^*(\tilde{X}^l, \hat{X}^k, \epsilon)$ respectively.

ii) an $|\epsilon^*|$ - CVR if

$\bar{\delta}_j(\hat{X}^k, \tilde{X}^l, \epsilon) \wedge [\delta_i(\hat{X}^k, \tilde{X}^l, =)$ for all $i \neq j]$.

We denote such an $|\epsilon^*|$ - CVR by the predicate $\bar{\delta}_j^*(\hat{X}^k, \tilde{X}^l, \epsilon)$.

Similar definitions apply to output trajectories; namely, the jth output trajectory $\phi_j(B_k(\hat{O}, T_a))$ and the jth output trajectory $\phi_j(B_l(\tilde{O}, T_b))$ are related by:

i) an ϵ - CVR if either

a) $\tilde{Y}_j^l(t_i) - \hat{Y}_j^k(t_i) \geq 0 \qquad \forall t_i \in T_a$ and

there exists at least one $t_i \in T_a$ such that $\tilde{Y}_j^l(t_i) - \hat{Y}_j^k(t_i) > \epsilon$ where $\epsilon \geq 0$,

or

b) $\hat{Y}_j^k(t_i) - \tilde{Y}_j^l(t_i) \geq 0 \qquad \forall t_i \in T_a$ and

there exists at least one $t_i \in T_a$ such that $\hat{Y}_j^k(t_i) - \tilde{Y}_j^l(t_i) > \epsilon$ where $\epsilon \geq 0$.

We denote these two cases with the predicates $\delta_j(\hat{Y}^k, \tilde{Y}^l, \epsilon)$ and $\delta_j(\tilde{Y}^l, \hat{Y}^k, \epsilon)$ respectively. If $\delta_j(\hat{Y}^k, \tilde{Y}^l, \epsilon)$ holds for all $j \in I_{|\hat{Y}|}$, then we write $\delta(\hat{Y}^k, \tilde{Y}^l, \epsilon)$. Similar remarks apply for $\delta_j(\tilde{Y}^l, \hat{Y}^k, \epsilon)$.

ii) an $|\epsilon|$ - CVR if

$|\hat{Y}_j^k(t_i) - \tilde{Y}_j^l(t_i)| \leq \epsilon \qquad \forall t_i \in T_a$ where $\epsilon \geq 0$.

We denote such an $|\epsilon|$ - CVR by the predicate $\bar{\delta}_j(\hat{Y}^k, \tilde{Y}^l, \epsilon)$.

If $\bar{\delta}_j(\hat{Y}^k, \tilde{Y}^l, \epsilon)$ holds for all $j \in I_{|\hat{Y}|}$, then we write $\bar{\delta}(\hat{Y}^k, \tilde{Y}^l, \epsilon)$.

Note that the predicates δ, δ^*, $\bar{\delta}$ and $\bar{\delta}^*$ suppress any reference to the dynamic objects in question and can be used only when such information is clear from the context.

Definition — Compatible Behaviour Instances

Let $B_k(\hat{O}, T_a)$ and $B_l(\tilde{O}, T_b)$ be two behaviour instances where $B_k(\hat{O}, T_a)$ is time compatible with $B_l(\tilde{O}, T_b)$. If $\delta(\hat{X}^k, \tilde{X}^l, =)$, then $B_k(\hat{O}, T_a)$ is compatible with $B_l(\tilde{O}, T_b)$.

Definition — Identical Behaviour Instances

Compatible behaviour instances with the same dynamic object are defined as identical behaviour instances.

Definition — Δ-Consistent Behaviour Instances

Let $B_k(\hat{O}, T_a)$ and $B_l(\tilde{O}, T_b)$ be two behaviour instances where $B_k(\hat{O}, T_a)$ is compatible with $B_l(\tilde{O}, T_b)$. If for each $j \in I_{|\hat{Y}|}$, $\bar{\delta}_j(\hat{Y}^k, \tilde{Y}^l, \epsilon_j)$, then $B_k(\hat{O}, T_a)$ is Δ-consistent with $B_l(\tilde{O}, T_b)$ where $\Delta = \max\{\epsilon_j\}$.

Remark — It is fundamental to our considerations that identical behaviour instances always be Δ-consistent with $\Delta = 0$.

The abstract system model developed in this section as a vehicle for supporting our study, shares some general features with similar models used in the systems theory literature (e.g., Klir [56] and Wymore [119]). It is however distinctive in two significant ways; namely, there is no explicit reference in the model to the process which transforms the systems input into an output and there is no explicit reference to the system state vector. These differences, in fact provide an especially simple model structure which is, nevertheless, appropriate for the requirements of our study.

Our notion of a behaviour instance includes some elements of the notions of an experimental frame and an experiment as proposed by Ören and Zeigler [87] and Zeigler [121].

2.1.2 External Specification of a Dynamic Object

We regard a dynamic object's behaviour as the set of all possible behaviour instances which the dynamic object can exhibit. It is of fundamental importance to be able to characterize in some effective way the behaviour of a dynamic object. The nature of this characterization depends, to a large extent, on the nature of the study that is being carried out with the dynamic object. Within the context of our concern with validation, such a characterization can be achieved by identifying relationships that must be satisfied by the input and output trajectories of the dynamic object. We refer to such a set of relationships as an external specification of a dynamic object, O, which we denote by $ES(O)$. Each relationship in $ES(O)$ is formulated to describe some aspect of the dynamic object's "expected" behaviour.

We regard an external specification of a dynamic object O as the composition of three disjoint sets of relationships; namely:

- a set of Formal Specifications ($FS(O)$)
- a set of Qualitative Specifications ($QS(O)$)
- a set of Observational Specifications ($OS(O)$)

i.e., $ES(O) \equiv FS(O) \cup QS(O) \cup OS(O)$

Definition — Formal Specifications

A formal specification of a dynamic object $O = (X, Y)$ is a non-causal relationship which is of the form $L(t, X(t), Y(t))$ where (i) $L(\ )$ is a boolean expression defined on its arguments and (ii) $L(\ )$ must hold for every behaviour instance $B(O, T)$.

If a formal specification $L(\ )$ depends only on t and X, then $L(\ )$ is called an "input feature" of the dynamic object O; otherwise it is called a "behavioural feature".

With most simulation models, there are known relationships among the components of the input and output vectors that must always hold. For example, in an automobile simulation, vertical displacement of the automobile must never be negative or in a queuing problem the server utilization must not exceed unity. The purpose of our notion of formal specifications is to accomodate such known relationships.

Before examining more precisely the concepts of qualitative and observational specifications, it is necessary to extend somewhat our notion of a behaviour instance. A large class of dynamic objects that are of interest have an inherent stochastic nature which implies that a single behaviour instance does not properly reflect properties of the object's output. Instead, some aggregate (e.g., an average) of a number of related behaviour instances needs to be obtained. In such a family of related behaviour instances, the input remains invariant and output variations arise because of the stochastic effects internal to the dynamic object. A suitable aggregation of the output

values that are generated, produces a "reliable estimate" for the object's output vector.

To accomodate the above perspective, we introduce the notion of an "extended" behaviour instance for a dynamic object $O = (X, Y)$, which we denote by $\bar{B}(O, T)$. Such an extended behaviour instance implies the use of a reliable estimate for the output vector, obtained via some suitable aggregation of values obtained from a set of behaviour instances which share a common value for the input vector.

The definitions which are provided below for qualitative and observational specifications are based on this notion of extended behaviour instances. It should, however, be emphasized that wherever appropriate, references to extended behaviour instances can be replaced with "ordinary" behaviour instances.

The presentation which follows uses implications and we refer to the left-hand-side of any such implication as its "cause part" and its right-hand-side as its "effect part".

Definition — Qualitative Specifications

A qualitative specification for a dynamic object $O = (X, Y)$ is a causal relationship which has one of the following forms:

1. **Ordinary causal relationship (OCR)**

   An OCR is an implication of the form:

   $$L^c(t, X(t), \overline{Y}(t)) \longrightarrow L^e(t, X(t), \overline{Y}(t))$$

   where

   (i) $L^c(\,)$ and $L^e(\,)$ are boolean expressions defined on their respective arguments, and $\overline{Y}(t)$ represents a "reliable estimate" of the output vector associated with O and

(ii) if $L^c(\ )$ holds for any extended behaviour instance $\bar{B}(O,T)$ then $L^e(\ )$ must also hold.

If $L^c(\ )$ and $L^e(\ )$ both depend only on t and X, then this qualitative specification is called an "input feature" of the dynamic object O; otherwise it is called a "behavioural feature".

In many simulation models there exist known causal relationships among the components of the input and output vectors. For example, in an automobile simulation, the engine torque must reduce to zero if the fuel level has reached zero.

2. **Change-in-value causal relationship (CVCR)**

A CVCR is an implication which relates to a pair of extended behaviour instances. Its form is:

$$L^{cc}(t, X(t), X'(t), \epsilon) \longrightarrow L^{ec}(t, \overline{Y}(t), \overline{Y}'(t), \epsilon)$$

where

(i) $L^{cc}(t, X(t), X'(t), \epsilon) = \delta_j^*(X, X', \epsilon) \wedge L^{cv}(t, X(t)) \wedge\ L^{cv}(t, X'(t))$

for some $j \in I_{|X|}$ where $L^{cv}(\ )$ is optional,

$L^{ec}(t, \overline{Y}(t), \overline{Y}'(t), \epsilon) = \delta_j(\bar{Y}, \bar{Y}', \epsilon) \oplus \delta_j(\bar{Y}', \bar{Y}, \epsilon) \oplus \bar{\delta}_j(\bar{Y}, \bar{Y}', \epsilon)$

for some $j \in I_{|Y|}$

(the symbol $\oplus$ represents the exclusive-or operator) and

(ii) if $L^{cc}(\ )$ holds for any pair of extended behaviour instances $(\bar{B}_k(O,T), \bar{B}_l(O,T))$ then $L^{ec}(\ )$ must also hold. That is, if $\delta_j^*(X^k, X^l, \epsilon)$ (i.e., $\delta_j^*(X, X', \epsilon)$ is TRUE for the pair $(\bar{B}_k(O,T), \bar{B}_l(O,T))$ for some given ϵ), and $L^{cv}(t, X(t))$

holds for both $\bar{B}_k(O,T)$ and $\bar{B}_l(O,T)$, then $\delta_j(\bar{Y}^k,\bar{Y}^l,\epsilon) \oplus \delta_j(\bar{Y}^l,\bar{Y}^k,\epsilon) \oplus \bar{\delta}_j(\bar{Y}^k,\bar{Y}^l,\epsilon)$

Frequently, some general features in the behaviour of a simulation model are expected to be observed when the value of some input component is changed. For example, in a queuing problem, if the number of servers is increased then the average queue length should decrease (here we assume that all other input components remain unchanged). Often however, the change in value of the input must exceed some threshold before any observable change can be expected. This threshold concept corresponds to the predicate $\delta_j^*(X,X',\epsilon)$ in the above.

Definition — Observational Specifications

An observational specification for a dynamic object $O = (X,Y)$ is a causal relationship which is of the form:

$$L^{co}(X,\tilde{\tilde{X}}) \longrightarrow L^{eo}(\overline{Y},\tilde{\tilde{Y}},\epsilon)$$

where

(i) $L^{co}(X,\tilde{\tilde{X}}) = \delta(X,\tilde{\tilde{X}},=)$,
$L^{eo}(\overline{Y},\tilde{\tilde{Y}},\epsilon) = \bar{\delta}(\bar{Y},\tilde{\tilde{Y}},\epsilon)$.
$\tilde{\tilde{X}}$ and $\tilde{\tilde{Y}}$ are the input and output vectors of some isomorphic dynamic object $\tilde{\tilde{O}}$ such that $B(O,T)$ is time compatible with $B(\tilde{\tilde{O}},\tilde{\tilde{T}})$, and

(ii) if $L^{co}(\ )$ holds for any pair of extended behaviour instances $(\bar{B}_k(O,T), B_l(\tilde{\tilde{O}},\tilde{\tilde{T}}))$ then $L^{eo}(\ )$ must also hold. That is, if $\delta(X^k,\tilde{\tilde{X}}^l,=)$ then $\bar{\delta}(\bar{Y}^k,\tilde{\tilde{Y}}^l,\epsilon)$ for some given ϵ.

Our notion of observational specifications is intended to handle the situation where behaviour data is available either from a "real" system or from another previously

validated simulation model. The objective in such circumstances is to ensure that the behaviour of the model being validated is consistent with the available data (outputs are "close" when the inputs are the same).

2.1.3 The Experiment Design Problem

Validation, in the context of dynamic objects, is the process of showing that a dynamic object's behaviour (i.e., set of all possible behaviour instances) is consistent with any particular external specification. However this is not possible in practice. Therefore from a more practical point of view, validation is the process of showing that *any particular subset of* the dynamic object's behaviour satisfies any particular external specification.

A subset of a dynamic object's behaviour can be obtained by carrying out experiments with the dynamic object. The carrying out of an experiment is the process of activating the dynamic object so that it exhibits one of its possible behaviour instances. This is achieved by specifying an input trajectory for the experiment which has certain properties that are of interest to the experimentor. We refer to such an input trajectory which is designed to attain a particular goal, as an input specification (or simply, specification) for that experiment. In fact throughout our discussion, we use the term "experiment" as an informal reference to its input specification.

Therefore, a fundamental aspect of the validation problem is concerned with designing experiments in an effective and efficient way. We call this the experiment design problem. Effectiveness is concerned with the coverage of the external specification whereas the efficiency is concerned with carrying out a minimum number of experiments. It is interesting to note that experiments for the validation of dynamic

objects are the counterparts of test-cases in software testing. Furthermore, the objective of designing experiments is similar to the objective of designing test-cases in software testing.

Suppose we carry out a set of experiments, Ω_{EXP}, with the dynamic object O which yields a set of behaviour instances which we denote by Ω_B. Suppose also that $ES(O) = FS(O) \bigcup \Omega$ is an external specification for O where $\Omega = QS(O) \bigcup OS(O)$. A specification in Ω is said to be covered by Ω_{EXP} if there exists at least one member within Ω_B which satisfies its cause part.

The experiment design problem is to determine a set of experiments $\Omega^{\star}_{EXP}$ which has the following properties:

(i) all input feature specifications in $FS(O)$ are satisfied (i.e., covered) by every member of $\Omega^{\star}_{EXP}$

(ii) every specification in Ω is covered by at least one member of $\Omega^{\star}_{EXP}$

(iii) each experiment in $\Omega^{\star}_{EXP}$ covers the largest possible number of specifications

(iv) there exists no other set of experiments that satisfies (i), (ii) and (iii) which has smaller cardinality than that of $\Omega^{\star}_{EXP}$

That subset of $FS(O)$ which are input features together with the boolean expressions which constitute the cause parts of the set Ω are of special interest because these expressions provide the basis for the experiment design. We refer to this set of expressions as the "test domain".

It should be observed that those members of $FS(O)$ which are behavioural features do not appear in the above description of the experiment design problem. This is because such specifications do not, in general, provide an explicit constraint on the

choice of the input vector X. These specifications however would be used in the evaluation of the results of the experiments (This is discussed further in section 2.3).

The experiment design problem is investigated in the subsequent chapters.

2.2 Knowledge Base for a Validation Software Environment

A fundamental characteristic of our approach to the validation problem is the acquisition of knowledge about the expected behaviour of the simulation model from a domain expert. It is essential to have a suitable framework for expressing this knowledge. The general framework for expressing an external specification for a dynamic object, as developed in the previous section, is entirely appropriate for this purpose because a dynamic object is a system abstraction that encompasses both the simulation model and the reference system.

In order to set the stage for our proposed architecture for a validation software environment, we address two issues related to the acquisition of knowledge about the expected behaviour of a simulation model; namely,

i) characterization of expected behaviour

ii) the refinement and validation of the knowledge about expected behaviour.

2.2.1 Characteristics of the Knowledge Base

A validation knowledge base for a simulation model is an external specification that describes expected behaviour of the simulation model. A knowledge base for a simulation model, in fact, is a collection of three disjoint sets of relationships; namely,

- Formal specifications
- Qualitative specifications
- Observational specifications

Formal specifications are factually true propositions and are analogous to facts in a rule-based system or a logic programming system (e.g., PROLOG). Formal specifications are intended to express those relationships on the inputs and outputs of a simulation model which must hold in every experiment carried out with the model.

Qualitative specifications are cause-effect relationships which characterize important aspects of the model's input-output behaviour. Qualitative specifications consist of two different kinds of causal relationships; namely, ordinary causal relationships and change-in-value causal relationships. Ordinary causal relationships are analogous to rules in a rule-based system or a logic programming system. Change-in-value causal relationships are intended to formalize relationships which characterize how changes in the input of the simulation model affect its output.

Observational specifications are concerned with the Δ - consistent behaviour instances of a simulation model. Suppose, for example, there is another reference model whose behaviour is known to correctly represent the desired behaviour of the simulation model under study. This reference model may be another previously validated

simulation model. Furthermore, suppose that reference data (a set of behaviour instances) obtained from the reference model is available. It is useful to use this reference data to examine the behaviour of the simulation model under study. Observational specifications are designed to address such scenarios.

Formal and qualitative specifications are typically provided by a domain expert whereas observational specifications correspond to data.

For the purpose of illustration, we present an external specification for a circuit-switching simulation model in Appendix A.

As with any knowledge base, the external specification for a simulation model will normally require refinement and validation. Once this step has been carried out, we refer to the resulting knowledge base as the validation reference information; i.e., external specification is transformed into validation reference information. This refinement and validation step is briefly examined in the following section.

2.2.2 Refinement and Validation of the Knowledge Base

An important prerequisite to the validation process is the refinement and validation of the knowledge base which has been provided for a simulation model in the form of an external specification. Refinement of the knowledge base is concerned with structuring the specifications so as to facilitate the development of algorithms for experiment design. Validation of the knowledge base, on the other hand, is concerned with the elimination of infeasible, conflicting and redundant specifications which unnecessarily tax the algorithms for experiment design. Our objectives in the refinement and validation of a simulation model knowledge base parallel those that typically apply in artifical intelligence applications and expert systems [10, 32, 42, 65, 66, 76, 113, 123].

We list below four steps related to the refinement and the validation of a simulation model knowledge base; steps 1, 2 and 3 are related to its refinement whereas step 4 is related to its validation.

1. **Normalization:** If the left-hand-side (LHS) (cause part) of a qualitative specification contains a boolean subexpression defined on Y (e.g., $L_1^c(t, \bar{Y}(t))$), find another qualitative specification which contains the same boolean subexpression on its right-hand-side (RHS) (effect part) and substitute the LHS of this specification (e.g., $L_1^c(t, X(t))$) in the LHS of the original specification where Y appears; e.g.,

   original form:

$$\begin{array}{rcl} & L_1^c(t, X(t)) & \longrightarrow \; L_1^e(t, \bar{Y}(t)) \\ L_2^c(t, X(t)) \;\wedge\; & L_1^c(t, \bar{Y}(t)) & \longrightarrow \; L_2^e(t, \bar{Y}(t)) \end{array}$$

   after refinement:

$$\begin{array}{rcl} & L_1^c(t, X(t)) & \longrightarrow \; L_1^e(t, \bar{Y}(t)) \\ L_2^c(t, X(t)) \;\wedge\; & L_1^c(t, X(t)) & \longrightarrow \; L_2^e(t, \bar{Y}(t)) \end{array}$$

   This type of substitution is possible only if a cause part which involves Y is traceable; i.e., appears as an effect part in some other specifications whose cause part is defined on X (but not on Y). If the elimination of Y by substitution is not possible then such specifications must be flagged for later consideration since there is no clear basis for designing experiments that trigger these specifications (apart from trial and error).

2. **Composition:** If there is more than one qualitative specification with the same LHS (cause part), combine them into one specification uniting their RHS's

(effect parts) with the conjunction operator ($\wedge$); e.g.,

original form:

$$\begin{array}{l} L^c(t, X(t)) \longrightarrow L_1^e(t, \bar{Y}(t)) \\ L^c(t, X(t)) \longrightarrow L_2^e(t, \bar{Y}(t)) \\ \vdots \\ L^c(t, X(t)) \longrightarrow L_n^e(t, \bar{Y}(t)) \end{array}$$

after refinement:

$$L^c(t, X(t)) \longrightarrow L_1^e(t, \bar{Y}(t)) \wedge L_2^e(t, \bar{Y}(t)) \wedge \cdots \wedge L_n^e(t, \bar{Y}(t))$$

3. **Decomposition:** If the LHS (cause part) of a qualitative specification contains a series of boolean subexpressions separated by the disjunction operator ($\vee$), then generate a sequence of rules which have these individual subexpressions as their cause parts; e.g.,

original form:

$$L_1^c(t, X(t), \bar{Y}(t)) \vee L_2^c(t, X(t), \bar{Y}(t)) \vee \cdots \vee L_n^c(t, X(t), \bar{Y}(t)) \longrightarrow L^e(t, X(t), \bar{Y}(t))$$

after refinement:

$$\begin{array}{l} L_1^c(t, X(t), \bar{Y}(t)) \longrightarrow L^e(t, X(t), \bar{Y}(t)) \\ L_2^c(t, X(t), \bar{Y}(t)) \longrightarrow L^e(t, X(t), \bar{Y}(t)) \\ \vdots \\ L_n^c(t, X(t), \bar{Y}(t)) \longrightarrow L^e(t, X(t), \bar{Y}(t)) \end{array}$$

4. **Validation:** The concern here is with ensuring the integrity of the validation reference information. Some possible anomalies are given below:

   (a) **Infeasible** i.e., the specification of a causal relationship from output to input; e.g., $L^c(t, \bar{Y}(t)) \longrightarrow L^e(t, X(t))$

(b) **Conflicting** i.e., there are two specifications with the same LHS (RHS) and contradictory RHS (LHS); e.g.,

$$\begin{array}{lll} L^c(t, X(t)) & \longrightarrow & L^e(t, \bar{Y}(t)) \\ L^c(t, X(t)) & \longrightarrow & \neg\ L^e(t, \bar{Y}(t)) \end{array}$$

(c) **Circular** i.e., chaining some set of specifications forms a cycle; e.g.,

$$\begin{array}{lll} L^c(t, \bar{Y}(t)) & \longrightarrow & L^e_1(t, \bar{Y}(t)) \\ L^e_1(t, \bar{Y}(t)) & \longrightarrow & L^e_2(t, \bar{Y}(t)) \\ L^e_2(t, \bar{Y}(t)) & \longrightarrow & L^c(t, \bar{Y}(t)) \end{array}$$

(d) **Subsumed** i.e., there are two specifications with the same RHS but one specification has an additional cause; e.g.,

$$\begin{array}{llll} L^c_1(t, X(t)) & \wedge\ L^c_2(t, X(t)) & \longrightarrow & L^e(t, \bar{Y}(t)) \\ L^c_2(t, X(t)) & & \longrightarrow & L^e(t, \bar{Y}(t)) \end{array}$$

2.3 Global Architecture for a Validation Software Environment

The concepts formulated in this chapter provide the basis for a global architecture for a software environment suitable for automating the validation of a simulation model. The knowledge base outlined in the preceding section plays a central role in this environment because it provides the basis for dealing with two fundamental aspects of the validation problem; namely, designing experiments and evaluating the results of experiments. Experiments to be carried out with a simulation model can be automatically designed from the knowledge base and this is the focus of our discussion in the subsequent chapters. The result of the execution of a simulation model (i.e., observed behaviour) for a given experiment can be compared to the expected behaviour captured in the knowledge base.

As shown in Figure 2.1, this environment consists of four modules; namely,

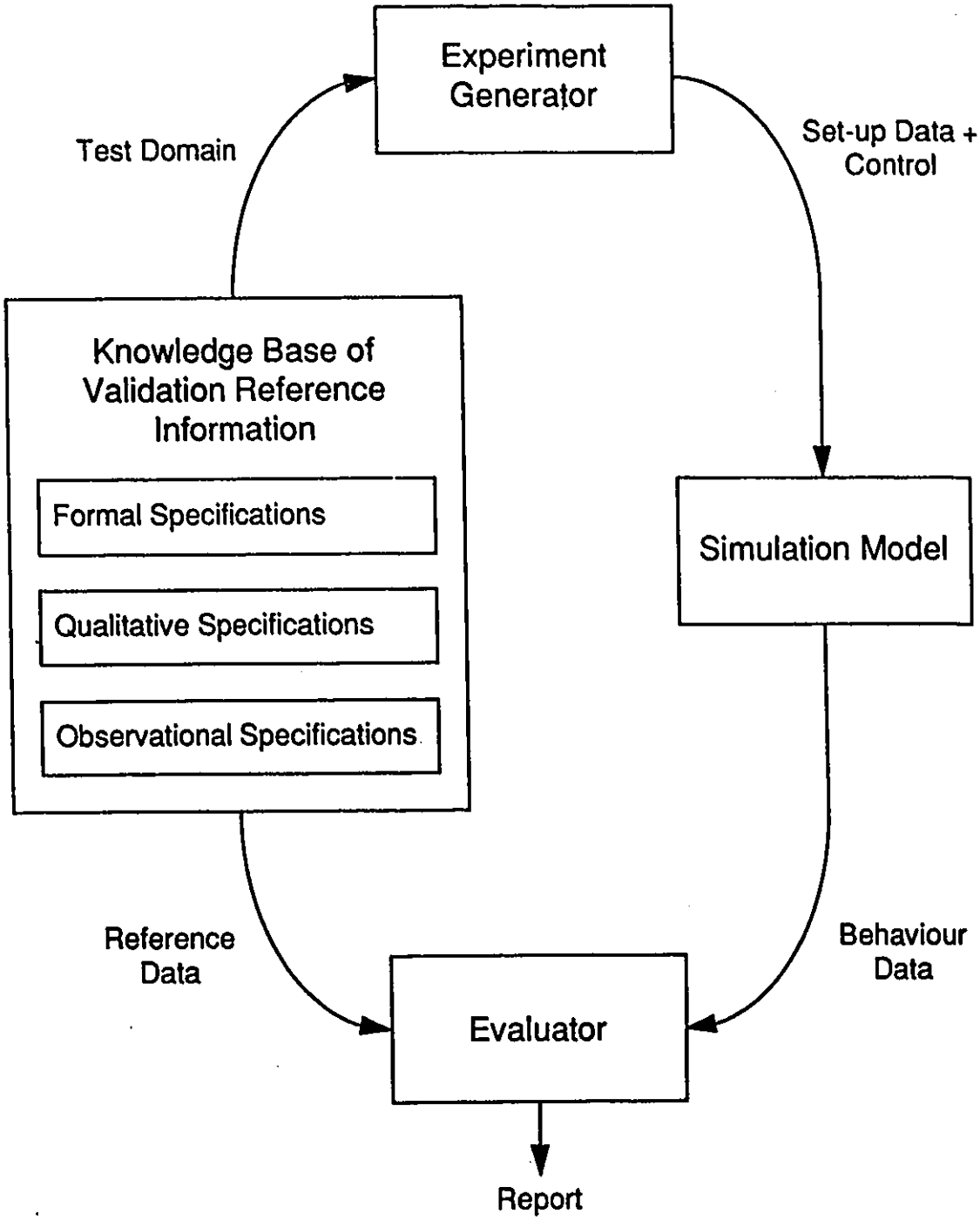

Figure 2.1. Global Architecture

i) The Simulation Model

ii) Knowledge Base of Validation Reference Information

iii) Experiment Generator

iv) Evaluator

The functions of these modules and their interrelationships are outlined below.

i) The Simulation Model

This module consists entirely of the simulation model which is under investigation. Its construction generally evolves through several stages as discussed in chapter 1. The model exists in the form of a computer program written in some programming language.

ii) Knowledge Base of Validation Reference Information

This module contains validation reference information for the simulation model. It is in the form of specifications provided by a domain expert and/or in the form of data obtained from systems and/or models which are known to correctly represent the expected behaviour of the simulation model under investigation. It contributes to two key functions; namely, experiment design and evaluation of simulation output.

iii) Experiment Generator

Test-cases (experiments) are designed from the specifications given in the Knowledge Base of Validation Reference Information by the Experiment Generator. These experiments are carried out with the simulation model to obtain behaviour data that is pertinent to the validation process.

It is possible to design a set of experiments such that each experiment simply tests one qualitative specification or one observational specification in conjunction with formal specifications. Such a strategy could be extremely inefficient because of the number of times the simulation model needs to be executed. Therefore, an important practical requirement for the Experiment Generator is to design a set of experiments in a way which efficiently utilizes the available validation reference information. That is; the objective of the Experiment Generator is to solve the experiment design problem as formulated in section 2.1.3.

Experiment design lies at the heart of any software environment for simulation model validation. The experiment design problem is a difficult problem because of its implicit minimization objective. Because of its importance, the principal concern of the remaining discussion in this thesis focuses on the experiment design problem.

iv) Evaluator

The final and essential step in the validation process is the critical evaluation of the simulation model output. The Evaluator realizes this function by checking the behaviour of the simulation model against the specifications within the Knowledge Base of Validation Reference Information. This process, in particular, utilizes the behavioural features of the formal specifications and the effect parts of all other specifications. Inconsistencies indicate an invalid model and appropriate reporting mechanisms identify the nature of the inconsistency.

Chapter 3

The Experiment Design Problem in a Constraint Set Framework

We pointed out in chapter 2 the importance of the validation reference information in the solution to the simulation model validation problem. It serves two functions; namely, designing experiments and evaluating the results of experiments. We also stressed the importance of designing experiments in an effective and efficient way, and formulated the experiment design problem. In this and the subsequent chapters, we investigate the experiment design problem (section 2.1.3) in a constraint set framework. We refer to this version of the problem as the C^3 problem. Specifically in section 3.1, we formulate the C^3 problem. In section 3.2, we show how to derive, from validation reference information, three constraint sets around which the C^3 problem can be defined. Furthermore because of the origins of these constraints the solution of the associated C^3 problem corresponds to a solution to the experiment design problem which is discussed in section 3.3.

3.1 Definition of the C^3 Problem

We begin by introducing some definitions that are important to the subsequent discussions. These are formulated around the following two sets of constraints:

$$S = \{h_i(X)\mathbf{R}_i 0, i = 1, 2, \cdots, m\}$$

$$S_e = \{\lambda_{a_i}(X, X', \varepsilon_i) \wedge h_i(X)\mathbf{R}_i 0 \wedge h_i(X')\mathbf{R}_i 0, i = m+1, m+2, \cdots, p; a_i \in I_n\}$$

with $X, X' \in \hat{\Re}_n \subset \Re_n$, $\mathbf{R}_i \in \{=, \leq, \geq\}$ and $h_i : \hat{\Re}_n \longrightarrow \Re$. We assume that, for all i, $h_i(X)$ is bounded and differentiable for all $X \in \hat{\Re}_n$. The predicate $\lambda_{a_i}(X^k, X^l, \varepsilon_i)$ has a TRUE value at the pair (X^k, X^l) if and only if $X^l_j = X^k_j$ for every $j \neq a_i$ and $X^l_{a_i} \geq X^k_{a_i} + \varepsilon_i$ where $\varepsilon_i > 0$. Throughout the sequel it will always be understood that $a_i \in I_n$.

a) The constraint $h_i(X)\mathbf{R}_i 0 \in S$ is satisfied at $X^* \in \hat{\Re}_n$ if $[h_i(X)]_{X=X^*}\mathbf{R}_i 0$.

b) Let $\hat{S}$ be a non-empty subset of S and let $\Phi = \{X \in \hat{\Re}_n$: every constraint in $\hat{S}$ is satisfied at $X\}$. $\hat{S}$ is said to be consistent with respect to $\hat{\Re}_n$ if $\Phi \neq \emptyset$; otherwise it is inconsistent with respect to $\hat{\Re}_n$. If $\Phi \neq \emptyset$ then $\hat{S}$ is said to be consistent at each $X \in \Phi$.

c) The perturbation constraint $\lambda_{a_i}(X, X', \varepsilon_i) \wedge h_i(X)\mathbf{R}_i 0 \wedge h_i(X')\mathbf{R}_i 0 \in S_e$ is satisfied at the pair $(X^k, X^l) \in \hat{\Re}_n \times \hat{\Re}_n$ if and only if

   i) $\lambda_{a_i}(X^k, X^l, \varepsilon_i)$,

   ii) $[h_i(X)]_{X=X^k}\mathbf{R}_i 0$ and

   iii) $[h_i(X')]_{X'=X^l}\mathbf{R}_i 0$.

d) Let $\hat{S}_e \subseteq S_e$ and $\hat{S} \subseteq S$ with $\hat{S} \bigcup \hat{S}_e \neq \emptyset$. A feasible bag for $\hat{S} \bigcup \hat{S}_e$ is a bag[1] of n - vectors $\{X^*\} \bigcup \{X^i : i = 1, 2, \cdots, | \hat{S}_e |\}$ with $X^* \in \hat{\Re}_n$ and $X^i \in \hat{\Re}_n$ for each i such that

i) $\hat{S}$ is consistent at X^* and at each X^i, and

ii) for every $c_i \in \hat{S}_e$ c_i is satisfied at the pair (X^*, X^i).

A set of constraints $\hat{S} \bigcup \hat{S}_e$ is consistent with respect to $\hat{\Re}_n$ if it has a non-empty feasible bag; otherwise it is inconsistent with respect to $\hat{\Re}_n$. Furthermore, $\hat{S} \bigcup \hat{S}_e$ is said to be consistent at any one of its feasible bags. An E_set of $\hat{S} \bigcup \hat{S}_e$, denoted by $E\_set(\hat{S} \bigcup \hat{S}_e)$, is a feasible bag with duplicate entries removed.

e) A non-empty subset $\hat{S}$ of $S \bigcup S_e$ can be defined using a non-empty index set $U_s \subseteq I_p$. We use $G(\hat{S})$ to denote the set of indices that are associated with $\hat{S}$; i.e., $G(\hat{S}) = U_s$. More precisely, let $\wp(S \bigcup S_e)$ be the power set of $S \bigcup S_e$, then G is a one-to-one mapping from $\wp(S \bigcup S_e) - \{\emptyset\}$ to $\wp(I_p) - \{\emptyset\}$; i.e., $G : \wp(S \bigcup S_e) - \{\emptyset\} \longrightarrow \wp(I_p) - \{\emptyset\}$. The inverse of G is defined in the obvious way; in particular, $G^{-1}(G(\hat{S})) = \hat{S}$.

More general definitions for the consistency of constraint sets are provided in Appendix B. The analysis of the C^3 problem will however be based on the above definitions in order to facilitate the presentation.

We now introduce the concept of a consistent cover of a set of constraints. This is carried out in the context of the following sets of constraints:

[1]Recall that a bag (multiset) is a set which allows the occurance of duplicate entries.

$$S_f = \left\{ \begin{array}{lll} f_i(X) \leq 0 & , & i = 1, 2, \cdots, l' \text{ and} \\ \bar{f}_i(X) = 0 & , & i = l'+1, l'+2, \cdots, l \end{array} \right\}$$

$$S_c = \left\{ \begin{array}{lll} g_i(X) \leq 0 & , & i = l+1, l+2, \cdots, m' \text{ and} \\ \bar{g}_i(X) = 0 & , & i = m'+1, m'+2, \cdots, m \end{array} \right\}$$

$$S_v = \left\{ \begin{array}{llll} \lambda_{a_i}(X, X', \epsilon_i) & \wedge & v_i(X) \leq 0 \wedge v_i(X') \leq 0 & , \quad i = m+1, m+2, \cdots, p' \text{ and} \\ \lambda_{a_i}(X, X', \epsilon_i) & \wedge & \bar{v}_i(X) = 0 \wedge \bar{v}_i(X') = 0 & , \quad i = p'+1, p'+2, \cdots, p;\ a_i \in I_n \end{array} \right\}$$

with $X, X' \in \hat{\Re}_n \subset \Re_n$. Furthermore, we assume that, for all i, $f_i(X)$, $\bar{f}_i(X)$, $g_i(X)$, $\bar{g}_i(X)$, $v_i(X)$ and $\bar{v}_i(X)$ are bounded for all $X \in \hat{\Re}_n$.

Let $\Phi_f = \{X \in \hat{\Re}_n : S_f$ is consistent at $X\}$ and assume that for every $c \in S_c \bigcup S_v$, the set $S_f \bigcup \{c\}$ is consistent. This property of the sets S_f, S_c and S_v is referred to as property Υ. (Note that property Υ ensures that $\Phi_f \neq \emptyset$.) Within the framework of this assumption, we define a *consistent cover*, E, of $S_f \bigcup S_c \bigcup S_v$ with respect to $\hat{\Re}_n$[2] is a non-empty set of subsets of $S_f \bigcup S_c \bigcup S_v$ with the following three properties:

1. each member of E contains S_f,

2. each member of E is consistent with respect to $\hat{\Re}_n$[2]

3. each constraint in $S_c \bigcup S_v$ appears in at least one member of E.

Observe that property Υ assures the existence of a consistent cover. We denote by $Min(S_f \bigcup S_c \bigcup S_v)$ the cardinality of a minimal consistent cover of $S_f \bigcup S_c \bigcup S_v$, i.e., there exists no consistent cover of $S_f \bigcup S_c \bigcup S_v$ with cardinality less than $Min(S_f \bigcup S_c \bigcup S_v)$. Note that $Min(S_f \bigcup S_c \bigcup S_v) \leq \mid S_c \bigcup S_v \mid$. Also note that each member, e_i, of a consistent cover, E, has a feasible bag and hence has an E_set, given by $E\_set(e_i)$, associated with it. Then, the E_number of E, denoted by $E\_number(E)$, is defined as:

[2]In the subsequent discussion the phrase "with respect to $\hat{\Re}_n$" will be dropped to simplify the presentation.

$$E\_number(E) = \left| \bigcup_{i=1}^{|E|} E\_set(e_i) \right|$$

The C^3 problem is the problem of finding a minimal consistent cover, E^*, of $S_f \bigcup S_c \bigcup S_v$ subject to the constraint that for every $e_i \in E^*$, e_i is maximum. (We define "maximum" to mean that for every $e_i \in E^*$ there does not exist a non-empty subset $\hat{S}_i$ of $S_c \bigcup S_v - e_i$ such that $e_i \bigcup \hat{S}_i$ is consistent.)

We examine the C^3 problem in an order of increasing complexity. That is,

1. $S_f = \emptyset$, $S_c \neq \emptyset$, and $S_v = \emptyset$

2. $S_f \neq \emptyset$, $S_c \neq \emptyset$, and $S_v = \emptyset$

3. $S_f \neq \emptyset$, $S_c \neq \emptyset$ and $S_v \neq \emptyset$

In chapter 4, we analyze the complexity of the C^3 problem. We show that a simplified version of the C^3 problem (i.e., when $S_f = \emptyset$, $S_v = \emptyset$ and in the absence of the constraint that each element of a minimal consistent cover be maximum) is a difficult problem. Specifically, we develop a graph theoretical characterization of the simplified version of the C^3 problem and establish its relation to both the edge cover problem and the clique cover problem. Correspondingly, a heuristic approach for the solution of the C^3 problem is developed in chapter 5 when $S_f \neq \emptyset$, $S_c \neq \emptyset$ and $S_v = \emptyset$. In chapter 6, we develop a formalism for extending the heuristic solution of the C^3 problem when $S_v \neq \emptyset$.

3.2 Derivation of Constraint Sets from Validation Reference Information

The C^3 problem is formulated around three sets of constraints denoted by S_f, S_c and S_v. Our goal in this section is to show how the validation reference information relating to the dynamic object $O = (X, Y)$ can give rise to the three sets implicit in the definition of the C^3 problem. This goal is achieved via the following steps which presume that preprocessing of the knowledge base as discussed in section 2.2.2 has already taken place.

1. Recall that in chapter 2, we defined the validation reference information as a collection of three disjoint sets of relationships which are:

   - $FS(O)$: a set of formal specifications where each member of $FS(O)$ is either of the form $L(t, X(t))$ or of the form $L(t, X(t), Y(t))$.
   - $QS(O)$: a set of qualitative specifications where each member of $QS(O)$ is either of the form; $L^c(t, X(t)) \longrightarrow L^c(t, X(t), \bar{Y}(t))$ or $L^{cc}(t, X(t), X'(t), \epsilon) \longrightarrow L^{cc}(t, \bar{Y}(t), \bar{Y}'(t), \epsilon)$.
   - $OS(O)$: a set of observational specifications where each member of $OS(O)$ is of the form: $L^{co}(X, \tilde{\tilde{X}}) \longrightarrow L^{co}(\bar{Y}, \tilde{\tilde{Y}}, \epsilon)$

   Without loss of generality, we can index each member of the sets $FS(O)$, $QS(O)$ and $OS(O)$ as follows:

$$FS(O) = \{L_i(t, X(t)), i = 1, 2, \cdots q_1'\} \bigcup \{L_i(t, X(t), Y(t)), i = q_1' + 1, \cdots q_2'\}$$

$$QS(O) = \{L_i^c(t, X(t)) \longrightarrow L_i^c(t, X(t), \bar{Y}(t)), i = q_2' + 1, \cdots q_3'\}$$

$$\bigcup \{L_i^{cc}(t, X(t), X'(t), \epsilon_i) \longrightarrow L_i^{cc}(t, \bar{Y}(t), \bar{Y}'(t), \epsilon_i), i = q_3' + 1, \cdots q_4'\}$$

$$OS(O) = \{L_i^{co}(X, \overset{\approx}{X}^i) \longrightarrow L_i^{co}(\bar{Y}, \overset{\approx}{Y}^i, \epsilon_i), i = q_4' + 1, \cdots q_5'\}$$

2. Since the C^3 problem is concerned with the experiment design problem, only the test domain (i.e., the input features of $FS(O)$, $L_i(t, X(t)), i = 1, 2, \cdots q_1'$, and the cause parts of each member of $QS(O)$ and $OS(O)$) is of interest. Note that the behavioural features of $FS(O)$ (i.e., $L_i(t, X(t), Y(t)), i = q_1' + 1, \cdots q_2'$) and the effect parts of the members of $QS(O)$ and $OS(O)$ are required in the evaluation of the results of the experiments.

   Notice that the specifications above depend on t which is the general case. At this point with the interest in clarifying the underlying concept, we restrict our further consideration to the case where the input vector to the object is not time-dependent; i.e., simply is a collection of initial states and parameters. The input information for the experiment design problem thus consists of three disjoint sets of boolean expressions; namely:

$$F = \{L_i(X), i = 1, 2, \cdots q_1\}$$

$$Q = \{L_i^c(X), i = q_1 + 1, \cdots q_2\} \bigcup \{L_i^{cc}(X, X', \epsilon_i), i = q_2 + 1, \cdots q_3\}$$

$$O = \{L_i^{co}(X, \overset{\approx}{X}^i), i = q_3 + 1, \cdots q_4\}$$

where $L_i^{cc}(X, X', \epsilon_i) = \lambda_{a_i}(X, X', \epsilon_i) \wedge L_i^{cv}(X) \wedge L_i^{cv}(X')$, $q_1 = q_1'$, $q_2 - q_1 = q_3' - q_2'$, $q_3 - q_2 = q_4' - q_3'$ and $q_4 - q_3 = q_5' - q_4'$.

3. Notice that the elements of $\{L_i^c(X), i = q_1 + 1, \cdots q_2\} \subseteq Q$ and the elements of O reference a single experiment whereas the elements of $\{L_i^{cc}(X, X', \varepsilon_i), i = q_2 + 1, \cdots q_3\} \subseteq Q$ reference a pair of experiments. Also recall that the elements of F express those relationships on the input vector of a simulation model which must hold in any experiment carried out with the model. Therefore, we reassemble the sets F, Q and O into three new disjoint sets; namely,

   - S'_f, a set whose elements must hold in every experiment
   - S'_c, a set whose elements reference a single experiment
   - S'_v, a set whose elements reference a pair of experiments

   After renaming indices, we have:

$$S'_f = \{L_i(X), i = 1, 2, \cdots r_1\}$$

$$S'_c = \{L_i^c(X), i = r_1 + 1, \cdots r_2\} \bigcup \{L_i^{co}(X, \tilde{\tilde{X}}^i), i = r_2 + 1, \cdots r_3\}$$

$$S'_v = \{\lambda_{a_i}(X, X', \varepsilon_i) \wedge L_i^{cv}(X) \wedge L_i^{cv}(X'), i = r_3 + 1, \cdots r_4;\ a_i \in I_n\}$$

   where $r_1 = q_1$, $r_2 = q_2$, $r_3 - r_2 = q_4 - q_3$ and $r_4 - r_3 = q_3 - q_2$.

4. Recall that $L_i(\ )$ is a boolean expression defined over its argument and, as a result of the decomposition phase in the refinement activity (see section 2.2.2), it does not contain the disjunction operator ($\vee$). Hence, $L_i(\ )$ can be a series of inequality and equality constraints separated by the conjunction operator ($\wedge$); i.e.,

$$L_i(X) = \bigwedge_{w=1}^{d_i} [h_{iw}(X)\mathbf{R}_{iw}0] \text{ where } \mathbf{R}_{iw} \in \{=, \leq, \geq\}.$$

Note that the same observation applies to $L_i^c(\ )$ and $L_i^{cv}(\ )$.

Since every element of S_f' must hold in every experiment then $L_1(X) \wedge L_2(X) \wedge \cdots \wedge L_{r_1}(X)$ must be TRUE in every experiment. That is, $\wedge_{i=1}^{r_1}[\wedge_{w=1}^{d_i}[h_{iw}(X)\mathbf{R}_{iw}0]]$ must be TRUE in every experiment. Consequently S_f' can be rewritten as:

$$S_f = \left\{ \begin{array}{lll} f_i(X) \leq 0 & , & i = 1, 2, \cdots, l' \text{ and} \\ \bar{f}_i(X) = 0 & , & i = l'+1, l'+2, \cdots, l \end{array} \right\}$$

where $l = d_1 + d_2 + \cdots + d_{r_1}$, which is a set whose elements must hold in every experiment.

Also recall that $L_i^{co}(X, \tilde{\tilde{X}}^i)$ is defined as $X - \tilde{\tilde{X}}^i = 0$. Note that $X - \tilde{\tilde{X}}^i = 0$ is a conjunction of n equality constraints; i.e., $\wedge_{w=1}^{n}[X_w - \tilde{\tilde{X}}_w^i = 0]$. Consequently S_c' can be rewritten as:

$$S_c^d = \{ \bigwedge_{w=1}^{d_i} [g_{iw}(X)\mathbf{R}_{iw}0], i = l+1, l+2, \cdots, m\}$$

where $m - l = r_3 - r_1$, which is a set whose elements reference a single experiment.

Similarly the set S_v' becomes:

$$S_v^d = \{\lambda_{a_i}(X, X', \varepsilon_i) \bigwedge_{w=1}^{d_i} [v_{iw}(X)\mathbf{R}_{iw}0 \wedge v_{iw}(X')\mathbf{R}_{iw}0], i = m+1, \cdots, p;\ a_i \in I_n\}$$

where $p - m = r_4 - r_3$, which is a set whose elements reference a pair of experiments.

In the interest of simplifying our analysis we take $d_i = 1$ for every $i \in [l+1, p]$. Thus sets S_c^d and S_v^d become respectively S_c and S_v, and are given below:

$$S_c = \left\{ \begin{array}{lll} g_i(X) \leq 0 & , & i = l+1, l+2, \cdots, m' \text{ and} \\ \bar{g}_i(X) = 0 & , & i = m'+1, m'+2, \cdots, m \end{array} \right\}$$

$$S_v = \left\{ \begin{array}{lcll} \lambda_{a_i}(X, X', \varepsilon_i) & \wedge & v_i(X) \leq 0 \wedge v_i(X') \leq 0 & , \quad i = m+1, \cdots, p' \text{ and} \\ \lambda_{a_i}(X, X', \varepsilon_i) & \wedge & \bar{v}_i(X) = 0 \wedge \bar{v}_i(X') = 0 & , \quad i = p'+1, \cdots, p; \; a_i \in I_n \end{array} \right\}$$

In order to establish a meaningful C^3 problem with the above sets it is essential that they have property Υ (section 3.1). The assurance of this property is taken to be another task of the preprocessing step that converts the knowledge base into the validation reference information.

In the subsequent chapters, our analysis of the C^3 problem as well as a mechanism for its solution is presented in terms of the above three constraint sets S_f, S_c and S_v. We note furthermore that our proposed solution method (section 5.1) is sufficiently powerful to accommodate the more general constraints sets S_c^d and S_v^d.

3.3 Relationship between the C^3 Problem and the Experiment Design Problem

From the procedure in section 3.2 which was used to extract the constraint sets S_f, S_c and S_v from the validation reference information, and from the definition of the C^3 problem (section 3.1), it can be observed that solving the C^3 problem with these constraint sets corresponds to solving the experiment design problem (section 2.1.3). In fact, the C^3 problem represents a formalization of the experiment design problem.

Let E^* be a solution to the C^3 problem which is associated with the sets S_f, S_c and S_v. We discuss below how E^* corresponds to a solution to the experiment design problem.

The set S_f corresponds to the set of all input features. Therefore, all input features are satisfied because for every $e \in E^*$ e is consistent and e contains S_f.

The set $S_c \bigcup S_v$ corresponds to $\Omega = QS(O) \bigcup OS(O)$. Therefore, all specifications in Ω are covered because for every $e \in E^*$ is consistent, and for every $c \in S_c \bigcup S_v$ there exists $e \in E^*$ such that $c \in e$.

Notice also that each $e \in E^*$ gives rise to one or more input specifications each of which contributes an experiment to the solution of the experiment design problem. Furthermore, $E\_number(E^*) = \mid \Omega^*_{EXP} \mid$; i.e., the total number of experiments for the experiment design problem.

Chapter 4

The Complexity of the C^3 Problem

In this chapter we analyze the C^3 problem in its simplest form (i.e., $S_f = \emptyset$, $S_v = \emptyset$ and in the absence of the constraint that each element of a minimal consistent cover be maximum). We call this specialized case, $\hat{C}^3$ problem. We approach the $\hat{C}^3$ problem from a graph-theoretical perspective. Representing the $\hat{C}^3$ problem as a graph facilitates the application of numerous results from Graph Theory and also helps characterize this problem. In section 4.1, we define the concept of a *constraint set graph* and a *consistent constraint set graph*. We establish a relation between the *maximum independent set* of a constraint set graph and $Min(S_c)$. In section 4.2.1, we show that the $\hat{C}^3$ problem is related to two well known problems in Graph Theory; namely, the edge cover problem and the clique cover problem. In section 4.2.2, we present a formulation of the $\hat{C}^3$ problem as a set-covering problem.

4.1 Graph-Theoretical Characterization

We introduce a new concept called the constraint set graph, compute the cost of constructing such a graph and establish a relation between the maximum independent

set of a constraint set graph and $Min(S_c)$.

Definition 4.1: The *Constraint Set Graph* $G_c = (V_c, E_c)$ for a set, S_c, of constraints is an undirected graph where $\mid V_c \mid = \mid S_c \mid = m$ and each vertex $v_i; i = 1, \cdots, m$ corresponds to one of the constraints in the constraint set S_c; i.e., $S_c \equiv V_c$. The edge set E_c is defined as follows:

$$E_c = \{\{v_i, v_j\} : \text{ there exists } X \in \hat{\Re}_n \text{ such that } v_i \text{ and } v_j \text{ are consistent at } X\}.$$

Note that G_c is completely connected (i.e., complete) if all constraints are pairwise consistent. Observe that any member of a consistent cover of S_c must correspond to a complete subgraph (i.e., clique) of G_c. However, a complete subgraph of G_c does not necessarily correspond to a consistent set of constraints.

Recall that the adjacency matrix of any undirected graph G, $G = (V, E)$, denoted by $A(G) = [a_{ij}]$, is the $\mid V \mid \times \mid V \mid$ matrix where

$$a_{ij} = \begin{cases} 1 & \text{if } v_i \text{ and } v_j \text{ are joined by an edge} \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 4.1: Let S_c be a set of constraints and let $\mid S_c \mid = m$. Let ρ_{ij} be the computational cost of deciding whether constraints i and j in S_c are consistent or not. Let $\tau = \max\{\rho_{ij}\}$. Then, the computational cost of characterizing the constraint set graph for S_c by constructing its adjacency matrix is $\frac{m(m-1)}{2}\tau$.

Proof: From the definition of the constraint set graph, there exists an edge $\{v_i, v_j\}$ between vertices v_i and v_j if and only if the constraints i and j in S_c are consistent. Therefore, the total number of pairs of constraints (i, j) for $i, j \in I_m$, to be examined for consistency are easily obtained by the following representation of all possible pairs of constraints.

$$
\begin{array}{llll}
(1,2) & (2,3) & \cdots & (m-2,m-1) \quad (m-1,m) \\
(1,3) & (2,4) & \cdots & (m-2,m) \\
\vdots & \vdots & \cdots & \\
(1,m-1) & (2,m) & & \\
(1,m) & & &
\end{array}
$$

The number of elements in this array is $(m-1)+(m-2)+\cdots+2+1=\frac{m(m-1)}{2}$. Therefore, the computational cost of constructing the constraint set graph for S_c is $\frac{m(m-1)}{2}\tau$. □

Definition 4.2: Independent set of a graph [38]

Let $G=(V,E)$ be an undirected graph. A subset $V' \subseteq V$ is called the *independent set* of G if $\forall u,v \in V'$, $\{u,v\} \notin E$. That is, no pair of vertices in V' is joined by an edge.

An independent set V' is *maximum* if G has no independent set V'' with $\mid V'' \mid > \mid V' \mid$. Associated with a maximum independent set of a graph is the decision problem called the independent set problem which is defined below.

Independent Set Problem [38]

INSTANCE: Graph $G=(V,E)$, positive integer $K \leq \mid V \mid$.

QUESTION: Does G contain an independent set, V', of size K or more, i.e., a subset $V' \subseteq V$ such that $\mid V' \mid \geq K$?

The *independent set* problem is known to be NP-Complete [38].

Theorem 4.2: Let $G_c=(V_c,E_c)$ be the constraint set graph associated with the set of constraints S_c. Let $V' \subseteq V_c$ be a maximum independent set of G_c. Then, the minimum size of any consistent cover of S_c is bounded from below by $\mid V' \mid$; i.e., $Min(S_c) \geq \mid V' \mid$.

Proof: From the definition of the maximum independent set of G_c, there is no arc between any pair of vertices in this set. This implies that every pair of vertices in V' is inconsistent and hence no pair can occur together within any member of the set of subsets of S_c which we seek to construct. Consequently the number of such subsets can not be less than $\mid V' \mid$. □

Theorem 4.3: Let $G_c = (V_c, E_c)$ be the constraint set graph associated with the set of constraints S_c. If G_c is completely connected with $\mid V_c \mid = m$, then $Min(S_c) \leq \lceil \frac{m}{2} \rceil = \overline{m}$.

Proof: Since G_c is completely connected, all constraints are pairwise consistent. A consistent cover of S_c with cardinality $\overline{m}$ is as follows:

If $\mid V_c \mid$ = odd then $\{v_1, v_2\}, \{v_3, v_4\}, \cdots, \{v_{m-2}, v_{m-1}\}, \{v_m\}$.

If $\mid V_c \mid$ = even then $\{v_1, v_2\}, \{v_3, v_4\}, \cdots, \{v_{m-1}, v_m\}$. □

Two other obvious cases relating $Min(S_c)$ to $\mid V' \mid$ are when

a) S_c is consistent in which case $Min(S_c) = \mid V' \mid = 1$ and

b) every pair of constraints in S_c is inconsistent in which case $Min(S_c) = \mid V' \mid = m$.

To facilitate the analysis of the $\hat{C}^3$ problem which follows in the next section, we define below a special case of a constraint set graph called a consistent constraint set graph.

Definition 4.3: The *Consistent Constraint Set Graph*, $G_{cc} = (V_{cc}, E_{cc})$ for a given set of constraints is a constraint set graph for which every clique (i.e., complete subgraph) of size 3 or more is consistent.

4.2 Analysis of the $\hat{C}^3$ Problem

4.2.1 Graph-Theoretic Perspective

In this section, we present the analysis of the $\hat{C}^3$ problem which shows that the $\hat{C}^3$ problem is related to both the edge cover problem and the clique cover problem. Note that the edge cover problem can be solved in polynomial-time [38] whereas the clique cover problem is known to be NP-Complete [38]. We show that the consistency of the constraint set graph is the key in deciding whether the $\hat{C}^3$ problem is related to the edge cover problem or to the clique cover problem.

Observe that from the definition of the constraint set graph, any clique of size greater than or equal to 3 may or may not be consistent. This is because a constraint set graph unlike an undirected graph has an additional consistency semantics associated with it. In order to deduce the consistency of the cliques of size greater than or equal to 3, one needs to solve the constraints corresponding to these cliques. Hence there are four interesting cases in the analysis of the $\hat{C}^3$ problem that can be identified:

a) There are no cliques of size greater than or equal to 3,

b) All cliques of size greater than or equal to 3 are inconsistent,

c) All cliques of size greater than or equal to 3 are consistent and
(consistent constraint set graph)

d) Some cliques of size greater than or equal to 3 are consistent while others are not

In the following discussion we first show that in cases (a) and (b) the $\hat{C}^3$ problem is related to the edge cover problem while in case (c) it is related to the clique cover problem. In case (d), it is an open question as to whether it is related to any known problem of Graph Theory.

The $\hat{C}^3$ problem finds the minimum consistent cover of a set of constraints. Observe that in cases (a) and (b) the maximum size of any subset in a minimum consistent cover is 2, i.e., it corresponds to an edge in the constraint set graph. Therefore, the $\hat{C}^3$ problem reduces to the problem of finding the smallest set of edges, $E' \subseteq E_c$, of the constraint set graph, $G_c = (V_c, E_c)$, such that every vertex $v \in V_c$ belongs to at least one $e \in E'$. This is, in fact the edge cover problem which can be solved in polynomial-time[38]. The following analysis (i.e., Lemma 4.1 and Theorem 4.4) formally establishes that in cases (a) and (b) the $\hat{C}^3$ problem in constraint set graph framework is equivalent to the edge cover problem.

Lemma 4.1: Let E^e be a minimum edge cover of the constraint set graph $G_c = (V_c, E_c)$ associated with the set of constraints S_c and assume G_c is connected. Then E^e is a consistent cover of S_c.

Proof: To prove that E^e is a consistent cover of S_c, we must show that

i) $\forall e \in E^e$ [e is consistent]

ii) $\forall c \in S_c$ [$\exists e \in E^e : c \in e$]

For every $e \in E^e$, e is an edge of the constraint set graph G_c. From the definition of a constraint set graph, the pair of constraints associated with edge e of G_c are consistent. This proves property (i).

From the definition of the constraint set graph $G_c = (V_c, E_c)$, $V_c \equiv S_c$. Since E^e is a minimum edge cover of G_c then $\forall c \in V_c \quad \exists e \in E^e$ such that $c \in e$. This proves property (ii). □

Theorem 4.4: Let E^c be a minimum consistent cover of the set of constraints S_c and E^e be a minimum edge cover of the constraint set graph $G_c = (V_c, E_c)$ associated with the set of constraints S_c and assume G_c is connected. Assume that either G_c does not contain cliques of size greater than or equal to 3, or all cliques of size greater than or equal to 3 in G_c are inconsistent. Then

1. E^e is a consistent cover of S_c and
2. $| E^c | \geq | E^e |$.

Proof: (1) From Lemma 4.1 E^e is a consistent cover of S_c.

(2) Observe that from the assumption, $| e | \leq 2$ for any $e \in E^c$ and therefore for any $e \in E^c$ either $| e | = 1$ or $| e | = 2$. Note also that the case where for all $e \in E^c$ have size 1 is not possible. This can be demonstrated in following way. Suppose $\forall e \in E^c \quad | e | = 1$ then this implies that $| E^c | = | S_c | = | V_c |$.

Let $S_c = \{v_1, v_2, \cdots, v_i, \cdots, v_{m-1}, v_m\}$. Then, $E^c = \{\{v_1\}, \{v_2\}, \cdots, \{v_i\}, \cdots, \{v_{m-1}\}, \{v_m\}\}$.

Since G_c is connected then for any $v_i \in S_c$ there exists $u \in S_c - \{v_i\}$ such that (u, v_i) is an edge of G_c. Without loss of generality, let $u = v_m$. Observe that $\{v_i, v_m\}$ is consistent because $\{v_i, v_m\}$ is an edge of G_c. Now, we can construct a consistent cover $\hat{E}^c$ as follows.

$\hat{E}^c = \{\{v_1\}, \{v_2\}, \cdots, \{v_i, v_m\}, \cdots, \{v_{m-1}\}\}$

Notice that $| \hat{E}^c | = | E^c | - 1$.

This is a contradiction ∴ E^c is a minimum consistent cover of S_c.

∴ The case where $\forall e \in E^c \quad | e | = 1$, is impossible.

There are two cases to consider:

Case (i): $\forall e \in E^c \quad | e | = 2$.

This implies that $\forall e \in E^c$ e is an edge of G_c. Furthermore $\forall c \in V_c \quad \exists e \in E^c$ such that $c \in e$. Hence E^c is an edge cover. Therefore, $| E^c | \geq | E^e |$.

Case (ii): $\exists e_1, e_2 \in E^c$ such that $| e_1 | = 1$ and $| e_2 | = 2$.

Let $\hat{E}^c = \{e_i \in E^c : \ | e_i | = 1\}$ and $\tilde{E}^c = \{e_i \in E^c : \ | e_i | = 2\}$. Then $| \hat{E}^c \cup \tilde{E}^c |$ $= | \hat{E}^c | + | \tilde{E}^c | = | E^c |$.

Since G_c is connected then for every $e = \{u\} \in \hat{E}^c$ there exists $v \in V_c$ such that $\{u, v\}$ is an edge of G_c. Thus the set $\hat{E}^c$ can be replaced by a set $\bar{E}^c$ with $| \bar{E}^c | = | \hat{E}^c |$ whose elements are the consistent sets $\{u, v\}$ with $\{u\} \in \hat{E}^c$.

Notice also that every element of $\bar{E}^c \cup \tilde{E}^c$ is an edge of G_c. Thus $\forall c \in V_c \quad \exists e \in \bar{E}^c \cup \tilde{E}^c$ such that $c \in e$. Hence $\bar{E}^c \cup \tilde{E}^c$ is an edge cover. Therefore, $| \bar{E}^c \cup \tilde{E}^c | \geq | E^c |$. But note that $| \bar{E}^c \cup \tilde{E}^c | = | E^c |$.

∴ $| E^c | \geq | E^e |$. □

The following analysis (i.e., Lemma 4.2 and Theorem 4.5) formally establishes that in case (c) the $\hat{C}^3$ problem in constraint set graph framework is equivalent to the clique cover problem.

Lemma 4.2: Let E^{cc} be a minimum clique cover of the consistent constraint set graph $G_c = (V_c, E_c)$ associated with the set of constraints S_c. Then E^{cc} is a consistent cover of S_c.

Proof: To prove that E^{cc} is a consistent cover of S_c, we must show that

i) $\forall e \in E^{cc}$ [e is consistent]

ii) $\forall c \in S_c$ $[\exists e \in E^{cc} : c \in e]$

For every $e \in E^{cc}$, e is a clique of the constraint set graph G_c. Observe that if $\mid e \mid = 1$, the constraint c associated with e is consistent because of property Υ (section 3.1). Observe also that if $\mid e \mid = 2$, e is an edge of the constraint set graph G_c. From the definition of a constraint set graph, the pair of constraints associated with edge e of G_c is consistent. Finally, by assumption, if $\mid e \mid \geq 3$, then e is consistent. Thus property (i) is proved.

Property (ii) follows from the definition of a clique cover and the observation that $V_c \equiv S_c$. □

Theorem 4.5: Let E^c be a minimum consistent cover of the set of constraints S_c and E^{cc} be a minimum clique cover of the consistent constraint set graph $G_c = (V_c, E_c)$ associated with the set of constraints S_c. Then

1. E^{cc} is a consistent cover of S_c and

2. $\mid E^c \mid \geq \mid E^{cc} \mid$.

Proof: (1) From Lemma 4.2 E^{cc} is a consistent cover of S_c.

(2) Since every $e \in E^c$ is consistent it follows directly that any pair of constraints in e is also consistent. Therefore, $\forall e \in E^c$ e corresponds to a clique in G_c.

Although it is true that every $c \in S_c$ belongs to some $e \in E^c$ because E^c is a minimum consistent cover, it may occur that c belongs to more than one $e \in E^c$. The

set E^c can however be easily replaced with $\mid E^c \mid = \mid \hat{E}^c \mid$ in which each constraint $c \in S_c$ belongs to a unique $e \in \hat{E}^c$ and for every $e \in \hat{E}^c$, e corresponds a clique in G_c. Thus for every $c \in V_c$ there exists exactly one $e \in \hat{E}^c$ such that $c \in e$. Hence, $\hat{E}^c$ is a clique cover of S_c. Therefore, we have that $\mid E^c \mid = \mid \hat{E}^c \mid \geq \mid E^{cc} \mid$. □

It is interesting to note that the $\hat{C}^3$ problem can be formulated as a set-covering problem which is discussed next.

4.2.2 A Set-Covering Formulation

Given a set $S = \{s_1, s_2, \cdots, s_m\}$ and a family $\mathcal{F} = \{S_1, S_2, \cdots, S_n\}$ of subsets $S_j \subseteq S$, any subfamily $\hat{\mathcal{F}}$ of $\mathcal{F}$ is called a *set-cover* of S if $\bigcup_{S_i \in \hat{\mathcal{F}}} S_i = S$. The *set-covering* problem is the problem of finding a set-cover of minimum cardinality. The set-covering problem is known to be NP-complete [38].

In the following discussion, we formulate a special set-covering problem from the $\hat{C}^3$ problem and demonstrate that its solution is a solution to the $\hat{C}^3$ problem. This special set-covering problem is referred to as the SSC problem.

Recall that for the $\hat{C}^3$ problem $S_c = \{g_i(X) \leq 0, i = 1, 2, \cdots, m'$ and $\bar{g}_i(X) = 0, i = m'+1, \cdots, m\}$. For the discussion below, it is not necessary to know whether a member of S_c is an inequality constraint or an equality constraint. Hence, we can consider S_c as a set of constraints of the form: $S_c = \{p_1(X), p_2(X), \cdots, p_m(X)\}$ where:

$$p_i(X) = \begin{cases} g_i(X) \leq 0 & \text{if } i \in [1, m'] \\ \bar{g}_i(X) = 0 & \text{if } i \in [m'+1, m] \end{cases}$$

Also recall that $\wp(S_c)$ denotes the power set of S_c; i.e., the set of all subsets of S_c. Let members of $\wp(S_c) - \{\emptyset\}$ be denoted by $P_1, P_2, \cdots, P_{\mathcal{M}}$ where $\mathcal{M} = 2^m - 1$. Each $P_j \in \wp(S_c) - \{\emptyset\}$ can be uniquely characterized by the $m-$ vector μ_j where

$$\mu_{ji} = \begin{cases} 1 & \text{if } p_i \in P_j \\ 0 & \text{otherwise} \end{cases}$$

Let $\Gamma_{\mathcal{M}}$ be the set of all possible $\mathcal{M}-$ vectors whose entries are either 0 or 1. For $y \in \Gamma_{\mathcal{M}}$, let $E(y) = \{P_i \in \wp(S_c) - \{\emptyset\} : y_i = 1\}$. Consider now the problem:

$$\min_{y \in \Gamma_{\mathcal{M}}} \quad \sum_{j=1}^{\mathcal{M}} y_j$$

subject to:

($) $\quad \sum_{j=1}^{\mathcal{M}} \mu_{ji} y_j \geq 1 \qquad \forall i \in I_m$

($$) for each non-zero y_j, the associated P_j is consistent

This problem is an extension of the set-covering problem which incorporates the consistency constraint of ($$) and is referred to as the SSC problem associated with the set of constraints S_c.

A $y \in \Gamma_{\mathcal{M}}$ which satisfies ($) and ($$) is called a feasible solution of the SSC problem.

An upper bound for $\min_{y \in \Gamma_{\mathcal{M}}} \sum_{j=1}^{\mathcal{M}} y_j$ is m. This occurs when no constraints are mutually consistent and hence the individual constraints themselves must be chosen. Note that the fundamental property (Υ) (section 3.1) makes this possible.

We demonstrate, in Theorem 4.6 that a solution to the SSC problem associated with the set of constraints S_c is a solution to the $\hat{C}^3$ problem. The proof depends on the following Lemma.

Lemma 4.3: Let $y \in \Gamma_{\mathcal{M}}$, then y is a feasible solution to the SSC problem if and only if $E(y)$ is a consistent cover of S_c.

Proof: (a) *if part*

To prove that $E(y)$ is a consistent cover of S_c, we must show that

i) $\forall P \in E(y)$ [P is consistent]

ii) $\forall p \in S_c$ $[\exists P \in E(y) : p \in P]$

Since y is a feasible solution to the SSC problem then y must satisfy (\$\$). By definition of $E(y)$, property (i) follows directly.

Since y is a feasible solution to the SSC problem then y must satisfy (\$). That is, $\sum_{j=1}^{\mathcal{M}} \mu_{ji} y_j \geq 1 \quad \forall i \in I_m$. This implies that $\forall i \in I_m \quad \exists a_i \in I_{\mathcal{M}}$ such that $\mu_{a_i i} = 1$ and $y_{a_i} = 1$. Hence, every $p_i \in S_c$ belongs to some $P_j \in E(y)$. This proves property (ii).

(b) *only if part*

Since $E(y)$ is a consistent cover of S_c then $\forall P_j \in E(y) \quad P_j$ is consistent and by definition of $E(y)$, $y_j = 1$. Therefore, y satisfies (\$\$).

Since $E(y)$ is a consistent cover of S_c then $\forall p_i \in S_c \quad \exists P_j \in E(y) : p_i \in P_j$ which implies that $\mu_{ji} = 1$ and by definition of $E(y)$, $y_j = 1$. Therefore, $\sum_{j=1}^{\mathcal{M}} \mu_{ji} y_j \not< 1 \quad \forall i \in I_m$. This proves that y satisfies (\$).

Hence, y is feasible solution to the SSC problem. □

Theorem 4.6: Let $y^* \in \Gamma_{\mathcal{M}}$ be a solution to the SSC problem, then $E(y^*)$ is a minimum consistent cover of S_c.

Proof: From Lemma 4.3, $E(y^*)$ is a consistent cover of S_c. If $E(y^*)$ is not a minimum consistent cover of S_c, then there must exist $\hat{y} \in \Gamma_{\mathcal{M}}$ such that $E(\hat{y})$ is a consistent cover of S_c with $\mid E(\hat{y}) \mid < \mid E(y^*) \mid$. From Lemma 4.3, $E(\hat{y})$ must be a feasible solution to the SSC problem. Furthermore, $\mid E(\hat{y}) \mid < \mid E(y^*) \mid$ implies that $\sum_{i=1}^{\mathcal{M}} \hat{y}_i < \sum_{i=1}^{\mathcal{M}} y_i^*$ which contradicts the assumption that y^* is a solution to the SSC problem. □

4.3 Summary

We analyzed the C^3 problem using graph-theoretical concepts. We introduced two new concepts called a *constraint set graph* and a *consistent constraint set graph*. We established a relation between a maximum independent set of a constraint set graph and $Min(S_c)$. Depending upon the consistency of the constraint set graph, we showed that a simplified version of the C^3 problem is related either to the edge cover problem or to the clique cover problem. We also formulated a simplified version of the C^3 problem as a special set-covering problem.

Because of its relationship to the clique cover problem which is known to be a difficult problem, we restrict our considerations in the following chapters to the development of a heuristic algorithm for the C^3 problem.

Chapter 5

An Approximation Algorithm for the C^3 Problem

In chapter 4, we showed that the C^3 Problem is related to the clique cover problem which is known to be a difficult problem [38]. In this chapter, we develop an approximation algorithm for solving the C^3 problem. The approximation algorithm maximizes the number of mutually consistent constraints (i.e. finds a maximum consistent subset) in every iteration with the 'hope' that this choice will result in a minimum number of consistent subsets.

In section 5.1, we identify a basic stage of the greedy heuristic algorithm for the C^3 problem and develop a solution procedure for this stage for the case where $S_v = \emptyset$. This greedy algorithm (which we call the C^3 algorithm) is presented in section 5.2.

5.1 Formulation of the PMIP Problem

The approximation algorithm for the C^3 problem (section 3.1) developed in section 5.2 is based on solving a problem P on every iteration. Let $U \subseteq I_p - I_l = \{l+1, l+2, \cdots, p\}$ be an index set for a prescribed non-empty subset of the constraint set, $S_c \bigcup S_v$. Then, P is the problem of finding an index set $\tilde{S}$ which identifies a largest possible consistent subset of $S_f \bigcup S_c \bigcup S_v$ subject to the constraint that $\tilde{S}$ contains all the elements of S_f and the largest possible number of elements of U. (Property Υ ensures that $\tilde{S} \neq \emptyset$.) Since our considerations in this chapter deal with the case where $S_v = \emptyset$, this implies $p = m$. The Venn diagram shown in Figure 5.1 illustrates in set-theoretic notation the relationships among sets $I_m - I_l$, U, I_l and $\tilde{S}$.

A solution procedure for problem P is formulated as a mixed integer programming problem. A first step in this process is to rewrite the constraints in $S_f \bigcup S_c$ in the following form:

$$f_i(X) \leq 0;\ i = 1, 2, \cdots, l'$$

$$\bar{f}_i(X) \geq 0;\ i = l'+1, l'+2, \cdots, l$$

$$\bar{f}_i(X) \leq 0;\ i = l'+1, l'+2, \cdots, l$$

$$g_i(X) \leq 0;\ i = l+1, l+2, \cdots, m'$$

$$\bar{g}_i(X) \geq 0;\ i = m'+1, m'+2, \cdots, m$$

$$\bar{g}_i(X) \leq 0;\ i = m'+1, m'+2, \cdots, m$$

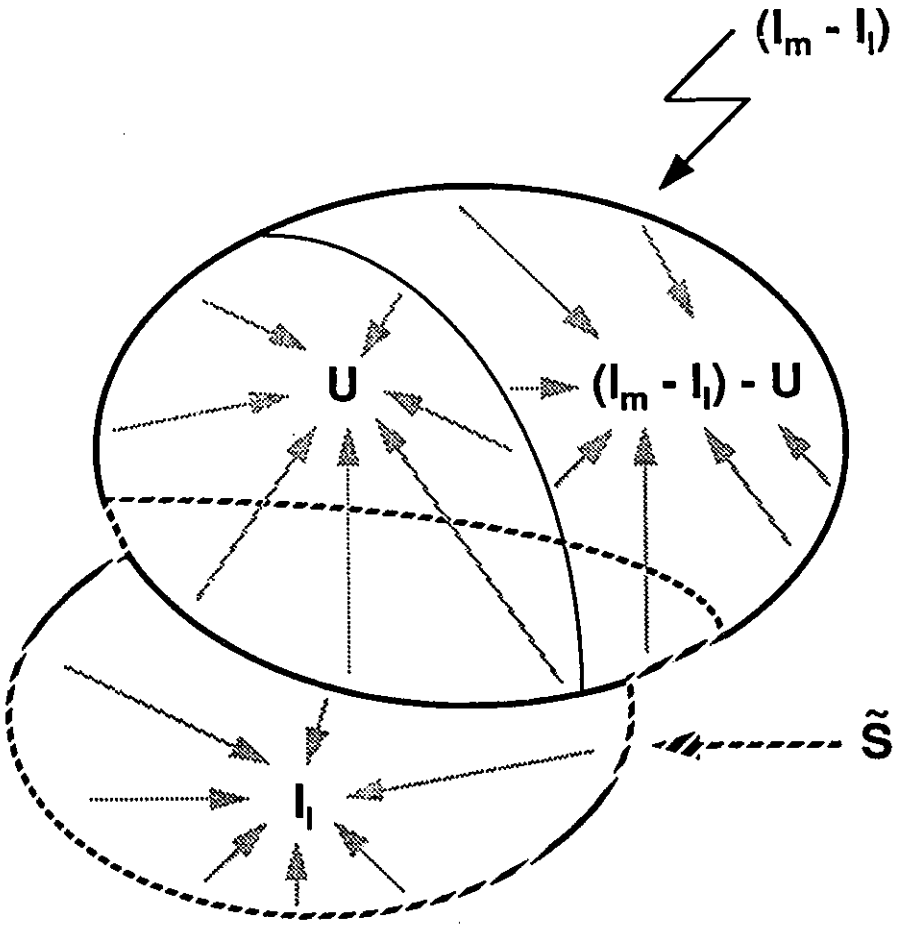

Figure 5.1. Venn diagram illustrating relationships among U, $(I_m - I_l)$, I_l, $\tilde{S}$

Consider now the following mixed integer programming problem which we call the PMIP problem:

$$\min_{\substack{X \in \hat{\Re}_n \\ C \in \Gamma}} \quad M \sum_{U} C_i + \sum_{I_m - I_l - U} C_i$$

subject to:

$$f_i(X) \leq 0;\ i = 1, 2, \cdots, l'$$

$$\bar{f}_i(X) \geq 0;\ i = l'+1, l'+2, \cdots, l$$

$$\bar{f}_i(X) \leq 0;\ i = l'+1, l'+2, \cdots, l$$

$$g_i(X) - BC_i \leq 0;\ i = l+1, l+2, \cdots, m' \qquad \text{.... (\$)}$$

$$\bar{g}_i(X) + BC_i \geq 0;\ i = m'+1, m'+2, \cdots, m$$

$$\bar{g}_i(X) - BC_i \leq 0;\ i = m'+1, m'+2, \cdots, m$$

with

$$B > \max_{X \in \hat{\Re}_n} \{ \mid g_i(X) \mid, \mid \bar{g}_i(X) \mid : i = l+1, l+2, \cdots, m \}$$

$$M > \mid I_m - I_l - U \mid$$

Γ is the set of all possible $(m-l)$ - vectors whose entries are either 0 or 1, and the binary variables C_i, $i = l+1, \cdots, m$ are called the *constraint satisfaction indicators.* These variables are the components of the $(m-l)-$ vector C. Notice that the specification for the coefficients, B, ensures that the search space for the PMIP problem is Φ_f; i.e., for every $\hat{X} \in \Phi_f$, there is $\hat{C} \in \Gamma$ such that $Z(\hat{Y}) = \left[M \sum_U C_i + \sum_{I_m - I_l - U} C_i \right]_{C = \hat{C}}$

is defined subject to (\$) where $\hat{Y} = \begin{bmatrix} \hat{X} \\ \hat{C} \end{bmatrix}$. The positive constant M is used for prioritizing the constraints of S_c.

In the case where constraint set $S_c^d = \{\bigwedge_{w=1}^{d_i}[g_{iw}(X)\mathbf{R}_{iw}0], i = l+1, l+2, \cdots, m\}$ has an element $c_k = \bigwedge_{w=1}^{d_k}[g_{kw}(X)\mathbf{R}_{kw}0]$ for which $d_k > 1$, we use the same *constraint satisfaction indicator* C_k for every individual constraint in the element c_k. B is correspondingly defined as follows:

$$B > \max_{X \in \Re_n} \{ \mid g_{iw}(X) \mid : w = 1, 2, \cdots, d_i \text{ and } i = l+1, l+2, \cdots, m\}$$

Theorem 5.1: Let the $(n+m-l)$ - vector $Y^* = \begin{bmatrix} X^* \\ C^* \end{bmatrix}$ be a solution to the PMIP problem and let $Q^* \subseteq I_m - I_l$ be the set of indices of the subvector C^* which have a zero value. If $\hat{Y} = \begin{bmatrix} \hat{Y} \\ \hat{C} \end{bmatrix}$ is any other solution generated for the PMIP problem, then $\mid \hat{Q} \bigcap U \mid = \mid Q^* \bigcap U \mid$ where $\hat{Q}$ is analogously defined to Q^*.

Proof: Because both Y^* and $\hat{Y}$ are solutions to the PMIP problem, it follows that:

$$Z(Y^*) = Z(\hat{Y}) \tag{5.1}$$

where $Z(Y) = M \sum_U C_i + \sum_{I_m - I_l - U} C_i$.

Note that

$$Z(Y^*) = M(\mid U \mid - \mid Q^* \bigcap U \mid) + (\mid I_m - I_l - U \mid - (\mid Q^* \mid - \mid Q^* \bigcap U \mid))$$

$$Z(\hat{Y}) = M(\mid U \mid - \mid \hat{Q} \bigcap U \mid) + (\mid I_m - I_l - U \mid - (\mid \hat{Q} \mid - \mid \hat{Q} \bigcap U \mid))$$

The equation (5.1) now becomes:

$$M \mid \hat{Q} \bigcap U \mid + \mid \hat{Q} \mid - \mid \hat{Q} \bigcap U \mid = M \mid Q^* \bigcap U \mid + \mid Q^* \mid - \mid Q^* \bigcap U \mid \tag{5.2}$$

Suppose $\mid \hat{Q} \bigcap U \mid \neq \mid Q^* \bigcap U \mid$. Then there are three cases to consider:

Case 1: $|\hat{Q}| - |\hat{Q} \cap U| = |Q^*| - |Q^* \cap U|$

Now the equation (2) becomes:

$$M|\hat{Q} \cap U| = M|Q^* \cap U|$$

$$\Longrightarrow |\hat{Q} \cap U| = |Q^* \cap U| \qquad \because M \neq 0$$

This is a contradiction.

Case 2: $|\hat{Q}| - |\hat{Q} \cap U| < |Q^*| - |Q^* \cap U|$

Since $|\hat{Q} \cap U| \neq |Q^* \cap U|$. This implies that: $|\hat{Q} \cap U| > |Q^* \cap U|$ or $|\hat{Q} \cap U| < |Q^* \cap U|$.

(a): Consider $|\hat{Q} \cap U| > |Q^* \cap U|$

$$\Longrightarrow |Q^* \cap U| + \Delta = |\hat{Q} \cap U| \qquad \text{where } \Delta \geq 1.$$

Now the equation (5.2) becomes:

$$M(|Q^* \cap U| + \Delta) + |\hat{Q}| - |\hat{Q} \cap U| = M|Q^* \cap U| + |Q^*| - |Q^* \cap U|$$

$$\Longrightarrow |Q^*| - |Q^* \cap U| = |\hat{Q}| - |\hat{Q} \cap U| + M\Delta \tag{5.3}$$

Observe that $0 \leq |Q^*| - |Q^* \cap U| \leq |I_m - I_l - U|$ and $0 \leq |\hat{Q}| - |\hat{Q} \cap U| \leq |I_m - I_l - U|$. Since $\Delta \geq 1$ and $M > |I_m - I_l - U|$, we have from equation (5.3):

$$|Q^*| - |Q^* \cap U| > |I_m - I_l - U|.$$

This is impossible.

(b): Consider $|\hat{Q} \bigcap U| < |Q^* \bigcap U|$

$$\Longrightarrow |\hat{Q} \bigcap U| + \Delta = |Q^* \bigcap U| \qquad \text{where } \Delta \geq 1.$$

Now the equation (5.2) becomes:

$$M|\hat{Q} \bigcap U| + |\hat{Q}| - |\hat{Q} \bigcap U| = M(|\hat{Q} \bigcap U| + \Delta) + |Q^*| - |Q^* \bigcap U|$$

$$\Longrightarrow |\hat{Q}| - |\hat{Q} \bigcap U| = |Q^*| - |Q^* \bigcap U| + M\Delta$$

$$\Longrightarrow |\hat{Q}| - |\hat{Q} \bigcap U| > |Q^*| - |Q^* \bigcap U| \qquad \because \Delta \geq 1 \text{ and } M > 0$$

This is a contradiction.

Case 3: $|\hat{Q}| - |\hat{Q} \bigcap U| > |Q^*| - |Q^* \bigcap U|$

Since $|\hat{Q} \bigcap U| \neq |Q^* \bigcap U|$. This implies that: $|\hat{Q} \bigcap U| > |Q^* \bigcap U|$ or $|\hat{Q} \bigcap U| < |Q^* \bigcap U|$.

(a): Consider $|\hat{Q} \bigcap U| > |Q^* \bigcap U|$

$$\Longrightarrow |Q^* \bigcap U| + \Delta = |\hat{Q} \bigcap U| \qquad \text{where } \Delta \geq 1.$$

Now the equation (5.2) becomes:

$$M(|Q^* \bigcap U| + \Delta) + |\hat{Q}| - |\hat{Q} \bigcap U| = M|Q^* \bigcap U| + |Q^*| - |Q^* \bigcap U|$$

$$\Longrightarrow |Q^*| - |Q^* \bigcap U| = |\hat{Q}| - |\hat{Q} \bigcap U| + M\Delta$$

$$\Longrightarrow |Q^*| - |Q^* \bigcap U| > |\hat{Q}| - |\hat{Q} \bigcap U| \qquad \because \Delta \geq 1 \text{ and } M > 0$$

This is a contradiction.

(b): Consider $|\hat{Q} \bigcap U| < |Q^* \bigcap U|$

$$\Longrightarrow |\hat{Q} \bigcap U| + \Delta = |Q^* \bigcap U| \qquad \text{where } \Delta \geq 1.$$

Now the equation (5.2) becomes:

$$M|\hat{Q} \bigcap U| + |\hat{Q}| - |\hat{Q} \bigcap U| = M(|\hat{Q} \bigcap U| + \Delta) + |Q^*| - |Q^* \bigcap U|$$

$$\Longrightarrow |\hat{Q}| - |\hat{Q} \bigcap U| = |Q^*| - |Q^* \bigcap U| + M\Delta \tag{5.4}$$

Observe that $0 \leq |Q^*| - |Q^* \bigcap U| \leq |I_m - I_l - U|$ and $0 \leq |\hat{Q}| - |\hat{Q} \bigcap U| \leq |I_m - I_l - U|$. Since $\Delta \geq 1$ and $M > |I_m - I_l - U|$, we have from equation (5.4):

$$|\hat{Q}| - |\hat{Q} \bigcap U| > |I_m - I_l - U|.$$

This is impossible. □

Theorem 5.2: Let $Y^* = \begin{bmatrix} X^* \\ C^* \end{bmatrix}$ be an $(n + m - l)$ - vector generated by solving the PMIP problem and let $Q^* \subseteq I_m - I_l$ be the set of indices of the subvector C^* which have a zero value. The set of constraints $\{g_i : i \in Q^*\}$ is a solution to problem P.

Proof: Observe first that for each $i \in Q^*$, $g_i(X^*)$ is satisfied, then the constraint set referenced by Q^* is consistent. Suppose Q^* is not a solution to problem P. Then there must exist a consistent subset of S_c characterized by some index $\hat{Q} \subseteq I_m - I_l$ such that:

either *a)* $|\hat{Q}| > |Q^*|$ and $|\hat{Q} \bigcap U| = |Q^* \bigcap U|$
or *b)* $|\hat{Q} \bigcap U| > |Q^* \bigcap U|$ and $|\hat{Q}| = |Q^*|$
or *c)* $|\hat{Q}| > |Q^*|$ and $|\hat{Q} \bigcap U| > |Q^* \bigcap U|$

Furthermore, associated with this assumption there must exist $\hat{X} \in \Phi_f$ such that $g_i(\hat{X})$ is satisfied for all $i \in \hat{Q}$. Let $\hat{C}$ be the $(m - l)$ - vector constructed according

to: $C_i = 0$ if $i \in \hat{Q}$ and $C_i = 1$ if $i \in I_m - I_l - \hat{Q}$. Let $\hat{Y} = \begin{bmatrix} \hat{X} \\ \hat{C} \end{bmatrix}$ and recall that the existence of $Z(\hat{Y})$ is assured.

Note that the essential structure of the minimization activity associated with the PMIP problem, ensures that $Z(\hat{Y}) \geq Z(Y^*)$; i.e.

$$Z(\hat{Y}) - Z(Y^*) \geq 0 \tag{5.5}$$

Furthermore,

$$Z(\hat{Y}) = M(\mid U \mid - \mid \hat{Q} \bigcap U \mid) + (\mid I_m - I_l - U \mid - (\mid \hat{Q} \mid - \mid \hat{Q} \bigcap U \mid)) \tag{5.6}$$

$$Z(Y^*) = M(\mid U \mid - \mid Q^* \bigcap U \mid) + (\mid I_m - I_l - U \mid - (\mid Q^* \mid - \mid Q^* \bigcap U \mid)) \tag{5.7}$$

From inequality (5.5) together with equations (5.6) and (5.7) we obtain:

$$(M-1)(\mid Q^* \bigcap U \mid - \mid \hat{Q} \bigcap U \mid) + (\mid Q^* \mid - \mid \hat{Q} \mid) \geq 0 \tag{5.8}$$

Note now that each of conditions (a), (b) and (c) would yield a negative value for the left-hand-side of inequality (5.8), thereby contradicting inequality (5.8). The solution characterized by Q^* must therefore be a solution to problem P. □

5.2 A Greedy Approximation Algorithm

Many problems of practical significance are intractable [38]. This has stimulated research into approximation algorithms [19, 38, 39, 52, 53, 57, 98, 99] which are guaranteed to find solutions that are "close" to optimal. Such guarantees are characterized by ratio bounds. These are defined as follows. An approximation algorithm for a problem has a ratio bound of $\rho(n)$ if for any input of size n, the cost of the solution produced by the approximation algorithm (denoted by C) is within a factor of $\rho(n) \geq 1$ of the cost C^* of an optimal solution. That is, for minimization problems, $C \leq \rho(n) \cdot C^*$. For problems such as vertex cover and the traveling salesman problem with triangle inequality, approximation algorithms have been developed which have a fixed ratio bound, independent of n, whereas approximation algorithms for the set-covering problem typically have a ratio bound which grows as a function of the input size n i.e., $\rho(n)$ is monotonically increasing with n. Excellent discussions of approximation algorithms are given in [18, 38, 48, 57, 88, 100, 101].

A greedy approximation algorithm for the C^3 problem is presented in this section. The approach has the same structure as that typically used for the set-covering problem [14, 18, 53, 63]. Indeed, all greedy algorithms have the same structure [57]. The greedy approach [14, 18, 29, 30, 53, 57, 58, 63, 91] which is sometimes called *myopic*, always makes the choice that promises to be the best at the moment. That is, it makes a locally optimal (greedy) choice in the hope that this choice will lead to a globally optimal solution [18]. There are many algorithms in the literature that can be viewed as applications of the greedy approach. They include minimum spanning tree algorithms [58, 91], Dijkstra's algorithm for shortest paths from a single source [28] and greedy set-covering heuristics [14, 18, 53, 63].

An approximation algorithm for the C^3 problem developed in this section is another application of the greedy approach. It is based on maximizing the number of mutually consistent constraints in every iteration.

To facilitate the specification of the algorithm for the C^3 problem given below, we first define the symbols used in the algorithm.

E is a consistent cover of $S_f \bigcup S_c \bigcup S_v$ and $\tilde{E}= G(E)$ is the associated index set (section 3.1) generated by the C^3 algorithm given below. $CS(U, \tilde{S})$ represents a computation which solves the problem P defined in section 5.1.

C^3_ALGORITHM $(I_p - I_l)$

1 $U \longleftarrow I_p - I_l$

2 $\tilde{E} \longleftarrow \emptyset$

3 **While** $U \neq \emptyset$

4 **do** $CS(U, \tilde{S})$

5 $U \longleftarrow U - \tilde{S}$

6 $\tilde{E} \longleftarrow \tilde{E} \bigcup \{\tilde{S}\}$

7 **return** $(\tilde{E})$

Suppose $U = \emptyset$ at the end of the k^{th} iteration of the algorithm where $k \geq 1$. Let $\tau_{\max} = \max_{1 \leq i \leq k} \tau_i$ where τ_i the computational cost of solving CS for the i^{th} iteration. Then, the running time of the algorithm is $O(\mid S_c \bigcup S_v \mid \cdot \tau_{\max})$.

Theorem 5.3: The C^3 algorithm terminates.

Proof: To prove that the C^3 algorithm terminates, we have to show that the While loop terminates eventually. That is, eventually $U = \emptyset$.

From property Υ (section 3.1), it follows that $\tilde{S} \neq \emptyset$ and from this we are assured that $U = \emptyset$ in at most $| S_c \bigcup S_v |$ iterations. □

Theorem 5.4: Let $E = G^{-1}(\tilde{E})$ where $\tilde{E}$ is the set returned by the C^3 algorithm. Then, E is a consistent cover of $S_f \bigcup S_c \bigcup S_v$.

Proof: We must show that:

1. $\forall E_i \in E$ $\quad [E_i$ is consistent $]$

2. $\forall E_i \in E \quad S_f \subset E_i \quad$ and $\quad \forall c \in S_c \bigcup S_v \quad [\exists E_i \in E : c \in E_i]$

Initially, $E = \emptyset$ and therefore, the property 1 is trivially true upon initialization.

At every iteration, $\tilde{E}$ is augmented by $\tilde{S}$. Observe that from the definition of CS, $\tilde{S}$ is an index set corresponding to a consistent subset of $S_f \bigcup S_c \bigcup S_v$. Therefore, the property 1 remains true.

Observe that at every iteration $\tilde{S}$ contains the index for every $c \in S_f$ by Theorem 5.2. Hence, for every $E_i \in E$, $S_f \subset E_i$. Initially, $U = I_p - I_r$ which implies that the indices of all $c \in S_c \bigcup S_v$ are in U. At every iteration, the elements in subset $\tilde{S}$ formed by the CS procedure are removed from U and the set $\tilde{S}$ is added to $\tilde{E}$. Therefore, at any iteration, the index for every $c \in S_c \bigcup S_v$ is either in U or in some $e \in \tilde{E}$. Property 2 follows from the fact that $U = \emptyset$ at the termination of the algorithm. □

From Theorem 5.3 and Theorem 5.4, we conclude that the C^3 algorithm is correct.

The C^3 algorithm has the same structure as the GREEDY_SET_COVER algorithm for the set-covering problem as given in [18]. Given a finite set X and a family $\mathcal{F} = \{S_1, S_2, \cdots, S_m\}$ of subsets of X, GREEDY_SET_COVER algorithm picks, at each iteration, the set S_i from $\mathcal{F}$ that covers the most remaining uncovered elements

of X. In a similar way, the C^3 algorithm, at each iteration, forms the largest consistent subset of $S_f \bigcup S_c \bigcup S_v$ that includes the most remaining unsatisfied constraints of $S_c \bigcup S_v$. This step is carried out by the procedure CS.

An analysis of the C^3 algorithm can be easily modeled after that given for the greedy heuristics algorithm for the set covering problem [14, 18] and an equivalent result of a logarithmic ratio bound can be obtained. That is, as the size of the problem instance (i.e., the cardinality of $S_c \bigcup S_v$) gets larger, the size of the approximate solution (i.e., the size of the consistent cover generated by the C^3 algorithm) may grow relative to the size of an optimal solution, $Min(S_c \bigcup S_v)$. The C^3 algorithm has a ratio bound of $\ln(\mid S_c \bigcup S_v \mid) + 1$.

The following two examples illustrate the working of the algorithm.

Example 5.1 Consider the set of linear constraints:

$$\begin{array}{lrcrcl} c_1: & X_1 & + & X_2 & \leq & 2 \\ c_2: & 6X_1 & + & 5X_2 & \leq & 30 \\ c_3: & -X_1 & + & X_2 & \geq & 3 \\ c_4: & 11X_1 & - & 3X_2 & \geq & 33 \\ c_5: & 4X_1 & - & 7X_2 & \geq & 28 \end{array}$$

$$(X_1, X_2) \in \hat{\Re}_2 \subset \hat{\Re}_2^+, S_c = \{c_1, c_2, c_3, c_4, c_5\}, S_f = S_v = \emptyset$$

Figure 5.2 shows the graphical representation of the feasible (shaded) regions formed by pairwise consistent constraints. From Figure 5.2, we find that only the constraint pairs: $\{1,2\}$, $\{2,3\}$, $\{2,4\}$, $\{3,4\}$ and $\{4,5\}$ are consistent.

Figure 5.3 shows the constraint set graph representation. Note (from Figure 5.2) that the clique $\{2,3,4\}$ is not consistent.

As shown in Example 8 of Appendix C, the C^3 algorithm generates, $\tilde{E}$= $\{\{2,4\}, \{4,5\}, \{1,2\}, \{2,3\}\}$, as a consistent cover with $|\tilde{E}| = 4$. (The order in which these

elements are listed is the order in which they are generated by the algorithm.) From Figure 5.2, the minimal consistent cover has size 3. In fact it is $\{\{1,2\},\{2,3\},\{4,5\}\}$ or $\{\{1,2\},\{3,4\},\{4,5\}\}$. Observe that the consistent subset $\{2,4\}$ in $\tilde{E}$ is redundant.

Example 5.2 Consider the set of linear constraints:

$$\begin{array}{lrcrcr} c_1: & 4X_1 & + & 3X_2 & \leq & 12 \\ c_2: & -2X_1 & + & X_2 & \geq & 2 \\ c_3: & -X_1 & + & X_2 & \geq & 6 \\ c_4: & 16X_1 & + & 17X_2 & \geq & 272 \\ c_5: & 2X_1 & - & 7X_2 & \geq & 14 \end{array}$$

$$(X_1, X_2) \in \hat{\Re}_2 \subset \hat{\Re}_2^+, S_c = \{c_1, c_2, c_3, c_4, c_5\}, S_f = S_v = \emptyset$$

Figure 5.4 shows the graphical representation of the feasible (shaded) regions formed by pairwise consistent constraints. Figure 5.5 shows the constraint set graph representation. Note (from Figure 5.4) that the clique $\{2,3,4\}$ is consistent.

As shown in Example 9 of Appendix C, the C^3 algorithm generates, $\tilde{E}$= $\{\{2,3,4\}$, $\{1,2\},\{4,5\}\}$, as a consistent cover with $|\tilde{E}| = 3$. (The order in which these elements are listed is the order in which they are generated by the algorithm.) From Figure 5.4, the minimal consistent cover is unique and has size 3. In fact it is $\{\{1,2\},\{2,3,4\},\{4,5\}\}$.

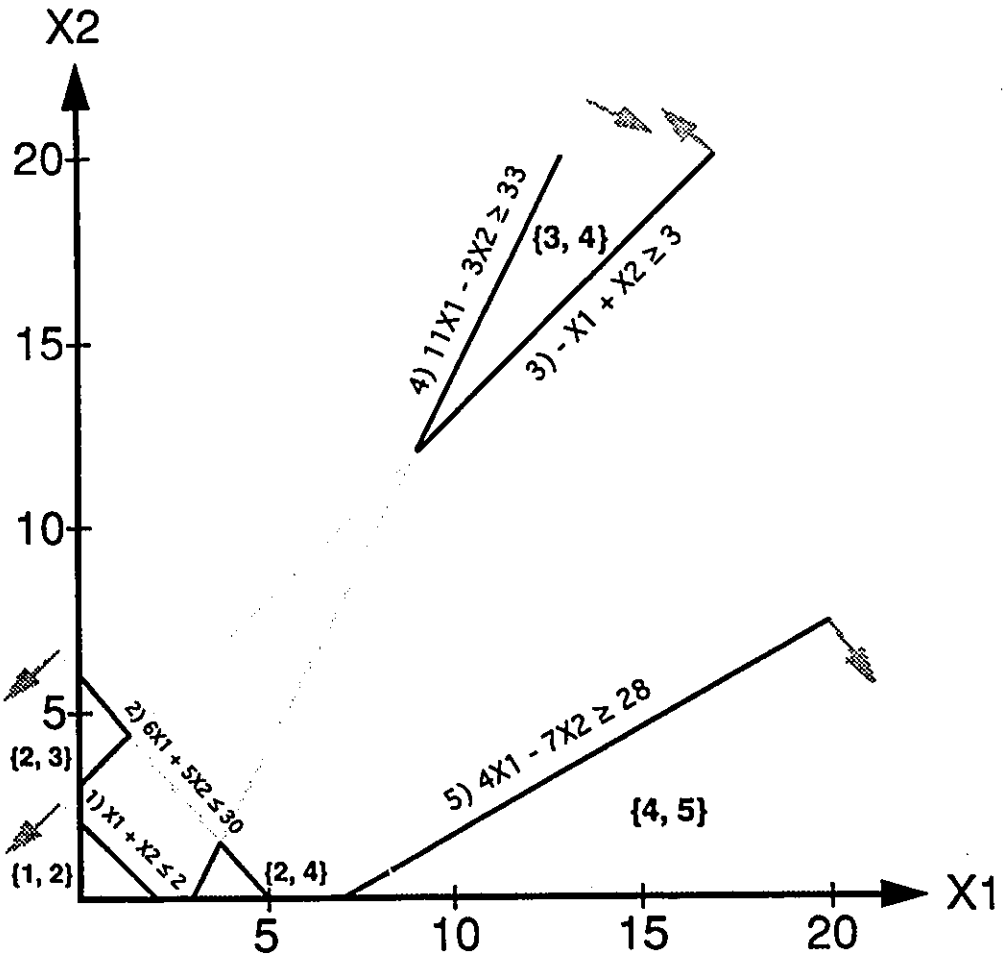

Figure 5.2. Constraint Graph for Example 5.1

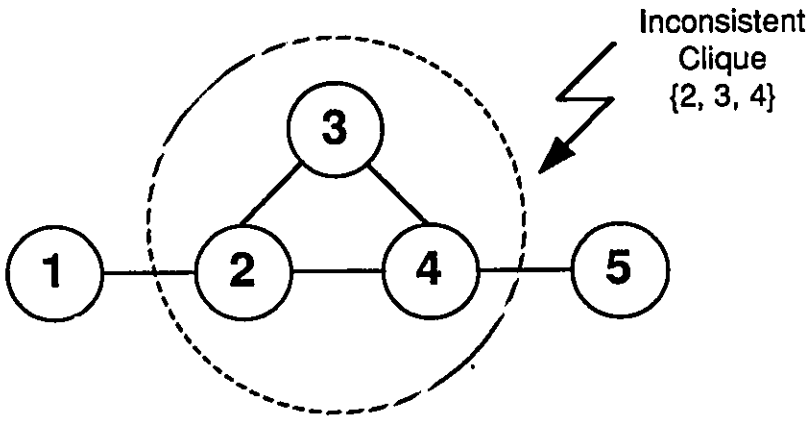

Figure 5.3. Constraint Set Graph for Example 5.1

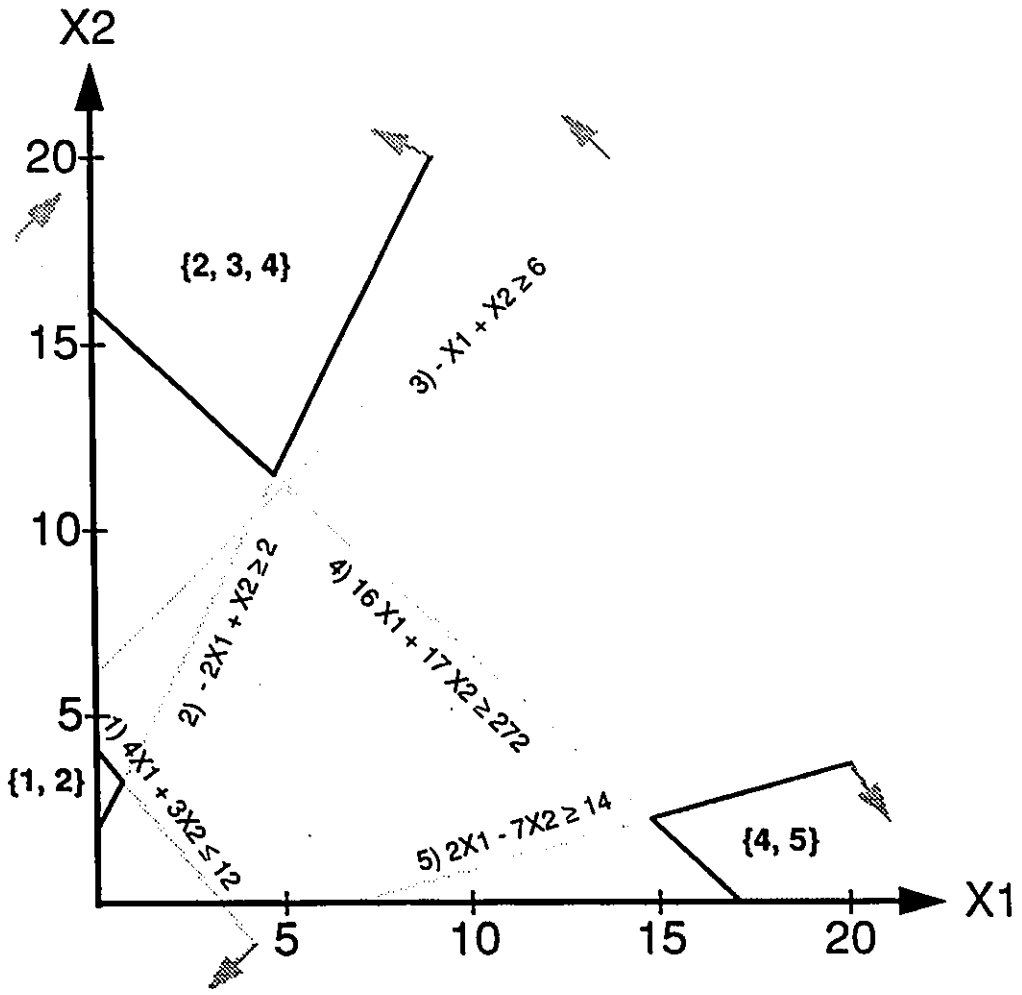

Figure 5.4. Constraint Graph for Example 5.2

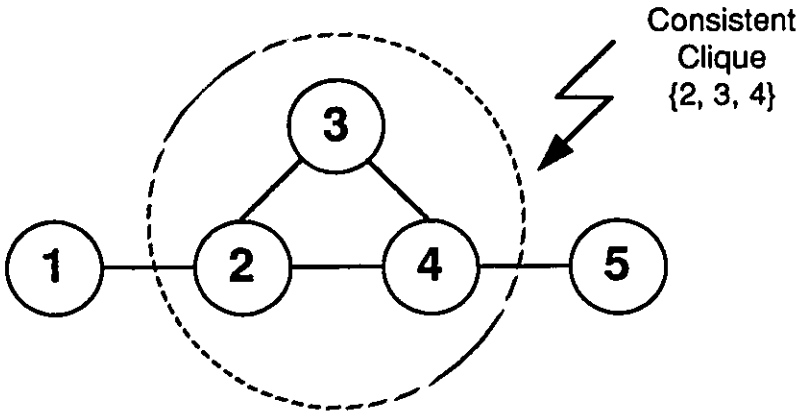

Figure 5.5. Constraint Set Graph for Example 5.2

Chapter 6

Analysis of Perturbation Constraints

In chapter 5, we presented the C^3 algorithm for the case where $S_v = \emptyset$; i.e., in the absence of perturbation constraints (PC's). In this chapter, we introduce a mechanism for handling perturbation constraints; i.e., the case where $S_v \neq \emptyset$. It is based on the concept of slack which is informally described as follows.

Consider for example, a simple linear constraint $2X_1+3X_2 \leq 6$ with $X_1, X_2 \in \Re_2^+$. Observe that $2X_1+3X_2 \leq 6$ is satisfied at $X^* = [1,1]$ and $X^e = [0,2]$. It is interesting to note that X_1 can be increased from $X_1^* = 1$ to $X_1^* + \epsilon_1 = 1 + \epsilon_1$ where $\epsilon_1 = \frac{1}{2}$ and the constraint $2X_1 + 3X_2 \leq 6$ remains satisfied at $X' = [1.5, 1]$. However, the same cannot be done for $X^e = [0,2]$. In fact, there exists no $\epsilon_1 > 0$ such that the constraint $2X_1+3X_2 \leq 6$ can be satisfied at $[X_1^e+\epsilon_1, X_2^e]$. The constraint $2X_1+3X_2 \leq 6$ though is satisfied at both X^* and X^e, however, as far as increasing X_1 is concerned, X^* is useful whereas X^e is not useful. The difference between X^* and X^e lies in the fact that the constraint $2X_1 + 3X_2 \leq 6$ is satisfied at X^* as a strict inequality (i.e., $2X_1^* + 3X_2^* = 5 < 6$) whereas $2X_1 + 3X_2 \leq 6$ is satisfied at X^e as an equality (i.e.,

$2X_1^c + 3X_2^c = 6$). More specifically, in the case of X^*, there is "room" for increasing X_1 (i.e., slack is positive) and in the case of X^c there is no "room" for increasing X_1 (i.e., slack is zero). Therefore, one way to handle increasing X_j in $h(X) \leq 0$ is to find such an X^* such that $h(X^*) < 0$ and use the positive slack in $h(X) \leq 0$ with respect to X^* for the purpose of increasing X_j. The development of a formal basis for increasing X_j is the objective of this chapter.

We formally define in section 6.1 the concept of slack and the associated concepts of insensitive and potentially sensitive to X_j for any $j \in I_n$. We carry out the PC analysis (in sections 6.2 and 6.3) from two different perspectives; namely, (1) slack at a constraint level and (2) slack at a variable level. The analysis in each case starts with the presentation of a formal basis for handling a single PC which is then extended to a set of PC's and then to the case where a set of PC's is combined with a non-empty $S_f \bigcup S_c$. The analysis is then incorporated into a rule form for handling PC's. Finally in section 6.4, we compare the resulting two rules.

To facilitate our analysis of PC's, we introduce the following notation and terminology:

a) For X an $n-$vector and ϵ a given scalar value, we use $\Lambda_j(X, \epsilon)$ to be the perturbed $n-$vector given by $[X_1, X_2, \cdots, X_j + \epsilon, X_{j+1}, \cdots, X_n]$.

b) Let S_e be a given set of perturbation constraints and $c = \lambda_j(X, X', \epsilon) \wedge h(X) \leq 0 \wedge h(X') \leq 0 \in S_e$. We refer to $\lambda_j(X, X', \epsilon)$ as the *primary* component of c, $h(X) \leq 0$ as the *secondary* component of c and X_j as the primary variable of c. There is a primary variable associated with each perturbation constraint in S_e and we use P_{S_e} to be the set of indices of j, such that X_j is a primary variable of $c \in S_e$. We call P_{S_e} the primary index set of S_e.

Lemma 6.1: Let $\varepsilon > 0$ then $\lambda_j(X, \Lambda_j(X,\varepsilon),\varepsilon)$ is TRUE for all $j \in I_n$.

Proof : Follows from the definition of the predicate λ_j (see section 3.1) and the definition of Λ_j given above. □

6.1 Concept of Slack

Throughout our discussion in this section, we assume that $h : \hat{\Re}_n \longrightarrow \Re$ is a bounded differentiable function.

Definition 6.1: Let $h(X) \leq 0$ be satisfied at X^* (i.e., $h(X^*) \leq 0$) then the value $w = -h(X^*)$ is called the slack in $h(X) \leq 0$ with respect to X^*. Furthermore, if $w > 0$ then there exists a non-zero ε such that $h(\Lambda_j(X^*,\varepsilon)) \leq 0$ for any $j \in I_n$.

Definition 6.2: $h(X) \leq 0$ is said to be insensitive to X_j if and only if $\frac{\partial h}{\partial X_j} \leq 0$ for all $X \in \hat{\Re}_n$.

Definition 6.3: If the constraint $h(X) \leq 0$ is not insensitive to X_j, then it is said to be potentially sensitive to X_j.

Definition 6.4: $h(X) = 0$ is said to be insensitive to X_j if and only if $\frac{\partial h}{\partial X_j} = 0$ for all $X \in \hat{\Re}_n$.

Theorem 6.1: $\frac{\partial h}{\partial X_j} = 0$ for all $X \in \hat{\Re}_n$ if and only if

(1) $h(X) \leq 0$ is insensitive to X_j and

(2) $\tilde{h}\,(X) \leq 0$ is insensitive to X_j where $\tilde{h}\,(X) = -h(X)$.

Proof: (a) *if part*

(1) Follows from Definition 6.2

(2) Observe that $\frac{\partial \tilde{h}}{\partial X_j} = -\frac{\partial h}{\partial X_j} = 0$ for all $X \in \hat{\Re}_n$. This implies from Definition 6.2 $\tilde{h}(X) \leq 0$ is insensitive to X_j.

(b) *only if part*

Since $h(X) \leq 0$ and $\tilde{h}(X) \leq 0$ are insensitive to X_j then $\frac{\partial h}{\partial X_j} \leq 0$ for all $X \in \hat{\Re}_n$ and $\frac{\partial \tilde{h}}{\partial X_j} \leq 0$ for all $X \in \hat{\Re}_n$. There are four cases to consider:

Case (1) : $\frac{\partial h}{\partial X_j} < 0$ and $\frac{\partial \tilde{h}}{\partial X_j} < 0$ for all $X \in \hat{\Re}_n$.

Case (2) : $\frac{\partial h}{\partial X_j} < 0$ and $\frac{\partial \tilde{h}}{\partial X_j} = 0$ for all $X \in \hat{\Re}_n$.

Case (3) : $\frac{\partial h}{\partial X_j} = 0$ and $\frac{\partial \tilde{h}}{\partial X_j} < 0$ for all $X \in \hat{\Re}_n$.

Case (4) : $\frac{\partial h}{\partial X_j} = 0$ and $\frac{\partial \tilde{h}}{\partial X_j} = 0$ for all $X \in \hat{\Re}_n$.

Cases (1), (2) and (3) are impossible because $\frac{\partial \tilde{h}}{\partial X_j} = -\frac{\partial h}{\partial X_j} = 0$. □

Corollary 6.1: $h(X) = 0$ is insensitive to X_j if and only if

(1) $h(X) \leq 0$ is insensitive to X_j and

(2) $\tilde{h}(X) \leq 0$ is insensitive to X_j where $\tilde{h}(X) = -h(X)$.

Proof: Follows from Definition 6.4 and Theorem 6.1. □

Lemma 6.2: Let the constraint $h(X) \leq 0$ be linear; i.e., $h(X) = \alpha_1 X_1 + \alpha_2 X_2 + \cdots + \alpha_j X_j + \cdots + \alpha_n X_n + b$, then $h(X) \leq 0$ is insensitive to X_j if and only if $\alpha_j \leq 0$.

Proof: Follows from Definition 6.2 because $\frac{\partial h}{\partial X_j} = \alpha_j$. □

Theorem 6.2: Let $h(X) = 0$ be a linear constraint. Then show that

1) if $\alpha_j < 0$ then $h(x) \leq 0$ is insensitive to X_j and $\tilde{h}(X) \leq 0$ is potentially sensitive to X_j

2) if $\alpha_j > 0$ then $h(x) \leq 0$ is potentially sensitive to X_j and $\tilde{h}\,(X) \leq 0$ is insensitive to X_j

3) if $\alpha_j = 0$ then $h(x) \leq 0$ and $\tilde{h}\,(X) \leq 0$ are both insensitive to X_j

where $\tilde{h}\,(X) = -h(X)$.

Proof : 1) Since $\alpha_j < 0$ then by Lemma 6.2 $h(X) \leq 0$ is insensitive to X_j. Observe that $\frac{\partial \tilde{h}}{\partial X_j} = -\frac{\partial h}{\partial X_j} = -\alpha_j > 0$. From Definitions 6.2 and 6.3 this implies that $\tilde{h}\,(X) \leq 0$ is potentially sensitive to X_j.

2) Since $\alpha_j > 0$ then from Lemma 6.2 and Definition 6.3 $h(X) \leq 0$ is potentially sensitive to X_j. Observe that $\frac{\partial \tilde{h}}{\partial X_j} = -\frac{\partial h}{\partial X_j} = -\alpha_j < 0$. From Definition 6.2 this implies that $\tilde{h}\,(X) \leq 0$ is insensitive to X_j.

3) Because $0 = \alpha_j = \frac{\partial h}{\partial X_j}$, the result follows directly from Theorem 6.1. □

Definition 6.5: A perturbation constraint whose secondary component is potentially sensitive to X_j, is itself said to be potentially sensitive to X_j.

Lemma 6.3: Suppose $h(X) \leq 0$ is insensitive to X_j. If $h(X) \leq 0$ is satisfied at X^* (i.e., $h(X^*) \leq 0$) then $h(\Lambda_j(X^*, \varepsilon)) \leq 0$ and $h(\Lambda_j(X^*, \varepsilon)) - h(X^*) \leq 0$ for $\varepsilon > 0$.

Proof : Follows from the definitions. □

Theorem 6.3: Suppose $h(X) \leq 0$ is insensitive to X_j. If $h(X) \leq 0$ is satisfied at X^* then the PC $\lambda_j(X, X', \varepsilon) \wedge h(X) \leq 0 \wedge h(X') \leq 0$ is satisfied at the pair $(X^*, \hat{X})$ where $\hat{X} = \Lambda_j(X^*, \varepsilon)$ with $\varepsilon > 0$.

Proof : Follows from the definitions given in section 3.1 and Lemmas 6.1 and 6.3.

□

The objective of our perturbation constraint analysis is to devise a mechanism for increasing X_j for any $j \in I_n$ when a given constraint is potentially sensitive to X_j. The analysis proceeds in two directions because of two different perspectives on PC analysis; namely, slack at the constraint level and slack at the variable level. Each leads to what is called a transition rule for introducing slack. This rule makes it possible to accomodate the $S_v \neq \emptyset$ case and thereby extend our earlier algorithm of chapter 5 to handle the general case. In section 6.2, we develop a formal basis for analyzing PC's based on "slack at a constraint level" which results in Transition Rule A. In section 6.3, we carry out PC analysis based on "slack at a variable level" which results in Transition Rule B. In section 6.4, we compare the transition rules A and B for introducing slack.

6.2 Slack at the Constraint Level

Lemma 6.4 and Corollary 6.2 provide the formal basis for handling a single PC within the framework of slack at the constraint level whereas Theorem 6.4 and Corollary 6.3 extend the analysis to a set of PC's. The analysis is extended to the case where a set of PC's is combined with a non-empty $S_f \bigcup S_c$ in Theorem 6.5. Finally based on this formal analysis, Transition Rule A presents the procedure for handling PC's using the "slack at the constraint level" framework.

6.2.1 Formal Basis

Lemma 6.4: Consider the following constraint satisfaction problem:

$$\begin{aligned} h(X) + w &\leq 0 \\ w &\geq h(\Lambda_j(X,\varepsilon)) - h(X) \qquad \cdots \quad (\$) \\ w &\geq 0 \end{aligned}$$

where $\varepsilon > 0$ is given. The $(n+1)$ - vector $\begin{bmatrix} X^* \\ w^* \end{bmatrix}$ is a solution to (\$) if and only if $h(X^*) \leq 0$, $h(\Lambda_j(X^*,\varepsilon)) \leq 0$ and $w^* \geq 0$.

Proof : (a) *if part*

Since $\begin{bmatrix} X^* \\ w^* \end{bmatrix}$ solves (\$) then $h(X^*)+w^* \leq 0$ and $w^* \geq 0$. This implies that $h(X^*) \leq 0$. Also $h(\Lambda_j(X^*,\varepsilon)) \leq h(X^*) + w^* \leq 0$.

(b) *only if part*

Choose $w^* = -h(X^*)$, then $h(X^*) + w^* = 0$ and $w^* \geq 0$.

Since $h(\Lambda_j(X^*,\varepsilon)) \leq 0$ then $h(\Lambda_j(X^*,\varepsilon)) + w^* \leq w^*$. But $w^* = -h(X^*)$. Hence, $h(\Lambda_j(X^*,\varepsilon)) - h(X^*) \leq w^*$.

Therefore, $\begin{bmatrix} X^* \\ w^* \end{bmatrix}$ is a solution to (\$). □

Corollary 6.2: Consider the following constraint satisfaction problem:

$$\begin{aligned} h(X) + w &\leq 0 \\ w &\geq h(\Lambda_j(X,\varepsilon)) - h(X) \qquad \cdots \quad (\$) \\ w &\geq 0 \end{aligned}$$

where $\varepsilon > 0$ is given. The $(n+1)$ - vector $\begin{bmatrix} X^* \\ w^* \end{bmatrix}$ is a solution to (\$) if and only if the PC $\lambda_j(X,X',\varepsilon) \wedge h(X) \leq 0 \wedge h(X') \leq 0$ is satisfied at the pair $(X^*, \Lambda_j(X^*,\varepsilon))$ and $w^* \geq 0$.

Proof : From Lemma 6.1, $\lambda_j(X^*, \Lambda_j(X^*,\varepsilon),\varepsilon)$ is TRUE. The remainder of the proof is a direct consequence of Lemma 6.4 and definitions given in section 3.1. □

Theorem 6.4: Consider the following constraint satisfaction problem:

$$\begin{array}{rcll} h_i(X) + w_i & \leq & 0 & \\ w_i & \geq & h_i(\Lambda_{a_i}(X, \varepsilon_i)) - h_i(X) & \cdots \quad (\$) \\ w_i & \geq & 0 & \\ i & = & 1, 2, \cdots, m & \end{array}$$

where $\varepsilon_i > 0$ is given for each $i \in I_m$. The $(n+m)$ - vector $\begin{bmatrix} X^* \\ w^* \end{bmatrix}$ where $w^* = [w_1^*, w_2^*, \cdots, w_m^*]$ is a solution to (\$) if and only if $h_i(X^*) \leq 0$, $h(\Lambda_{a_i}(X^*, \varepsilon_i)) \leq 0$ and $w_i^* \geq 0$ for $i = 1, 2, \cdots, m$.

Proof : The proof follows, for any particular $i \in I_m$ directly from Lemma 6.4.

□

Corollary 6.3: Consider the following constraint satisfaction problem:

$$\begin{array}{rcll} h_i(X) + w_i & \leq & 0 & \\ w_i & \geq & h_i(\Lambda_{a_i}(X, \varepsilon_i)) - h_i(X) & \cdots \quad (\$) \\ w_i & \geq & 0 & \\ i & = & 1, 2, \cdots, m & \end{array}$$

where $\varepsilon_i > 0$ is given for each $i \in I_m$. The $(n+m)$ - vector $\begin{bmatrix} X^* \\ w^* \end{bmatrix}$ where $w^* = [w_1^*, w_2^*, \cdots, w_m^*]$ is a solution to (\$) if and only if $S_e = \{\lambda_{a_i}(X, X', \varepsilon_i) \wedge h_i(X) \leq 0 \wedge h_i(X') \leq 0, i = 1, 2, \cdots, m\}$ is consistent at the feasible bag $\{X^*\} \bigcup \{\Lambda_{a_i}(X^*, \varepsilon_i) : i = 1, 2, \cdots, m\}$ and $w_i^* \geq 0$ for $i = 1, 2, \cdots, m$.

Proof : The proof, for any particular $i \in I_m$ follows directly from Corollary 6.2.

□

Theorem 6.5: Consider the following two sets of constraints:

$$S = \{h_i(X) \leq 0, i = 1, 2, \cdots, m\}$$

$$S_e = \{\lambda_{a_i}(X, X', \varepsilon_i) \wedge h_i(X) \leq 0 \wedge h_i(X') \leq 0, i = m+1, m+2, \cdots, p\}$$

where for every $i \in I_m$ $h_i(X) \leq 0$ is potentially sensitive to X_j for every $j \in P_{S_e}$.

Consider also the following constraint satisfaction problem:

$$\left.\begin{array}{rcl} h_i(X) + w_i & \leq & 0 \\ w_i & \geq & 0 \\ i & = & 1, \cdots, m \\ w_i & \geq & h_k(\Lambda_{a_i}(X, \varepsilon_i)) - h_k(X) \\ i & = & 1, \cdots, m \text{ and each } k \in (I_p - I_m) \end{array}\right] \cdots (\$)$$

$$\left.\begin{array}{rcl} h_i(X) + w_i & \leq & 0 \\ w_i & \geq & h_i(\Lambda_{a_i}(X, \varepsilon_i)) - h_i(X) \\ w_i & \geq & 0 \\ i & = & m+1, \cdots, p \end{array}\right] \cdots (\$\$)$$

where $\varepsilon_i > 0$ is given for each $i \in (I_p - I_m)$. The $(n+p)$ - vector $\begin{bmatrix} X^* \\ w^* \end{bmatrix}$ where $w^* = [w_1^*, w_2^*, \cdots, w_p^*]$ is a solution to the combination of ($) and ($$) if and only if $S \bigcup S_e$ is consistent at the feasible bag $\{X^*\} \bigcup \{\Lambda_{a_i}(X^*, \varepsilon_i) : i = m+1, m+2, \cdots, p\}$ and $w_i^* \geq 0$ for $i = 1, 2, \cdots, p$.

Proof : From Corollary 6.3, we have that the $(n+p-m)$ - vector $\begin{bmatrix} X^* \\ \bar{=} \\ w^* \end{bmatrix}$ where $\bar{=}$ $w^* = [w_{m+1}^*, w_{m+2}^*, \cdots, w_p^*]$ is a solution to ($$) if and only if $S_e = \{\lambda_{a_i}(X, X', \varepsilon) \wedge h(X) \leq 0 \wedge h(X') \leq 0, i = m+1, m+2, \cdots, p\}$ is consistent at the feasible bag $\{X^*\} \bigcup \{\Lambda_{a_i}(X^*, \varepsilon_i) : i = m+1, m+2, \cdots, p\}$. Therefore, we only need to show that the $(n+m)$ - vector $\begin{bmatrix} X^* \\ \bar{w}^* \end{bmatrix}$ where $\bar{w}^* = [w_1^*, w_2^*, \cdots, w_m^*]$ is a solution to ($) if and only if S is consistent at X^* and $\Lambda_{a_i}(X^*, \varepsilon_i)$ for $i = m+1, m+2, \cdots, p$.

(a) *if part*

Since $\begin{bmatrix} X^* \\ \bar{w}^* \end{bmatrix}$ solves ($) then for every $i = 1, 2, \cdots, m$ $h_i(X^*) + w_i^* \leq 0$ and $w_i^* \geq 0$. This implies that $h_i(X^*) \leq 0$ for every $i = 1, 2, \cdots, m$. That is, S is consistent at X^*.

Also $h_i(\Lambda_{a_k}(X^*, \varepsilon_k)) \leq h_i(X^*) + w_i^* \leq 0$ for every $i = 1, 2, \cdots, m$ and for each $k = m+1, m+2, \cdots, p$. That is, S is consistent at $\Lambda_{a_k}(X^*, \varepsilon_k)$ for $k = m+1, m+2, \cdots, p$.

(b) *only if part*

Choose $w_i^* = -h_i(X^*)$, then $h_i(X^*)+w_i^* = 0$ and $w_i^* \geq 0$ for every $i = 1, 2, \cdots, m$.

Since $h_i(\Lambda_{a_k}(X^*, \epsilon_k)) \leq 0$ then $h_i(\Lambda_{a_k}(X^*, \epsilon_k))+w_i^* \leq w_i^*$ for every $i = 1, 2, \cdots, m$ and for each $k = m+1, m+2, \cdots, p$. But $w_i^* = -h_i(X^*)$ for every $i = 1, 2, \cdots, m$. Hence, $h_i(\Lambda_{a_k}(X^*, \epsilon_k)) - h_i(X^*) \leq w_i^*$ for every $i = 1, 2, \cdots, m$ and for each $k = m+1, m+2, \cdots, p$.

Therefore, $\begin{bmatrix} X^* \\ \bar{w}^* \end{bmatrix}$ where $\bar{w}^* = [w_1^*, w_2^*, \cdots, w_m^*]$ is a solution to (\$). □

6.2.2 Transition Rule A

Based on the concept of slack at the constraint level, we now present a solution procedure for the problem P (section 5.1) for the case where $S_v \neq \emptyset$. This solution procedure is again formulated as a mixed integer programming problem. It consists of three steps which are given below.

Step (A.1) Rewrite the constraints in $S_f \bigcup S_c \bigcup S_v$ in the following form:

$$f_i(X) \leq 0;\ i = 1, \cdots, l'$$

$$\bar{f}_i(X) \leq 0;\ i = l'+1, \cdots, l$$

$$\tilde{f}_i(X) \leq 0;\ i = l'+1, \cdots, l$$

$$g_i(X) \leq 0;\ i = l+1, \cdots, m'$$

$$\bar{g}_i(X) \leq 0;\ i = m'+1, \cdots, m$$

$$\tilde{g}_i(X) \leq 0;\ i = m'+1, \cdots, m$$

$$\lambda_{a_i}(X, X', \epsilon_i) \wedge v_i(X) \leq 0 \wedge v_i(X') \leq 0;\ i = m+1, \cdots, p'$$

$$\lambda_{a_i}(X, X', \varepsilon_i) \wedge \bar{v}_i(X) \leq 0 \wedge \bar{v}_i(X') \leq 0;\ i = p' + 1, \cdots, p$$

$$\lambda_{a_i}(X, X', \varepsilon_i) \wedge \tilde{v}_i(X) \leq 0 \wedge \tilde{v}_i(X') \leq 0;\ i = p' + 1, \cdots, p$$

where $\tilde{f}_i(X) = -\bar{f}_i(X)$, $\tilde{g}_i(X) = -\bar{g}_i(X)$ and $\tilde{v}_i(X) = -\bar{v}_i(X)$.

Step (A.2) Transform the constraint satisfaction problem in step (A.1) into the following equivalent form: (equivalence is assured by Theorem 6.5)

$$\begin{aligned} f_i(X) + w_i &\leq 0 \\ w_i &\geq 0;\ i = 1, \cdots, l' \\ w_i &\geq f_i(\Lambda_{a_k}(X, \varepsilon_k)) - f_i(X);\ i = 1, \cdots, l' \text{ and each } k \in (I_p - I_m) \end{aligned}$$

$$\begin{aligned} \bar{f}_i(X) + w_i &\leq 0 \\ w_i &\geq 0;\ i = l' + 1, \cdots, l \\ w_i &\geq \bar{f}_i(\Lambda_{a_k}(X, \varepsilon_k)) - \bar{f}_i(X);\ i = l' + 1, \cdots, l \text{ and each } k \in (I_p - I_m) \end{aligned}$$

$$\begin{aligned} \tilde{f}_i(X) + \tilde{w}_i &\leq 0 \\ \tilde{w}_i &\geq 0;\ i = l' + 1, \cdots, m \\ \tilde{w}_i &\geq \tilde{f}_i(\Lambda_{a_k}(X, \varepsilon_k)) - \tilde{f}_i(X);\ i = l' + 1, \cdots, l \text{ and each } k \in (I_p - I_m) \end{aligned}$$

$$\begin{aligned} g_i(X) + w_i &\leq 0 \\ w_i &\geq 0;\ i = l + 1, \cdots, m' \\ w_i &\geq g_i(\Lambda_{a_k}(X, \varepsilon_k)) - g_i(X);\ i = l + 1, \cdots, m' \text{ and each } k \in (I_p - I_m) \end{aligned}$$

$$\begin{aligned} \bar{g}_i(X) + w_i &\leq 0 \\ w_i &\geq 0;\ i = m' + 1, \cdots, m \\ w_i &\geq \bar{g}_i(\Lambda_{a_k}(X, \varepsilon_k)) - \bar{g}_i(X);\ i = m' + 1, \cdots, m \text{ and each } k \in (I_p - I_m) \end{aligned}$$

$$\begin{aligned} \tilde{g}_i(X) + \tilde{w}_i &\leq 0 \\ \tilde{w}_i &\geq 0;\ i = m' + 1, \cdots, m \\ \tilde{w}_i &\geq \tilde{g}_i(\Lambda_{a_k}(X, \varepsilon_k)) - \tilde{g}_i(X);\ i = m' + 1, \cdots, m \text{ and each } k \in (I_p - I_m) \end{aligned}$$

$$\begin{aligned} v_i(X) + w_i &\leq 0 \\ w_i &\geq 0 \\ w_i &\geq v_i(\Lambda_{a_i}(X, \varepsilon_i)) - v_i(X);\ i = m + 1, \cdots, p' \end{aligned}$$

$$
\begin{aligned}
\bar{v}_i(X) + w_i &\leq 0 \\
w_i &\geq 0 \\
w_i &\geq \bar{v}_i(\Lambda_{a_i}(X, \varepsilon_i)) - \bar{v}_i(X);\ i = p'+1, \cdots, p \\
\tilde{v}_i(X) + \tilde{w}_i &\leq 0 \\
\tilde{w}_i &\geq 0 \\
\tilde{w}_i &\geq \tilde{v}_i(\Lambda_{a_i}(X, \varepsilon_i)) - \tilde{v}_i(X);\ i = p'+1, \cdots, p
\end{aligned}
$$

Refinement: For any $i \in I_{l'}$ if $f_i(X) \leq 0$ is insensitive to X_{a_k} for some $k \in (I_p - I_m)$ then the constraint $w_i \geq f_i(\Lambda_{a_k}(X, \varepsilon_k)) - f_i(X)$ can be deleted as a consequence of Lemma 6.3. Furthermore, for any $i \in I_{l'}$ if $f_i(X) \leq 0$ is insensitive to X_j for every $j \in P_{S_*}$ (i.e., for all $k \in (I_p - I_m)$ $w_i \geq f_i(\Lambda_{a_k}(X, \varepsilon_k)) - f_i(X)$ is deleted) then the slack variable w_i in the constraint $f_i(X) + w_i \leq 0$ and the constraint $w_i \geq 0$ can also be deleted because no longer exists a role for the slack variable w_i.

The above refinement procedure also applies to the constraints $\bar{f}_i(X) \leq 0$ for $i = l'+1, \cdots, l$, $\tilde{f}_i(X) \leq 0$ for $i = l'+1, \cdots, l$, $g_i(X) \leq 0$ for $i = l+1, \cdots, m'$, $\bar{g}_i(X) \leq 0$ for $i = m'+1, \cdots, m$ and $\tilde{g}_i(X) \leq 0$ for $i = m'+1, \cdots, m$.

Similarly, for any $i \in (I_{p'} - I_m)$ if $v_i(X) \leq 0$ is insensitive to X_{a_i} then the constraints $w_i \geq v_i(\Lambda_{a_i}(X, \varepsilon_i)) - v_i(X)$ and $w_i \geq 0$, and the slack variable w_i in the constraint $v_i(X) + w_i \leq 0$ can be deleted as a consequence of Theorem 6.3. This also applies to the constraints $\bar{v}_i(X) \leq 0$ for $i = p'+1, \cdots, p$ and $\tilde{v}_i(X) \leq 0$ for $i = p'+1, \cdots, p$.

Step (A.3) Formulate the following mixed integer programming problem (which we call the PMIP/A problem)

$$
\begin{array}{ll}
\min & M \sum_{U} C_i + \sum_{I_p - I_l - U} C_i \\
X \in \hat{\Re}_n & \\
w \in \Re_p^+ & \\
\tilde{w} \in \Re_m^+ & \\
C \in \Gamma &
\end{array}
$$

subject to:

$$
\begin{aligned}
f_i(X) + w_i &\leq 0 \\
w_i &\geq 0;\ i = 1, \cdots, l' \\
w_i + BC_k &\geq f_i(\Lambda_{a_k}(X, \varepsilon_k)) - f_i(X);\ i = 1, \cdots, l' \\
&\qquad \text{and each } k \in (I_p - I_m)
\end{aligned}
$$

$$
\begin{aligned}
\bar{f}_i(X) + w_i &\leq 0 \\
w_i &\geq 0;\ i = l' + 1, \cdots, l \\
w_i + BC_k &\geq \bar{f}_i(\Lambda_{a_k}(X, \varepsilon_k)) - \bar{f}_i(X);\ i = l' + 1, \cdots, l \\
&\qquad \text{and each } k \in (I_p - I_m)
\end{aligned}
$$

$$
\begin{aligned}
\tilde{f}_i(X) + \tilde{w}_i &\leq 0 \\
\tilde{w}_i &\geq 0;\ i = l' + 1, \cdots, m \\
\tilde{w}_i + BC_k &\geq \tilde{f}_i(\Lambda_{a_k}(X, \varepsilon_k)) - \tilde{f}_i(X);\ i = l' + 1, \cdots, l \\
&\qquad \text{and each } k \in (I_p - I_m)
\end{aligned}
$$

$$
\begin{aligned}
g_i(X) + w_i - BC_i &\leq 0 \\
w_i &\geq 0;\ i = l + 1, \cdots, m' \\
w_i + BC_k &\geq g_i(\Lambda_{a_k}(X, \varepsilon_k)) - g_i(X);\ i = l + 1, \cdots, m' \\
&\qquad \text{and each } k \in (I_p - I_m)
\end{aligned}
$$

$$
\begin{aligned}
\bar{g}_i(X) + w_i - BC_i &\leq 0 \\
w_i &\geq 0;\ i = m' + 1, \cdots, m \\
w_i + BC_k &\geq \bar{g}_i(\Lambda_{a_k}(X, \varepsilon_k)) - \bar{g}_i(X);\ i = m' + 1, \cdots, m \\
&\qquad \text{and each } k \in (I_p - I_m)
\end{aligned}
$$

$$
\begin{aligned}
\tilde{g}_i(X) + \tilde{w}_i - BC_i &\leq 0 \\
\tilde{w}_i &\geq 0;\ i = m' + 1, \cdots, m \\
\tilde{w}_i + BC_k &\geq \tilde{g}_i(\Lambda_{a_k}(X, \varepsilon_k)) - \tilde{g}_i(X);\ i = m' + 1, \cdots, m \\
&\qquad \text{and each } k \in (I_p - I_m)
\end{aligned}
$$

$$
\begin{aligned}
v_i(X) + w_i - BC_i &\leq 0 \\
w_i &\geq 0 \\
w_i + BC_i &\geq v_i(\Lambda_{a_i}(X, \varepsilon_i)) - v_i(X);\ i = m + 1, \cdots, p'
\end{aligned}
$$

$$
\begin{aligned}
\bar{v}_i(X) + w_i - BC_i &\leq 0 \\
w_i &\geq 0 \\
w_i + BC_i &\geq \bar{v}_i(\Lambda_{a_i}(X, \varepsilon_i)) - \bar{v}_i(X);\ i = p' + 1, \cdots, p
\end{aligned}
$$

$$\begin{array}{rcrcl} \tilde{v}_i(X) + \tilde{w}_i & - & BC_i & \leq & 0 \\ \tilde{w}_i & & & \geq & 0 \\ \tilde{w}_i & + & BC_i & \geq & \tilde{v}_i(\Lambda_{a_i}(X, \varepsilon_i)) - \tilde{v}_i(X);\ i = p' + 1, \cdots, p \end{array}$$

with

$$B > \max_{X \in \tilde{\mathfrak{X}}_n} \{ \mid f_i(X) \mid, \mid \bar{f}_i(X) \mid, \mid g_i(X) \mid, \mid \bar{g}_i(X) \mid, \mid v_i(X) \mid, \mid \bar{v}_i(X) \mid : i = 1, 2, \cdots, p \}$$

$$M > \mid I_p - I_l - U \mid$$

$$\bar{m} = (l - l') + (m - m') + (p - p')$$

Γ is the set of all possible $(p - l)$ - vectors whose entries are either 0 or 1. The binary variables C_i, $i = l + 1, \cdots, p$ are called the *constraint satisfaction indicators.* These variables are the components of the $(p - l)-$ vector C.

It should be observed that a solution to PMIP/A problem in this "slack at the constraint level" framework also satisfies properties which are analogous to those established for the PMIP problem in Theorems 5.1 and 5.2 (section 5.1). This can be shown by identifying the vector $\begin{bmatrix} X \\ w \\ \tilde{w} \end{bmatrix}$ of the PMIP/A problem with the X vector of the PMIP problem.

6.2.3 Example

We illustrate the application of Transition Rule A with the help of an example in which the constraints are linear. Observe that in this case, $f_i(\Lambda_{a_k}(X, \varepsilon_k)) - f_i(X) = \alpha_{ia_k}\varepsilon_k$. This observation equally applicable to $\bar{f}_i(X)$, $\tilde{f}_i(X)$, $g_i(X)$, $\bar{g}_i(X)$, $\tilde{g}_i(X)$, $v_i(X)$, $\bar{v}_i(X)$ and $\tilde{v}_i(X)$.

Consider the following set of constraints:

$$
\begin{array}{lrcrcr}
c_1: & 2X_1 & & & \leq & 41 \\
c_2: & & & 3X_2 & \leq & 65 \\
c_3: & 4X_1 & + & 3X_2 & \leq & 12 \\
c_4: & -2X_1 & + & X_2 & \geq & 2 \\
c_5: & -X_1 & + & X_2 & \geq & 6 \\
c_6: & 16X_1 & + & 17X_2 & \geq & 272 \\
c_7: & 2X_1 & - & 7X_2 & \geq & 14
\end{array}
$$

$$
\begin{array}{lrcrcr}
c_8: & \lambda_1(X, X', \varepsilon_8) & \wedge & 2X_1 - 5X_2 \leq 16 & \wedge & 2X_1' - 5X_2' \leq 16 \\
c_9: & \lambda_2(X, X', \varepsilon_9) & \wedge & 9X_1 \leq 46 & \wedge & 9X_1' \leq 46 \\
c_{10}: & \lambda_1(X, X', \varepsilon_{10}) & \wedge & 5X_1 + 8X_2 \leq 45 & \wedge & 5X_1' + 8X_2' \leq 45
\end{array}
$$

where:

$(X_1, X_2) \in \hat{\Re}_2 \subset \hat{\Re}_2^+$, $\{c_1, c_2\} = S_f$, $\{c_3, c_4, c_5, c_6, c_7\} = S_c$ and $\{c_8, c_9, c_{10}\} = S_v$.

$\varepsilon_8 > 0$, $\varepsilon_9 > 0$ and $\varepsilon_{10} > 0$ (user defined).

With respect to the general problem given in section 6.2.2, $l' = l = 2$, $m' = m = 7$ and $p' = p = 10$.

(E.1) Apply step (A.1) as outlined in section 6.2.2:

$$
\begin{array}{lrcrcrcr}
c_1: & 2X_1 & & & - & 41 & \leq & 0 \\
c_2: & & & 3X_2 & - & 65 & \leq & 0 \\
c_3: & 4X_1 & + & 3X_2 & - & 12 & \leq & 0 \\
c_4: & 2X_1 & - & X_2 & + & 2 & \leq & 0 \\
c_5: & X_1 & - & X_2 & + & 6 & \leq & 0 \\
c_6: & -16X_1 & - & 17X_2 & + & 272 & \leq & 0 \\
c_7: & -2X_1 & + & 7X_2 & + & 14 & \leq & 0
\end{array}
$$

$$
\begin{array}{lrcl}
c_8: & \lambda_1(X,X',\varepsilon_8) & \wedge & 2X_1 - 5X_2 - 16 \leq 0 \ \wedge \ 2X_1' - 5X_2' - 16 \leq 0 \\
c_9: & \lambda_2(X,X',\varepsilon_9) & \wedge & 9X_1 - 46 \leq 0 \ \wedge \ 9X_1' - 46 \leq 0 \\
c_{10}: & \lambda_1(X,X',\varepsilon_{10}) & \wedge & 5X_1 + 8X_2 - 45 \leq 0 \ \wedge \ 5X_1' + 8X_2' - 45 \leq 0
\end{array}
$$

(E.2) Apply step (A.2) as outlined in section 6.2.2:

$$
\begin{array}{rcrcrcrcr}
2X_1 & & & - & 41 & + & w_1 & \leq & 0 \\
 & & & & & & w_1 & \geq & 0 \\
 & & 3X_2 & - & 65 & + & w_2 & \leq & 0 \\
 & & & & & & w_2 & \geq & 0 \\
 & & & & & & w_1 & \geq & 2\varepsilon_8 \\
 & & & & & & w_1 & \geq & 2\varepsilon_{10} \\
 & & & & & & w_2 & \geq & 3\varepsilon_9 \\
4X_1 & + & 3X_2 & - & 12 & + & w_3 & \leq & 0 \\
 & & & & & & w_3 & \geq & 0 \\
2X_1 & - & X_2 & + & 2 & + & w_4 & \leq & 0 \\
 & & & & & & w_4 & \geq & 0 \\
X_1 & - & X_2 & + & 6 & + & w_5 & \leq & 0 \\
 & & & & & & w_5 & \geq & 0 \\
-16X_1 & - & 17X_2 & + & 272 & & & \leq & 0 \\
-2X_1 & + & 7X_2 & + & 14 & + & w_7 & \leq & 0 \\
 & & & & & & w_7 & \geq & 0 \\
 & & & & & & w_3 & \geq & 4\varepsilon_8 \\
 & & & & & & w_3 & \geq & 3\varepsilon_9 \\
 & & & & & & w_3 & \geq & 4\varepsilon_{10} \\
 & & & & & & w_4 & \geq & 2\varepsilon_8 \\
 & & & & & & w_4 & \geq & 2\varepsilon_{10} \\
 & & & & & & w_5 & \geq & \varepsilon_8 \\
 & & & & & & w_5 & \geq & \varepsilon_{10} \\
 & & & & & & w_7 & \geq & 7\varepsilon_9 \\
2X_1 & - & 5X_2 & - & 16 & + & w_8 & \leq & 0 \\
 & & & & & & w_8 & \geq & 0 \\
 & & & & & & w_8 & \geq & 2\varepsilon_8 \\
9X_1 & & & - & 46 & & & \leq & 0 \\
5X_1 & + & 8X_2 & - & 45 & + & w_{10} & \leq & 0 \\
 & & & & & & w_{10} & \geq & 0 \\
 & & & & & & w_{10} & \geq & 5\varepsilon_{10}
\end{array}
$$

(E.3) Apply step (A.3) as outlined in section 6.2.2:

$$\min_{\substack{X \in \hat{\Re}_2^+ \\ w \in \hat{\Re}_8^+ \\ C \in \Gamma}} \quad M \sum_{U} C_i + \sum_{I_p - I_l - U} C_i$$

subject to:

$$\begin{array}{rcrcrcrcrcl}
2X_1 & & & - & 41 & + & w_1 & & & \leq & 0 \\
 & & & & & & w_1 & & & \geq & 0 \\
 & & 3X_2 & - & 65 & + & w_2 & & & \leq & 0 \\
 & & & & & & w_2 & & & \geq & 0 \\
 & & & & & & w_1 & + & BC_8 & \geq & 2\varepsilon_8 \\
 & & & & & & w_1 & + & BC_{10} & \geq & 2\varepsilon_{10} \\
 & & & & & & w_2 & + & BC_9 & \geq & 3\varepsilon_9 \\
4X_1 & + & 3X_2 & - & 12 & + & w_3 & - & BC_3 & \leq & 0 \\
 & & & & & & w_3 & & & \geq & 0 \\
2X_1 & - & X_2 & + & 2 & + & w_4 & - & BC_4 & \leq & 0 \\
 & & & & & & w_4 & & & \geq & 0 \\
X_1 & - & X_2 & + & 6 & + & w_5 & - & BC_5 & \leq & 0 \\
 & & & & & & w_5 & & & \geq & 0 \\
-16X_1 & - & 17X_2 & + & 272 & & & - & BC_6 & \leq & 0 \\
-2X_1 & + & 7X_2 & + & 14 & + & w_7 & - & BC_7 & \leq & 0 \\
 & & & & & & w_7 & & & \geq & 0 \\
 & & & & & & w_3 & + & BC_8 & \geq & 4\varepsilon_8 \\
 & & & & & & w_3 & + & BC_9 & \geq & 3\varepsilon_9 \\
 & & & & & & w_3 & + & BC_{10} & \geq & 4\varepsilon_{10} \\
 & & & & & & w_4 & + & BC_8 & \geq & 2\varepsilon_8 \\
 & & & & & & w_4 & + & BC_{10} & \geq & 2\varepsilon_{10} \\
 & & & & & & w_5 & + & BC_8 & \geq & \varepsilon_8 \\
 & & & & & & w_5 & + & BC_{10} & \geq & \varepsilon_{10} \\
 & & & & & & w_7 & + & BC_9 & \geq & 7\varepsilon_9 \\
2X_1 & - & 5X_2 & - & 16 & + & w_8 & - & BC_8 & \leq & 0 \\
 & & & & & & w_8 & & & \geq & 0 \\
 & & & & & & w_8 & + & BC_8 & \geq & 2\varepsilon_8 \\
9X_1 & & & - & 46 & & & - & BC_9 & \leq & 0 \\
5X_1 & + & 8X_2 & - & 45 & + & w_{10} & - & BC_{10} & \leq & 0 \\
 & & & & & & w_{10} & & & \geq & 0 \\
 & & & & & & w_{10} & + & BC_{10} & \geq & 5\varepsilon_{10}
\end{array}$$

with

$$B > \max_{X \in \hat{\mathfrak{X}}_2^+} \{ \mid 2X_1 - 41 \mid, \mid 3X_2 - 65 \mid, \mid 4X_1 + 3X_2 - 12 \mid, \mid 2X_1 - X_2 + 2 \mid \mid X_1 - X_2 + 6 \mid, \mid -16X_1 - 17X_2 + 272 \mid, \mid -2X_1 + 7X_2 + 14 \mid, \mid 2X_1 - 5X_2 - 16 \mid, \mid 9X_1 - 46 \mid, \mid 5X_1 + 8X_2 - 45 \mid \}$$

$$M > \mid I_p - I_l - U \mid$$

Γ is the set of all possible $(p - l)$ - vectors whose entries are either 0 or 1.

The discussion of Example 15 in Appendix C shows that the C^3 algorithm generates the consistent cover $\tilde{E} = \{\{1,2,4,5,6,8,9\}, \{1,2,3,8,9,10\}, \{1,2,7,8,10\}\}$ for this problem. The optimality of this result is also demonstrated in Appendix C.

6.3 Slack at the Variable Level

In this section, we develop the PC analysis from the perspective of slack at the variable level. As in the previous section, the presentation begins with the analysis of a single PC (Lemma 6.5 and Corollary 6.4) which is then extended to a set of PC's (Theorem 6.6 and Corollary 6.5) and to the case where a set of PC's is combined with a non-empty $S_f \bigcup S_c$ (Theorem 6.7). Finally, we incorporate this formal analysis into Transition Rule B for handling PC's using the "slack at the variable level" framework.

6.3.1 Formal Basis

<u>Lemma 6.5:</u> Consider the following constraint satisfaction problem:

$$\begin{aligned} h(X) + w &\leq 0 \\ w &\geq h(\Lambda_j(X, \Delta)) - h(X) \qquad \cdots \quad (\$) \\ w &\geq 0 \\ \Delta &\geq \varepsilon \end{aligned}$$

where $\varepsilon > 0$ is given. The $(n+2)$ - vector $\begin{bmatrix} X^* \\ w^* \\ \Delta^* \end{bmatrix}$ is a solution to (\$) if and only if $h(X^*) \leq 0$, $h(\Lambda_j(X^*, \Delta^*)) \leq 0$, $w^* \geq 0$ and $\Delta^* \geq 0$.

<u>Proof :</u> (a) *if part*

Since $\begin{bmatrix} X^* \\ w^* \\ \Delta^* \end{bmatrix}$ solves (\$) then $h(X^*) + w^* \leq 0$ and $w^* \geq 0$. This implies that $h(X^*) \leq 0$. Also $h(\Lambda_j(X^*, \Delta^*)) \leq h(X^*) + w^* \leq 0$.

(b) *only if part*

Choose $w^* = -h(X^*)$, then $h(X^*) + w^* = 0$ and $w^* \geq 0$.

Choose $\Delta^* = \varepsilon$.

Since $h(\Lambda_j(X^*, \Delta^*)) \leq 0$ then $h(\Lambda_j(X^*, \Delta^*)) + w^* \leq w^*$. But $w^* = -h(X^*)$. Hence, $h(\Lambda_j(X^*, \Delta^*)) - h(X^*) \leq w^*$.

Therefore, $\begin{bmatrix} X^* \\ w^* \\ \Delta^* \end{bmatrix}$ is a solution to (\$). □

Corollary 6.4: Consider the following constraint satisfaction problem:

$$\begin{array}{rcll} h(X) + w & \leq & 0 & \\ w & \geq & h(\Lambda_j(X, \Delta)) - h(X) & \cdots \quad (\$) \\ w & \geq & 0 & \\ \Delta & \geq & \varepsilon & \end{array}$$

where $\varepsilon > 0$ is given. The $(n+2)$ - vector $\begin{bmatrix} X^* \\ w^* \\ \Delta^* \end{bmatrix}$ is a solution to (\$) if and only if the PC $\lambda_j(X, X', \varepsilon) \wedge h(X) \leq 0 \wedge h(X') \leq 0$ is satisfied at the pair $(X^*, \Lambda_j(X^*, \Delta^*))$, $w^* \geq 0$ and $\Delta^* \geq 0$.

Proof : From Lemma 6.1, $\lambda_j(X^*, \Lambda_j(X^*, \Delta^*), \varepsilon)$ is TRUE. The remainder of the proof is a direct consequence of Lemma 6.5 and definitions given in section 3.1. □

Theorem 6.6: Consider the following constraint satisfaction problem:

$$\begin{array}{rcll} h_i(X) + w_i & \leq & 0 & \\ w_i & \geq & h_i(\Lambda_{a_i}(X, \Delta_{a_i})) - h_i(X) & \cdots \quad (\$) \\ w_i & \geq & 0 & \\ \Delta_{a_i} & \geq & \varepsilon_i & \\ i & = & 1, 2, \cdots, m & \end{array}$$

where $\varepsilon_i > 0$ is given for each $i \in I_m$. Let $V = \{a_i : i \in I_m\}$. The $(n + m + \mid V \mid)$ - vector $\begin{bmatrix} X^* \\ w^* \\ \Delta^* \end{bmatrix}$ is a solution to (\$) if and only if $h_i(X^*) \leq 0$, $h_i(\Lambda_{a_i}(X^*, \Delta^*_{a_i})) \leq 0$, $w_i^* \geq 0$ and $\Delta^*_{a_i} \geq 0$ for $i = 1, 2, \cdots, m$.

Proof : The proof follows, for any particular $i \in I_m$ directly from Lemma 6.5. □

Corollary 6.5: Consider the following constraint satisfaction problem:

$$\begin{array}{rcll} h_i(X) + w_i & \leq & 0 & \\ w_i & \geq & h_i(\Lambda_{a_i}(X, \Delta_{a_i})) - h_i(X) & \cdots (\$) \\ w_i & \geq & 0 & \\ \Delta_{a_i} & \geq & \varepsilon_i & \\ i & = & 1, 2, \cdots, m & \end{array}$$

where $\varepsilon_i > 0$ is given for each $i \in I_m$. The $(n+m+ \mid P_{S_e} \mid)$ - vector $\begin{bmatrix} X^* \\ w^* \\ \Delta^* \end{bmatrix}$ is a solution to (\$) if and only if $S_e = \{\lambda_{a_i}(X, X', \varepsilon_i) \wedge h_i(X) \leq 0 \wedge h_i(X') \leq 0, i = 1, 2, \cdots, m\}$ is consistent at the feasible bag $\{X^*\} \bigcup \{\Lambda_{a_i}(X^*, \Delta^*_{a_i}) : a_i \in P_{S_e}\}$, $w_i^* \geq 0$ and $\Delta^*_{a_i} \geq 0$ for $i = 1, 2, \cdots, m$.

Proof : The proof, for any particular $i \in I_m$ follows directly from Corollary 6.4.

□

Theorem 6.7: Consider the following two sets of constraints:

$$S = \{h_i(X) \leq 0, i = 1, 2, \cdots, m\}$$

$$S_e = \{\lambda_{a_i}(X, X', \varepsilon_i) \wedge h_i(X) \leq 0 \wedge h_i(X') \leq 0, i = m+1, m+2, \cdots, p\}$$

where for every $i \in I_m$ $h_i(X) \leq 0$ is potentially sensitive to X_j for every $j \in P_{S_e}$. Consider also the following constraint satisfaction problem:

$$\left.\begin{array}{rcll} h_i(X) + \sum_{j \in P_{S_e}} \bar{w}_{ij} & \leq & 0 & \\ i & = & 1, \cdots, m & \\ \bar{w}_{ij} & \geq & h_i(\Lambda_j(X, \Delta_j)) - h_i(X) & \\ \bar{w}_{ij} & \geq & 0 & \\ i & = & 1, \cdots, m \text{ and each } j \in P_{S_e} & \end{array}\right\} \cdots (\$)$$

$$\left.\begin{array}{rcll} h_i(X) + w_i & \leq & 0 & \\ w_i & \geq & h_i(\Lambda_{a_i}(X, \Delta_{a_i})) - h_i(X) & \\ w_i & \geq & 0 & \\ \Delta_{a_i} & \geq & \varepsilon_i & \\ i & = & m+1, \cdots, p & \end{array}\right\} \cdots (\$\$)$$

where $\varepsilon_i > 0$ is given for each $i \in (I_p - I_m)$. If the $(n + (p-m) + \bar{m} + m \cdot \bar{m})$ - vector $\begin{bmatrix} X^* \\ w^* \\ \Delta^* \\ \bar{w}^* \end{bmatrix}$ where $w^* = [w^*_{m+1}, w^*_{m+2}, \cdots, w^*_p]$ and $\bar{w}^* = [\bar{w}^*_{1,1}, \cdots, \bar{w}^*_{1,\bar{m}}, \cdots, \bar{w}^*_{m,1}, \cdots, \bar{w}^*_{m,\bar{m}}]$ is a solution to the combination of (\$) and (\$\$) then $S \bigcup S_e$ is consistent at the feasible bag $\{X^*\} \bigcup \{\Lambda_j(X^*, \Delta^*_j) : j \in P_{S_e}\}$ where $\bar{m} = \mid P_{S_e} \mid$.

Proof : Because $\begin{bmatrix} X^* \\ w^* \\ \Delta^* \end{bmatrix}$ is a solution to (\$\$) it follows from Corollary 6.5 that S_e is consistent at the feasible bag $\{X^*\} \bigcup \{\Lambda_j(X^*, \Delta^*_j) : j \in P_{S_e}\}$. Therefore, we only need to show that S is consistent at X^* and $\Lambda_j(X^*, \Delta^*_j)$ for every $j \in P_{S_e}$.

Since $\begin{bmatrix} X^* \\ \Delta^* \\ \bar{w}^* \end{bmatrix}$ solves (\$) then for every $i = 1, 2, \cdots, m$ $h_i(X^*) + \sum_{j \in P_{S_e}} \bar{w}^*_{ij} \leq 0$ and $\bar{w}^*_{ij} \geq 0$ for each $j \in P_{S_e}$. This implies that $h_i(X^*) \leq 0$ for every $i = 1, 2, \cdots, m$. That is, S is consistent at X^*.

Also $h_i(\Lambda_j(X^*, \Delta^*_j)) \leq h_i(X^*) + \bar{w}^*_{ij} \leq 0$ for every $i = 1, 2, \cdots, m$ and for each $j \in P_{S_e}$. That is, S is consistent at $\Lambda_j(X^*, \Delta^*_j)$ for every $j \in P_{S_e}$. □

6.3.2 Transition Rule B

Based on the concept of slack at the variable level, we now present an alternate solution procedure for problem P (section 5.1) for the case where $S_v \neq \emptyset$.

Step (B.1) This step is same as step (A.1) of Transition Rule A.

Step (B.2) Transform the constraint satisfaction problem in step (B.1) into the following equivalent form: (equivalence is assured by Theorem 6.7)

$$\begin{aligned} f_i(X) + \textstyle\sum_{j \in P_{S_v}} \bar{w}_{ij} &\leq 0;\ i = 1, \cdots, l' \\ \bar{w}_{ij} &\geq f_i(\Lambda_j(X, \Delta_j)) - f_i(X) \\ \bar{w}_{ij} &\geq 0;\ i = 1, \cdots, l' \text{ and each } j \in P_{S_v} \end{aligned}$$

$$
\begin{aligned}
\bar{f}_i(X) + \textstyle\sum_{j \in P_{S_v}} \bar{w}_{ij} &\leq 0;\ i = l'+1, \cdots, l \\
\bar{w}_{ij} &\geq \bar{f}_i(\Lambda_j(X, \Delta_j)) - \bar{f}_i(X) \\
\bar{w}_{ij} &\geq 0;\ i = l'+1, \cdots, l \text{ and each } j \in P_{S_v}
\end{aligned}
$$

$$
\begin{aligned}
\tilde{f}_i(X) + \textstyle\sum_{j \in P_{S_v}} \tilde{w}_{ij} &\leq 0;\ i = l'+1, \cdots, l \\
\tilde{w}_{ij} &\geq \tilde{f}_i(\Lambda_j(X, \Delta_j)) - \tilde{f}_i(X) \\
\tilde{w}_{ij} &\geq 0;\ i = l'+1, \cdots, l \text{ and each } j \in P_{S_v}
\end{aligned}
$$

$$
\begin{aligned}
g_i(X) + \textstyle\sum_{j \in P_{S_v}} \bar{w}_{ij} &\leq 0;\ i = l+1, \cdots, m' \\
\bar{w}_{ij} &\geq g_i(\Lambda_j(X, \Delta_j)) - g_i(X) \\
\bar{w}_{ij} &\geq 0;\ i = l+1, \cdots, m' \text{ and each } j \in P_{S_v}
\end{aligned}
$$

$$
\begin{aligned}
\bar{g}_i(X) + \textstyle\sum_{j \in P_{S_v}} \bar{w}_{ij} &\leq 0;\ i = m'+1, \cdots, m \\
\bar{w}_{ij} &\geq \bar{g}_i(\Lambda_j(X, \Delta_j)) - \bar{g}_i(X) \\
\bar{w}_{ij} &\geq 0;\ i = m'+1, \cdots, m \text{ and each } j \in P_{S_v}
\end{aligned}
$$

$$
\begin{aligned}
\tilde{g}_i(X) + \textstyle\sum_{j \in P_{S_v}} \tilde{w}_{ij} &\leq 0;\ i = m'+1, \cdots, m \\
\tilde{w}_{ij} &\geq \tilde{g}_i(\Lambda_j(X, \Delta_j)) - \tilde{g}_i(X) \\
\tilde{w}_{ij} &\geq 0;\ i = m'+1, \cdots, m \text{ and each } j \in P_{S_v}
\end{aligned}
$$

$$
\begin{aligned}
v_i(X) + w_i &\leq 0 \\
w_i &\geq v_i(\Lambda_{a_i}(X, \Delta_{a_i})) - v_i(X) \\
w_i &\geq 0 \\
\Delta_{a_i} &\geq \varepsilon_i;\ i = m+1, \cdots, p'
\end{aligned}
$$

$$
\begin{aligned}
\bar{v}_i(X) + w_i &\leq \\
w_i &\geq \bar{v}_i(\Lambda_{a_i}(X, \Delta_{a_i})) - \bar{v}_i(X) \\
w_i &\geq 0 \\
\Delta_{a_i} &\geq \varepsilon_i;\ i = p'+1, \cdots, p
\end{aligned}
$$

$$
\begin{aligned}
\tilde{v}_i(X) + \tilde{w}_i &\leq 0 \\
\tilde{w}_i &\geq \tilde{v}_i(\Lambda_{a_i}(X, \Delta_{a_i})) - \tilde{v}_i(X) \\
\tilde{w}_i &\geq 0;\ i = p'+1, \cdots, p
\end{aligned}
$$

Refinement: For any $i \in I_{l'}$ if $f_i(X) \leq 0$ is insensitive to X_j for some $j \in P_{S_v}$ then the constraints $\bar{w}_{ij} \geq f_i(\Lambda_j(X, \Delta_j)) - f_i(X)$ and $\bar{w}_{ij} \geq 0$, and the slack variable $\bar{w}_{ij}$ in the constraint $f_i(X) + \sum_{j \in P_{S_v}} \bar{w}_{ij} \leq 0$ can be deleted as a consequence of

Lemma 6.3. This refinement procedure also applies to the constraints $\bar{f}_i(X) \leq 0$ for $i = l'+1, \cdots, l$, $\tilde{f}_i(X) \leq 0$ for $i = l'+1, \cdots, l$, $g_i(X) \leq 0$ for $i = l+1, \cdots, m'$, $\bar{g}_i(X) \leq 0$ for $i = m'+1, \cdots, m$ and $\tilde{g}_i(X) \leq 0$ for $i = m'+1, \cdots, m$.

Similarly, for any $i \in (I_{p'} - I_m)$ if $v_i(X) \leq 0$ is insensitive to X_{a_i} then the constraints $w_i \geq v_i(\Lambda_{a_i}(X, \Delta_{a_i})) - v_i(X)$ and $w_i \geq 0$, and the slack variable w_i in the constraint $v_i(X) + w_i \leq 0$ can be deleted as a consequence of Theorem 6.3. This also applies to the constraints $\bar{v}_i(X) \leq 0$ for $i = p'+1, \cdots, p$ and $\tilde{v}_i(X) \leq 0$ for $i = p'+1, \cdots, p$.

Step (B.3) Formulate the following mixed integer programming problem (which we call the PMIP/B problem)

$$\begin{array}{ll} \min & M \sum_{U} C_i + \sum_{I_p - I_l - U} C_i \\ X \in \hat{\Re}_n & \\ \bar{w} \in \Re^{+}_{\bar{m}} & \\ \tilde{w} \in \Re^{+}_{\tilde{m}} & \\ w \in \Re^{+}_{r} & \\ \tilde{w} \in \Re^{+}_{s} & \\ \Delta \in \Re^{+}_{m} & \\ C \in \Gamma & \end{array}$$

subject to:

$$\begin{array}{rcrcl} f_i(X) + \sum_{j \in P_{S_v}} \bar{w}_{ij} & & & \leq & 0;\ i = 1, \cdots, l' \\ & & \bar{w}_{ij} + BC_j & \geq & f_i(\Lambda_j(X, \Delta_j)) - f_i(X) \\ & & \bar{w}_{ij} & \geq & 0;\ i = 1, \cdots, l' \text{ and each } j \in P_{S_v} \end{array}$$

$$\begin{array}{rcrcl} \bar{f}_i(X) + \sum_{j \in P_{S_v}} \bar{w}_{ij} & & & \leq & 0;\ i = l'+1, \cdots, l \\ & & \bar{w}_{ij} + BC_j & \geq & \bar{f}_i(\Lambda_j(X, \Delta_j)) - \bar{f}_i(X) \\ & & \bar{w}_{ij} & \geq & 0;\ i = l'+1, \cdots, l \text{ and each } j \in P_{S_v} \end{array}$$

$$\begin{array}{rcrcl} \tilde{f}_i(X) + \sum_{j \in P_{S_v}} \tilde{w}_{ij} & & & \leq & 0;\ i = l'+1, \cdots, l \\ & & \tilde{w}_{ij} + BC_j & \geq & \tilde{f}_i(\Lambda_j(X, \Delta_j)) - \tilde{f}_i(X) \\ & & \tilde{w}_{ij} & \geq & 0;\ i = l'+1, \cdots, l \text{ and each } j \in P_{S_v} \end{array}$$

$$
\begin{array}{rcrcl}
g_i(X) + \sum_{j \in P_{S_v}} \bar{w}_{ij} & - & BC_i & \leq & 0;\ i = m+1, \cdots, m' \\
\bar{w}_{ij} & + & BC_j & \geq & g_i(\Lambda_j(X, \Delta_j)) - g_i(X) \\
 & & \bar{w}_{ij} & \geq & 0;\ i = m+1, \cdots, m' \text{ and each } j \in P_{S_v}
\end{array}
$$

$$
\begin{array}{rcrcl}
\bar{g}_i(X) + \sum_{j \in P_{S_v}} \bar{w}_{ij} & - & BC_i & \leq & 0;\ i = m'+1, \cdots, m \\
\bar{w}_{ij} & + & BC_j & \geq & \bar{g}_i(\Lambda_j(X, \Delta_j)) - \bar{g}_i(X) \\
 & & \bar{w}_{ij} & \geq & 0;\ i = m'+1, \cdots, m \text{ and each } j \in P_{S_v}
\end{array}
$$

$$
\begin{array}{rcrcl}
\tilde{g}_i(X) + \sum_{j \in P_{S_v}} \tilde{w}_{ij} & - & BC_i & \leq & 0;\ i = m'+1, \cdots, m \\
\tilde{w}_{ij} & + & BC_j & \geq & \tilde{g}_i(\Lambda_j(X, \Delta_j)) - \tilde{g}_i(X) \\
 & & \tilde{w}_{ij} & \geq & 0;\ i = m'+1, \cdots, m \text{ and each } j \in P_{S_v}
\end{array}
$$

$$
\begin{array}{rcrcl}
v_i(X) + w_i & - & BC_i & \leq & 0 \\
w_i & + & BC_i & \geq & v_i(\Lambda_{a_i}(X, \Delta_{a_i})) - v_i(X) \\
 & & w_i & \geq & 0 \\
\Delta_{a_i} & + & BC_i & \geq & \varepsilon_i;\ i = m+1, \cdots, p'
\end{array}
$$

$$
\begin{array}{rcrcl}
\bar{v}_i(X) + w_i & - & BC_i & \leq & 0 \\
w_i & + & BC_i & \geq & \bar{v}_i(\Lambda_{a_i}(X, \Delta_{a_i})) - \bar{v}_i(X) \\
 & & w_i & \geq & 0 \\
\Delta_{a_i} & + & BC_i & \geq & \varepsilon_i;\ i = p'+1, \cdots, p
\end{array}
$$

$$
\begin{array}{rcrcl}
\tilde{v}_i(X) + \tilde{w}_i & - & BC_i & \leq & 0 \\
\tilde{w}_i & + & BC_i & \geq & \tilde{v}_i(\Lambda_{a_i}(X, \Delta_{a_i})) - \tilde{v}_i(X) \\
 & & \tilde{w}_i & \geq & 0;\ i = p'+1, \cdots, p
\end{array}
$$

with

$$
B > \max_{X \in \tilde{\mathfrak{X}}_n} \{ \mid f_i(X) \mid, \mid \bar{f}_i(X) \mid, \mid g_i(X) \mid, \mid \bar{g}_i(X) \mid, \mid v_i(X) \mid, \mid \bar{v}_i(X) \mid : i = 1, 2, \cdots, p \}
$$

$$
M > \mid I_p - I_l - U \mid
$$

$$
\bar{m} = \mid P_{S_v} \mid, \tilde{m} = m \cdot \bar{m}, \bar{n} = (l - l' + m - m') \cdot \bar{m}, r = p - m \text{ and } s = p - p'
$$

Γ is the set of all possible $(p - l)$ - vectors whose entries are either 0 or 1. The binary variables C_i, $i = l+1, \cdots, p$ are called the *constraint satisfaction indicators.* These variables are the components of the $(p - l)$– vector C.

It should be observed that a solution to the PMIP/B problem in this "slack at the variable level" framework also satisfies properties which are analogous to those established for the PMIP problem in Theorems 5.1 and 5.2 (section 5.1). This can be shown by identifying the vector $\begin{bmatrix} X \\ \bar{w} \\ \tilde{w} \\ w \\ \tilde{\tilde{w}} \\ \Delta \end{bmatrix}$ of the PMIP/B problem with the X vector of the PMIP problem.

An extensive set of computational results with the C^3 algorithm using the solution procedure PMIP/B are given in Appendix C.

6.3.3 Example

We illustrate the application of Transition Rule B with the same example used in the application of Transition Rule A. Observe that in this case $f_i(\Lambda_j(X, \Delta_j)) - f_i(X) = \alpha_{ij}\Delta_j$. Furthermore, because $\Delta_j \geq \epsilon_i > 0$ and $\frac{\partial f_i}{\partial X_j} = \alpha_{ij} > 0$ (by assumption $f_i(X)$ is potentially sensitive to X_j) it follows that $\alpha_{ij}\Delta_j > 0$. A reasonable choice for $\bar{w}_{ij}$ is $\alpha_{ij}\Delta_j$. This in particular allows the elimination of the constraints $\bar{w}_{ij} \geq f_i(\Lambda_j(X, \Delta_j)) - f_i(X)$ and $\bar{w}_{ij} \geq 0$.

Similar observations apply to $\bar{f}_i(X)$, $\tilde{f}_i(X)$, $g_i(X)$, $\bar{g}_i(X)$, $\tilde{g}_i(X)$, $v_i(X)$, $\bar{v}_i(X)$ and $\tilde{v}_i(X)$.

Consider the following set of constraints:

$$
\begin{array}{lrcrcr}
c_1: & 2X_1 & & & \leq & 41 \\
c_2: & & & 3X_2 & \leq & 65 \\
c_3: & 4X_1 & + & 3X_2 & \leq & 12 \\
c_4: & -2X_1 & + & X_2 & \geq & 2 \\
c_5: & -X_1 & + & X_2 & \geq & 6 \\
c_6: & 16X_1 & + & 17X_2 & \geq & 272 \\
c_7: & 2X_1 & - & 7X_2 & \geq & 14
\end{array}
$$

$$
\begin{array}{lrcrcrcr}
c_8: & \lambda_1(X,X',\varepsilon_8) & \wedge & 2X_1 - 5X_2 \leq 16 & \wedge & 2X_1' - 5X_2' \leq 16 \\
c_9: & \lambda_2(X,X',\varepsilon_9) & \wedge & 9X_1 \leq 46 & \wedge & 9X_1' \leq 46 \\
c_{10}: & \lambda_1(X,X',\varepsilon_{10}) & \wedge & 5X_1 + 8X_2 \leq 45 & \wedge & 5X_1' + 8X_2' \leq 45
\end{array}
$$

where:

$(X_1, X_2) \in \hat{\Re}_2 \subset \hat{\Re}_2^+$, $\{c_1, c_2\} = S_f$, $\{c_3, c_4, c_5, c_6, c_7\} = S_c$ and $\{c_8, c_9, c_{10}\} = S_v$.

$\varepsilon_8 > 0$, $\varepsilon_9 > 0$ and $\varepsilon_{10} > 0$ (user defined).

With respect to the general problem given in section 6.3.2, $l' = l = 2$, $m' = m = 7$ and $p' = p = 10$.

(E.1) Apply step (B.1) as outlined in section 6.3.2:

$$
\begin{array}{lrcrcrcr}
c_1: & 2X_1 & & & - & 41 & \leq & 0 \\
c_2: & & & 3X_2 & - & 65 & \leq & 0 \\
c_3: & 4X_1 & + & 3X_2 & - & 12 & \leq & 0 \\
c_4: & 2X_1 & - & X_2 & + & 2 & \leq & 0 \\
c_5: & X_1 & - & X_2 & + & 6 & \leq & 0 \\
c_6: & -16X_1 & - & 17X_2 & + & 272 & \leq & 0 \\
c_7: & -2X_1 & + & 7X_2 & + & 14 & \leq & 0
\end{array}
$$

$$
\begin{array}{lrcrcrcr}
c_8: & \lambda_1(X,X',\varepsilon_8) & \wedge & 2X_1 - 5X_2 - 16 \leq 0 & \wedge & 2X_1' - 5X_2' - 16 \leq 0 \\
c_9: & \lambda_2(X,X',\varepsilon_9) & \wedge & 9X_1 - 46 \leq 0 & \wedge & 9X_1' - 46 \leq 0 \\
c_{10}: & \lambda_1(X,X',\varepsilon_{10}) & \wedge & 5X_1 + 8X_2 - 45 \leq 0 & \wedge & 5X_1' + 8X_2' - 45 \leq 0
\end{array}
$$

(E.2) Apply step (B.2) as outlined in section 6.3.2:

$$
\begin{array}{rcrcrcrcrcl}
2X_1 & & & - & 41 & + & 2\Delta_1 & & & \leq & 0 \\
 & & 3X_2 & - & 65 & & & + & 3\Delta_2 & \leq & 0 \\
4X_1 & + & 3X_2 & - & 12 & + & 4\Delta_1 & + & 3\Delta_2 & \leq & 0 \\
2X_1 & - & X_2 & + & 2 & + & 2\Delta_1 & & & \leq & 0 \\
X_1 & - & X_2 & + & 6 & + & \Delta_1 & & & \leq & 0 \\
-16X_1 & - & 17X_2 & + & 272 & & & & & \leq & 0 \\
-2X_1 & + & 7X_2 & + & 14 & + & & + & 7\Delta_2 & \leq & 0 \\
2X_1 & - & 5X_2 & - & 16 & + & 2\Delta_1 & & & \leq & 0 \\
 & & & & & & \Delta_1 & & & \geq & \varepsilon_8 \\
9X_1 & & & - & 46 & & & & & \leq & 0 \\
 & & & & & & & & \Delta_2 & \geq & \varepsilon_9 \\
5X_1 & + & 8X_2 & - & 45 & + & 5\Delta_1 & & & \leq & 0 \\
 & & & & & & \Delta_1 & & & \geq & \varepsilon_{10}
\end{array}
$$

(E.3) Apply step (B.3) as outlined in section 6.3.2:

$$
\begin{array}{ll}
\min & M \sum_{U} C_i + \sum_{I_p - I_l - U} C_i \\
X \in \hat{\Re}_2^+ & \\
\Delta \in \Re_2^+ & \\
C \in \Gamma &
\end{array}
$$

subject to:

$$
\begin{array}{rcrcrcrcrcrcl}
2X_1 & & & - & 41 & + & 2\Delta_1 & & & & & \leq & 0 \\
 & & 3X_2 & - & 65 & & & + & 3\Delta_2 & & & \leq & 0 \\
4X_1 & + & 3X_2 & - & 12 & + & 4\Delta_1 & + & 3\Delta_2 & - & BC_3 & \leq & 0 \\
2X_1 & - & X_2 & + & 2 & + & 2\Delta_1 & & & - & BC_4 & \leq & 0 \\
X_1 & - & X_2 & + & 6 & + & \Delta_1 & & & - & BC_5 & \leq & 0 \\
-16X_1 & - & 17X_2 & + & 272 & & & & & - & BC_6 & \leq & 0 \\
-2X_1 & + & 7X_2 & + & 14 & + & & + & 7\Delta_2 & - & BC_7 & \leq & 0 \\
2X_1 & - & 5X_2 & - & 16 & + & 2\Delta_1 & & & - & BC_8 & \leq & 0 \\
 & & & & & & \Delta_1 & & & + & BC_8 & \geq & \varepsilon_8 \\
9X_1 & & & - & 46 & & & & & - & BC_9 & \leq & 0 \\
 & & & & & & & & \Delta_2 & + & BC_9 & \geq & \varepsilon_9 \\
5X_1 & + & 8X_2 & - & 45 & + & 5\Delta_1 & & & - & BC_{10} & \leq & 0 \\
 & & & & & & \Delta_1 & & & + & BC_{10} & \geq & \varepsilon_{10}
\end{array}
$$

with

$$B > \max_{X \in \hat{\mathbb{R}}_2^+} \{ \mid 2X_1 - 41 \mid, \mid 3X_2 - 65 \mid, \mid 4X_1 + 3X_2 - 12 \mid, \mid 2X_1 - X_2 + 2 \mid \mid X_1 - X_2 + 6 \mid, \mid -16X_1 - 17X_2 + 272 \mid, \mid -2X_1 + 7X_2 + 14 \mid, \mid 2X_1 - 5X_2 - 16 \mid, \mid 9X_1 - 46 \mid, \mid 5X_1 + 8X_2 - 45 \mid \}$$

$$M > \mid I_p - I_l - U \mid$$

Γ is the set of all possible $(p - l)$ - vectors whose entries are either 0 or 1.

The discussion of Example 15 in Appendix C shows that the C^3 algorithm generates the consistent cover $\tilde{E}$= $\{\{1,2,4,5,6,8,9\},\{1,2,3,8,9,10\},\{1,2,7,8,10\}\}$ for this problem. The optimality of this result is also demonstrated in Appendix C.

6.4 Comparison of Transition Rules for Introducing Slack

Three general points of comparison between Transition Rule A and Transition Rule B are:

i) quality of solution (i.e., number of members in the resulting consistent cover)

ii) complexity (i.e., number of constraints added to handle a set of PC's)

iii) special feature

i) Quality of Solution

The quality of a solution generated by Transition Rule A may be better than that of Transition Rule B (i.e., the number of consistent subsets generated by Rule A is less than or equal to the number of consistent subsets generated by Rule B). This follows from the results of Theorem 6.5 and Theorem 6.7. Theorem 6.7 is a sufficiency condition for handling a set of PC's combined with a non-empty $S_f \bigcup S_c$ (see Example 6.1 below). On the other hand, Theorem 6.5 establishes both sufficient and necessary conditions.

Example 6.1: Consider the following set of constraint satisfaction problem:

$$\left.\begin{array}{lrcl} c_1: & -X_1+3 & \leq & 0 \\ c_2: & -X_2+5 & \leq & 0 \\ c_3: & X_1+X_2-9 & \leq & 0 \\ c_4: & \lambda_1(X,X',\varepsilon_4) \wedge 2X_1+3X_2-30 \leq 0 \wedge 2X_1'+3X_2'-30 & \leq & 0 \\ c_5: & \lambda_2(X,X',\varepsilon_5) \wedge 3X_2-35 \leq 0 \wedge 3X_2'-35 & \leq & 0 \end{array}\right] \quad \cdots(\$)$$

where $\varepsilon_4 = 1$ and $\varepsilon_5 = 1$. It can be easily verified that $\{X^*\} \bigcup \{X^4, X^5\}$ is a feasible bag for (\$) where $X^* = [3,5]$, $X^4 = [4,5]$ and $X^5 = [3,6]$.

Now using Transition Rule B for (\$), we obtain the following constraint satisfaction problem:

$$\left.\begin{array}{rcrcl} -X_1+3 & - & BC_1 & \leq & 0 \\ -X_2+5 & - & BC_2 & \leq & 0 \\ X_1+X_2-9+\Delta_1+\Delta_2 & - & BC_3 & \leq & 0 \\ 2X_1+3X_2+2\Delta_1 & - & BC_4 & \leq & 0 \\ \Delta_1 & + & BC_4 & \geq & \epsilon_4 \\ 3X_2+3\Delta_2 & - & BC_5 & \leq & 0 \\ \Delta_2 & + & BC_5 & \geq & \epsilon_5 \end{array}\right] \quad \cdots(\$\$)$$

where $X \in \hat{\Re}_n$, $\Delta_1 \geq 0$ and $\Delta_2 \geq 0$.

Let the $(n+2+2+5)$ - vector $\begin{bmatrix} X^* \\ \Delta^* \\ w^* \\ C^* \end{bmatrix}$ be a solution to (\$\$). In order that all 5 of these constraints (i.e., c_1, c_2, c_3, c_4, and c_5) coexist in the same consistent subset, each of C_1^*, C_2^*, C_3^*, C_4^* and C_5^* must be zero. However $C_1^* = 0$ implies $X_1^* \geq 3$ and $C_2^* = 0$ implies $X_2^* \geq 5$. These values together with $C_3^* = 0$ imply $\Delta_1^* + \Delta_2^* \leq 1$. But then $C_4^* = C_5^* = 0$ implies $\Delta_1^* \geq 1$ and $\Delta_2^* \geq 1$ which results in an impossible requirement. In other words Transition Rule B can not find a solution subvector $C^* = 0$.

ii) Complexity

Table 6.1 compares the complexity of Transition Rules A and B from the point of view of the additional constraints that are introduced. In this table $\mathcal{N} = 2 \mid S_f \cup S_c \mid \times \mid S_v \mid + 2 \mid S_v \mid$. The table entries give the number of additional constraints (excluding the non-negativity constraints) under a worst-case scenario. This scenario has the following features:

a) all constraints are equality constraints

b) all constraints are potentially sensitive to X_j for each $j \in P_{S_v}$

c) $\mid P_{S_v} \mid = \mid S_v \mid$

Case	Transition Rule A	Transition Rule B
Nonlinear	$\mathcal{N}$	$\mathcal{N}$
Linear	$\frac{\mathcal{N}}{2}$	$\mid S_v \mid$

Table 6.1: Complexity of handling PC's

The factor of 2 that arises in the table entry for Transition Rule A/Linear case is a direct consequence of Theorem 6.2. It should be stressed that the value shown for Transition Rule B/Linear case is independent of feature a) above. Observe that in case of linear constraints, the weakness of Rule B regarding the quality of a solution is offset by a substantial decrease in the number of constraints to be added to handle a set of PC's.

iii) Special Feature

Transition Rule A imposes a fixed perturbation on the input vector for the model. More specifically, observe that the perturbation size in the vectors that appear in the feasible bag in Theorem 6.5 is fixed. On the other hand, Transition Rule B allows for a variable perturbation. This can be seen by noting in Theorem 6.7 that the perturbation size in the vectors that appear in the feasible bag are subject only to a lower bound constraint. However, Transition Rule A can nevertheless be easily modified to handle variable perturbation by substituting a new variable Δ_i for the user defined value ε_i in the specification of the perturbed vectors and then adding an associated constraint $\Delta_i \geq \varepsilon_i$.

Chapter 7

Conclusions and Further Study

The purpose of this research project has been to provide a contribution to the simulation model validation problem. This has been achieved along a variety of dimensions.

The underlying knowledge-based approach to the validation problem adopted in the study is novel. The fundamental advantage of the approach is that it enables the incorporation of domain specific knowledge into the validation problem. Such knowledge has traditionally been a primary source of direction for the simulation model validation activity but no formal basis for its incorporation has existed. The work outlined in this thesis provides such a basis.

From a system perspective, our approach has three phases; namely, experiment design, experiment execution and experiment evaluation. The key element for both the experiment design and experiment evaluation phases is a knowledge base of validation reference information. This knowledge base contains a characterization of the intended and/or expected behaviour of the simulation model. An effective means for describing this knowledge base has been developed in our work. This is based

on a simple abstract system model called a dynamic object with which there is associated the notion of "behaviour". We explore various properties of the behaviour of dynamic objects which then provide the basis for characterizing the behaviour of simulation models, and hence for characterizing the knowledge base of validation reference information.

We regard this knowledge base as an external specification of simulation model behaviour. We develop this specification as the union of three disjoint sets of relationships which we refer to as Formal specifications, Qualitative specifications and Observational specifications. Each of these categories of relationships is defined in terms of our abstract model of a dynamic object.

A significant part of our contribution relates to the experiment design problem. This corresponds to the problem of identifying set of experiments whose execution will efficiently and effectively validate the simulation model with respect to its available external specification. A precise specification for this problem is given in a constraint set framework and a procedure for extracting this constraint set from the external specification is presented. We refer to this problem as the consistent constraint cover (C^3) problem. Some insights into its properties are provided and algorithms for its solution are formulated.

An analysis of the complexity of the C^3 problem is provided. This is based on a graph-theoretical approach using two new concepts; namely, a constraint set graph and a consistent constraint set graph. A relationship between a maximum independent set of a constraint set graph and the solution of a simplified version of the C^3 problem (referred to as the $\hat{C}^3$ problem) is developed. It is established that the $\hat{C}^3$ problem is related to both the edge cover problem and the clique cover problem.

Because of the relationship of the C^3 problem to the clique cover problem which is known to be NP-complete, an approximate (rather than optimal) solution algorithm (referred to as the C^3 algorithm) for the C^3 problem has been designed and its performance evaluated. The change-in-value causal relationships within the set of qualitative specifications presented a particularly challenging aspect of the algorithm design. A method for handling such relationships has been developed using the concept of "slack". An implementation of the C^3 algorithm has been carried out and used on a set of examples.

In as much as the C^3 algorithm provides an effective solution procedure for the experiment design aspect of our knowledge-based approach to the validation problem, we have provided a substantial contribution to the practical realization of this approach.

The research presented in this thesis provides a foundation for developing an effective software environment for the validation of simulation models. Further research and development work will however be necessary to achieve this goal. Some of the topics that need to be considered are listed below:

i) The knowledge base of validation reference information is a crucial component of our proposed simulation model validation system. A means for conveniently and effectively acquiring the necessary knowledge about expected model behaviour from a domain expert needs to be designed and developed. In this regard it may be useful to explore the suitability of a C/E net [20, 41, 43, 89, 95, 118] based formalism together with an implementation that incorporates an effective graphical interface.

ii) Our work in this thesis has not addressed the experiment evaluation component of our system architecture. The design and development of this component should provide many research challenges particularly in the area of data management. In addition there exist conceptual issues relating to the definition of suitable metrics to be used in the evaluation of the experiment results. It would, furthermore, be desirable to develop appropriate means for incorporating classical statistical validation techniques into the evaluation process.

iii) We provide a solution approach for the problem P that must be solved on each iteration of the C^3 algorithm. When all constraints are linear this approach can be handled with existing mixed integer programming tools [110]. When the constraints are not all linear, software tools for accomodating the approach need to be investigated. It would therefore be useful to study this problem further.

iv) A particular point of view was adopted in our study for defining a consistent constraint set when perturbation constraints are present. This perspective made possible an effective approximate solution procedure for the C^3 problem. However, the approach gives rise to some deviation from the originally stated goal of minimizing the number of experiments required to effectively "exercise" the knowledge base. An alternate definition of constraint set consistency in the presence of perturbation constraints could be worthwhile.

v) The complexity of the C^3 problem (with or without the inclusion of perturbation constraints) remains an open problem. Our preliminary study of this issue suggests that it is NP-complete but a thorough analysis would be an interesting research problem.

Appendix A – An External Specification for a Circuit-Switching Simulation Model

For the purpose of illustration, we present below an external specification for Circuit-Switching Simulation Model [17]. Note that the external specification provided in this example for the Circuit-Switching Simulation Model has not been acquired from a telecommunication (i.e., domain) expert as required by our approach. Instead the external specification has been created by the author from a study of textual knowledge [11, 17, 40, 46, 61, 97].

Telephone Networks [11, 40, 46, 97]

One form of communication networks is telephone networks. Telephone networks, like other networks consist of nodes and connections. The nodes in a telephone network are the switching centers (SC) that contain devices (switches) which assist in selecting and establishing a speech path from one user (caller) to another (called). The connections are the trunk (circuit) groups between the SCs. A speech path may pass through several nodes. The purpose of these systems is to provide the means of connecting two subscribers of the system when one of them requests the service. Each subscriber is connected to one of the SCs in the system by a circuit. Since it is not economically feasible to connect each SC to all other SCs, a network is divided into sub areas, sub-sub areas etc., which usually match with the geographical and administrational areas of the country. In the subdivision, the SCs are hierarchically ordered which implies that those on the higher levels are pure transit SCs, while those on lower levels, local SCs or Tandems, have subscribers connected to them.

The switches that establish and maintain a communication path between a pair of telephone sets are provided from a common pool when required by a call and returned to the pool when no longer needed. This brings up the situation that the system may be unable to set up a call on demand because of a lack of available resource (equipment) at that time. Then the solution to this problem of "how much equipment must be provided so that the proportion of calls subject to delays will be below an acceptable level" becomes an important issue. over a specified time interval

The lack of suitable theoretical methods for network planning leaves simulation as the only practical aid to provide insight into the behaviour of such large systems. This has led to several research efforts in developing a general purpose tool for simulation of network traffic. A commercially available Communications Network Simulation package, COMNET [17] is the most advanced. COMNET II.5 is a performance analysis tool for telecommunication networks. With the appropriate description of a network and its routing algorithms, it simulates the operation of the network and provides measures of the network performance. COMNET II.5 is developed in SIMSCRIPT II.5 which is a comprehensive computer simulation language [61].

Statement of a Circuit-Switching Simulation Problem [17]

A specific problem which we examine is shown in Figure A.1. It represents a voice network consisting of three switching nodes: Branch PBX A, Branch PBX B, and a Headquarters Tandem PBX. In the sequel, we refer to PBX A, PBX B and TANDEM PBX simply as A, B and TANDEM respectively.

We assume that the circuit group from A to B is subject to random failures. The time between failures and the repair times are exponentially distributed.

The network has normal priority calls and high priority calls. The high priority

calls can preempt normal priority calls. Normal priority point-to-point calls originate from A to B, from A to the TANDEM, from B to A, from B to the TANDEM, from the TANDEM to A and from the TANDEM to B. High priority point-to-point calls originate from the TANDEM to A and from the TANDEM to B. The interarrival distribution for all call categories is exponential. The average holding time (i.e., duration) for all call categories is normally distributed.

Calls originating at A and destined for B are routed to the TANDEM when all circuits from A to B are busy. Calls originating at B and destined for A are routed to the TANDEM when all circuits from B to A are busy. A and B have no tandem switching capabilities (i.e., calls originating at the TANDEM and destined for A (B) can not be routed through B (A) in the event that all circuits from the TANDEM to A (B) are busy). The routing strategy is source-node routing.

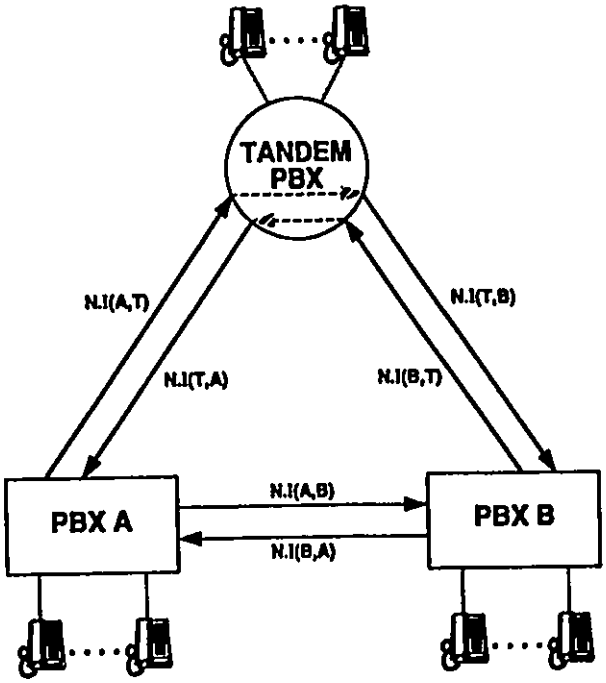

Figure A.1. The Circuit-Switching Network Topology

Domain Knowledge for the Circuit-Switching Simulation Problem

- With source-node routing, every call category has an ordered list of end-to-end paths. When a call occurs during simulation, it is routed over the first path in the list that has all circuit groups operational and sufficient capacity on each circuit group of the path. If no path can be found and preemption is not possible, the call is blocked. Furthermore, with source-node routing call queueing is not allowed.

- Point-to-point circuits connect exactly two nodes. Circuit-switched traffic can only be routed over point-to-point circuit groups. The capacity of a point-to-point circuit group can be defined by the number of circuits in the group. A circuit-switched call that is routed over a cicruit group with capacity defined by the number of circuits occupies one circuit in the group for the duration of the call. Furthermore, the number of circuits applies only to point-to-point circuit groups with capacity defined in units of circuits. All circuits in the same circuit group are identical.

- Each category of traffic has an originating node, destination node, and a class of service identifier. Each class of service has an attribute called priority number. If a network has call preemption capability, a high priority call can preempt a low priority call when no alternate route is available; preemption is tried successively on each path at each circuit group on the path that caused blocking. Modeling call preemption imposes additional overhead on the simulation to keep track of individual calls on circuit groups. Additional overhead is also incurred during the simulation for each circuit group that is subject to failure.

Specification of the Input and Output Vectors for a Circuit-Switching Simulation Model

Input Vector (X)

The input vector, X, for any simulation model for the system would have many components. A partial list of these components is provided below.

Note: (S, D) means "from Source S to Destination D"; where S and D can be either A, B or T (T = TANDEM) subject to $S \neq D$.

$N.l(S,D)$: number of circuits from S to D

$siml$: duration of the simulation

$\mu a(S,D)$: mean interarrival time for normal priority calls from S to D

$\mu e(T,D)$: mean interarrival time for high priority calls from T to D

$\mu f(A,B)$: mean time to failure of the circuit group from A to B

μh : mean holding time for all call categories

$\mu r(A,B)$: mean time to repair the circuit group from A to B

Output Vector (Y)

The output vector, Y, for any simulation model for the system would have many components. A partial list of these components is provided below:

$N.a(S,D)$: number of normal priority calls attempted from S to D; i.e., the number of calls attempted over some circuit group (an attempt need not be successful). The number of calls attempted includes both first-offered and alternate-routed traffic.

$N.bf(S,D)$: number of normal priority calls blocked by circuit group failure from S to D; i.e., the number of call attempts that cannot be completed over the circuit group because the circuit group has failed. It is still

possible that the call could find another route.

$N.bt(S,D)$: number of normal priority calls blocked by circuit group traffic from S to D; i.e., the number of call attempts that cannot be completed over the circuit group because insufficient capacity is available as a result of other traffic on the circuit group. It is still possible that the call could find another route.

$N.bte(T,D)$: number of high priority calls blocked by circuit group traffic from T to D

$N.c(S,D)$: number of normal priority calls carried from S to D; i.e., the number of calls that are successfully routed over a circuit group.

$N.c1(S,D)$: number of normal priority calls carried either from A to B or from B to A directly (without TANDEM)

$N.c2(S,D)$: number of normal priority calls carried either from A to B or from B to A via TANDEM

$N.ce(T,D)$: number of high priority calls carried from T to D

$N.d(S,D)$: number of normal priority calls disconnected from S to D; i.e., the number of calls disconnected (and cleared) due to a failure on the circuit group.

$N.e(T,D)$: number of high priority calls attempted from T to D

$N.gf(S,D)$: circuit group failures; i.e., the number of times that a circuit group from S to D fails.

$N.p(T,D)$: number of normal priority calls preempted from T to D; i.e., the number of carried calls preempted by higher priority calls. A call that acquires circuit group capacity through preemption does not

count as a blocked call.

$P.a(S,D)$: percent availability; i.e., the percentage of time that the circuit group from S to D is operational.

$P.u(S,D)$: utilization percentage of a circuit group from S to D; i.e., the average number of busy circuits in a group divided by the total number of circuits in the group, multiplied by 100.

$lo(S,D)$: carried load (erlangs) of a circuit group from S to D; i.e., the average number of circuits that are busy due to circuit-switched calls.

$pb(S,D)$: blocking probability of a circuit group from S to D; i.e., the fraction of call attempts that are blocked due to a combination of circuit group failures and circuit group traffic.

$repl(S,D)$: total repair time of the circuit group from S to D

Notes:

1. Suppose that input variable $N.l(A,B)$ has been assigned the j^{th} location in the input vector, X of the simulation model. In terms of our notation of section 2.1.1 one of the "change in value" predicates associated with $N.l(A,B)$ would be $\delta_j^*(X^k, X^l, \epsilon)$. To proceed with this notation it would then be necessary here to identify an explicit location in X for each of our input variables. We circumvent this with a simplified notation where $N.l(A,B)\uparrow_k^l$ is used to represent $\delta_j^*(X^k, X^l, \epsilon)$. This simplified notation also suppresses the value ϵ which identifies the size of the change. An appropriate value would have to be given by the user. Similarly, we use $P.a(A,B)\uparrow_k^l$ to represent $\delta_i(Y^k, Y^l, \epsilon)$ on the assumption that $P.a(A,B)$ has been assigned the i^{th} location of the output vector, Y.

2. There is no assurance that the input (X) and output (Y) vectors specified above are sufficiently comprehensive for a practical simulation study. Their components have been selected to ensure that a sufficiently illustrative external specification can be provided.

3. It is also interesting to observe here how our external specification can be developed in the absence of the simulation model.

An External Specification for a Circuit-Switching Simulation Model

Given below is an external specification from the example problem under consideration. It is intended simply to be illustrative and certainly not exhaustive.

Formal Specifications

Input features

All explicitly listed components of X will lie between a lower bound and an upper bound.

Behavioural features

All explicitly listed components of Y must be non-negative.

1. Carried load (in erlangs) is less than or equal to the number of circuits:

$$lo(S, D) \leq N.l(S, D)$$

2. Only circuit group from A to B is subject to random failures:

$$N.gf(S, D) = 0 \ \wedge \ repl(S, D) = 0 \ \wedge \ P.a(S, D) = 100 \quad \text{for } (S, D) \neq (A, B)$$

3. For normal priority calls, number of calls attempted is the sum of the number of calls blocked by circuit group failure and traffic, and the number of calls carried:

$$N.a(S, D) = N.bf(S, D) + N.bt(S, D) + N.c(S, D)$$

4. For normal priority calls, number of calls blocked by circuit group failure is the sum of the number of calls preempted and disconnected:

$$N.bf(S, D) = N.p(T, D) + N.d(S, D)$$

5. For normal priority calls, number of calls carried is the sum of the number of calls carried with or without the TANDEM:

$$N.c(S,D) = N.c2(S,D) + N.c1(S,D)$$

6. For high priority calls, number of calls attempted is the sum of the number of calls carried and blocked by circuit group traffic:

$$N.e(T,D) = N.ce(T,D) + N.bte(T,D)$$

7. Blocking propbability is the number of calls blocked divided by the number of calls attempted:

$$pb(S,D) = [(N.bf(S,D) + N.bt(S,D) + N.bte(T,D))/(N.a(S,D) + N.e(T,D))]$$

8. Utilization percentage of circuit groups:

$$P.u(S,D) = [lo(S,D)/N.l(S,D)] * 100 \ \wedge \ P.u(S,D) \leq 100$$

9. Percent availability of circuit groups:

$$P.a(S,D) = [(siml - repl(S,D))/siml] * 100 \ \wedge \ P.a(A,B) \leq 100$$

Qualitative Specifications

OCR

Behavioural features

1. If the mean time to failure is very large then there will be very few circuit group failures and very few normal priority calls will be disconnected:

$$\textbf{if} \quad (1/\mu f(A,B)) \cong 0 \quad \textbf{then} \quad N.gf(A,B) \cong 0 \ \wedge \ N.d(A,B) \cong 0$$

2. If the mean repair time to failure is very small then total repair time will be very small and percent availability will approach 100:

$$\textbf{if} \quad \mu r(A,B) \cong 0 \qquad \textbf{then} \quad repl(A,B) \cong 0 \ \wedge \ P.a(A,B) \cong 100$$

3. If the mean interarrival time for high priority calls is very large then there will be very few high priority calls attempted and very few normal priority calls preempted:

$$\textbf{if} \quad (1/\mu e(T,D)) \cong 0 \quad \textbf{then} \quad N.e(T,D) \cong 0 \ \wedge \ N.p(T,D) \cong 0$$

4. If very few high priority calls are attempted then there will be very few normal priority calls preempted:

$$\textbf{if} \quad N.e(T,D) \cong 0 \qquad \textbf{then} \quad N.p(T,D) \cong 0$$

CVCR

1. If the mean interarrival time for normal priority calls increases then the number of normal priority calls attempted, carried, disconnected and preempted will decrease:

$$\textbf{if} \quad \mu a(S,D)\uparrow_k^l \qquad \textbf{then} \quad N.a(S,D)\uparrow_l^k \wedge N.c(S,D)\uparrow_l^k \wedge N.d(S,D)\uparrow_l^k \wedge N.p(T,D)\uparrow_l^k$$

2. If the mean interarrival time for high priority calls increases then the number of high priority calls attempted, carried and blocked by traffic will decrease and also the number of normal priority calls preempted will decrease:

$$\textbf{if} \quad \mu e(T,D)\uparrow_k^l \qquad \textbf{then} \quad N.e(T,D)\uparrow_l^k \wedge N.ce(T,D)\uparrow_l^k \wedge N.bte(T,D)\uparrow_l^k \wedge N.p(T,D)\uparrow_l^k$$

3. If the number of normal priority calls attempted decreases then the number of normal priority calls carried, preempted and blocked by failure and traffic will decrease:

$$\textbf{if} \quad N.a(S,D)\uparrow_l^k \qquad \textbf{then} \quad N.c(S,D)\uparrow_l^k \wedge N.p(T,D)\uparrow_l^k \quad N.bf(S,D)\uparrow_l^k \wedge N.bt(S,D)\uparrow_l^k$$

4. If the number of high priority calls attempted decreases then the number of high priority calls carried and blocked by traffic will decrease and also the number of normal priority calls preempted will decrease:

$$\textbf{if} \quad N.e(T,D)\uparrow_l^k \qquad \textbf{then} \quad N.ce(T,D)\uparrow_l^k \wedge N.bte(T,D)\uparrow_l^k \wedge N.p(T,D)\uparrow_l^k$$

Appendix B – Generalized Consistency Definitions

Let S^d and S_e^d be sets of constraints as follows:

$$S^d = \{ \bigwedge_{w=1}^{d_i} [h_i(X)\mathbf{R}_{iw}0], i = 1, 2, \cdots, m\}$$

$$S_e^d = \{\lambda_{a_i}(X, X', \varepsilon_i) \bigwedge_{w=1}^{d_i} [h_{iw}(X)\mathbf{R}_{iw}0 \wedge h_{iw}(X')\mathbf{R}_{iw}0], i = m+1, \cdots, p; a_i \in I_n\}$$

with $X, X' \in \hat{\Re}_n \subset \Re_n$, $\mathbf{R}_{iw} \in \{=, \leq, \geq\}$ and $h_{iw} : \hat{\Re}_n \longrightarrow \Re$. We assume that, for all i and for all w, $h_{iw}(X)$ is bounded and differentiable for all $X \in \hat{\Re}_n$. The predicate $\lambda_{a_i}(X^k, X^l, \varepsilon_i)$ has a TRUE value at the pair (X^k, X^l) if and only if $X_j^l = X_j^k$ for every $j \neq a_i$ and $X_{a_i}^l \geq X_{a_i}^k + \varepsilon_i$ where $\varepsilon_i > 0$.

We provide below extensions of the definitions given earlier in section 3.2.

a) The constraint $\bigwedge_{w=1}^{d_i}[h_{iw}(X)\mathbf{R}_{iw}0] \in S^d$ is satisfied at $X^* \in \hat{\Re}_n$ if $\bigwedge_{w=1}^{d_i}[[h_{iw}(X)]_{X=X^*}\mathbf{R}_{iw}0]$.

b) Let $\hat{S}^d$ be a non-empty subset of S^d and let $\Phi^d = \{X \in \hat{\Re}_n$: every constraint in $\hat{S}^d$ is satisfied at $X\}$. $\hat{S}^d$ is said to be consistent with respect to $\hat{\Re}_n$ if $\Phi^d \neq \emptyset$; otherwise it is inconsistent with respect to $\hat{\Re}_n$. If $\Phi^d \neq \emptyset$ then $\hat{S}^d$ is said to be consistent at each $X \in \Phi^d$.

c) The constraint $\lambda_{a_i}(X, X', \varepsilon_i) \bigwedge_{w=1}^{d_i}[h_{iw}(X)\mathbf{R}_{iw}0 \wedge h_{iw}(X)\mathbf{R}_{iw}0] \in S_e^d$ is satisfied at the pair $(X^k, X^l) \in \hat{\Re}_n \times \hat{\Re}_n$ if and only if

   i) $\lambda_{a_i}(X^k, X^l, \varepsilon_i)$,

   ii) $\bigwedge_{w=1}^{d_i}[[h_{iw}(X)]_{X=X^k}\mathbf{R}_{iw}0]$ and

   iii) $\bigwedge_{w=1}^{d_i}[[h_{iw}(X)]_{X=X^l}\mathbf{R}_{iw}0]$.

d) Let $\hat{S}_e^d \subseteq S_e^d$ and $\hat{S}^d \subseteq S^d$ with $\hat{S}^d \bigcup \hat{S}_e^d \neq \emptyset$. A feasible set for $\hat{S}^d \bigcup \hat{S}_e^d$ is a bag of n - vectors $\{X^*\} \bigcup \{X^i : i = 1, 2, \cdots, \mid \hat{S}_e^d \mid\}$ with $X^* \in \hat{\Re}_n$ and $X^i \in \hat{\Re}_n$ for each i such that

i) $\hat{S}^d$ is consistent at X^* and at each X^i, and

ii) for every $c_i \in \hat{S}_e^d$ c_i is satisfied at the pair (X^*, X^i).

A set of constraints $\hat{S}^d \bigcup \hat{S}_e^d$ is consistent with respect to $\hat{\Re}_n$ if it has a non-empty feasible bag; otherwise it is inconsistent with respect to $\hat{\Re}_n$. Furthermore, $\hat{S}^d \bigcup \hat{S}_e^d$ is said to be consistent at any one of its feasible bags.

Appendix C – Computational Results With the C^3 Algorithm

We have undertaken an implementation of the C^3 algorithm in order to gauge its performance. In this appendix, we present the results of a set of examples treated with this implementation.

Recall that a key element in the C^3 algorithm is the need to solve a mixed integer programming problem on each iteration (Problem P, see section 5.1). Because solution tools for this problem are not readily available for the general case, we restrict our considerations to the case where the associated constraints are linear. As discussed in section 6.4, PMIP/B approach has size advantages when the problem is linear and for this reason it is used in our implementation.

The software tool chosen to handle the mixed integer programming problem is LINDO[107, 110]. This choice is based on the adequacy of the tool and its availability for a personal computer environment.

We first describe the format of the results by giving a detailed explanation for the example EXP1. All other examples follow the same format. Table C.1 summarizes the performance of the C^3 algorithm for this set of examples. The results are given at the end of this appendix.

Example EXP1: Consider the following set of constraints:

$$
\begin{array}{lrcrcrl}
c_1: & X_1 & & & \geq & 5 & \\
c_2: & X_1 & & & \leq & 25 & \\
c_3: & & & X_2 & \geq & 5 & \\
c_4: & X_1 & + & 4X_2 & \leq & 85 & \\
c_5: & X_1 & & & \geq & 15 & \\
c_6: & & & X_2 & \geq & 10 & \wedge \\
c_7: & X_1 & & & \geq & 20 & \\
c_8: & 3X_1 & + & 5X_2 & \geq & 105 & \wedge \\
c_9: & X_1 & & & \leq & 15 &
\end{array}
$$

$$
\begin{array}{lllllll}
c_{10}: & \lambda_2(X, X', \varepsilon_{10}) & \wedge & X_1 \leq 10 & \wedge & X_1' \leq 10 \\
c_{11}: & \lambda_2(X, X', \varepsilon_{11}) & \wedge & -X_1 + X_2 \geq 1 & \wedge & -X_1' + X_2' \geq 1 \\
c_{12}: & \lambda_1(X, X', \varepsilon_{12}) & \wedge & X_1 \geq 18 & \wedge & X_1' \geq 18
\end{array}
$$

$$c_{13}: \quad X_1 + X_2 \leq 5$$

where:

$(X_1, X_2) \in \hat{\Re}_2 \subset \hat{\Re}_2^+$, $\{c_1, c_2, c_3, c_4\} = S_f$, $\{c_5, (c_6, c_7), (c_8, c_9), c_{13}\} = S_c^d$, $\{c_{10}, c_{11}, c_{12}\}$ $= S_v$ and $[\varepsilon_{10}, \varepsilon_{11}, \varepsilon_{12}] = [4, 4, 2]$.

The feasible regions represented by constraints in the sets S_f, S_c^d and S_v are shown in Figure C.1. It also shows an inconsistent region; i.e., the region formed by the constraint c_{13} which is not consistent with the set S_f. Furthermore, it shows two consistent subsets, S1 and S2, generated by the C^3 algorithm. Since $S_v \neq \emptyset$, two feasible bags, A1 $\bigcup$ A2 and B1 $\bigcup$ B2 where A1 = [5, 14], A2 = [5, 18], B1 = [20, 15] and B2 = [25, 15] are also shown. (Note that the arrows from A1 to A2 and from B1 to B2 indicate the direction of the increment in the primary variables X_2 and X_1 respectively.)

An equivalent presentation of the constraints in EXP1 is given in Tableau EXP1.LP. The format of this tableau corresponds to the LINDO program that follows in Tableau EXP1.PMIP/B. Tableau EXP1.PMIP/B contains the PMIP/B formulation for the

constraints in example EXP1. For most of the examples the constraints are presented using only this LINDO format. We describe below the annotations used in Tableau EXP1.LP:

a) The constraints $c_1, c_2, \cdots, c_{13}$ are numbered simply as 1, 2, $\cdots$, 13 respectively.

b) Every element of S_f is annotated by !F. Note that in LINDO environment the comments are preceeded by !. Every element of S_v is annotated by !& CV X_j. The use of & is intended to indicate that every element of S_v is a conjunction of two components; namely, a primary component and a secondary component. The primary variable associated with the primary component follows the notation CV; e.g., CV X1 implies that X1 is the primary variable. No annotations are used for elements of S_c.

c) For every element of S_c^d which is a conjunction of two or more constraints, each individual constraint (except the last one) is annotated by !&. (Note that the same would apply to elements of the set S_v^d except that the last individual constraint is annotated by !& CV X_j.) In Tableau EXP1.LP, the constraints c_6 and c_8 are annotated by !& to indicate that c_6 and c_7 form a compound constraint and c_8 and c_9 form another compound constraint.

Note that the perturbation variables Δ_j are denoted by Ej, and the values used for M and B are 100 and 100000 respectively in Tableau EXP1.PMIP/B.

The solution to EXP1 evolves over a series of iterations as shown in Tableaus: EXP1.S1, EXP1.S2 and EXP1.S3. Each iteration produces the specifications for one member of the consistent cover. The Tableau for each iteration shows the value of the

constraint satisfaction indicator C and the associated feasible value of the variable X.

For Tableau EXP1.S1, the consistent subset is $\{c_1, c_2, c_3, c_4, (c_8, c_9), c_{10}, c_{11}\}$. The associated feasible value for X is $[5, 14]$ and the perturbation value, E, is $[0, 4]$. Hence, the feasible bag for Tableau EXP1.S1 is $\{[5, 14], [5, 18]\}$.

For Tableau EXP1.S2, the consistent subset is $\{c_1, c_2, c_3, c_4, c_5, (c_6, c_7), c_{12}\}$. The associated feasible value for X is $[20, 15]$ and the perturbation value, E, is $[5, 0]$. Hence, the feasible bag for Tableau EXP1.S2 is $\{[20, 15], [25, 15]\}$. The only remaining unsatisfied constraint is c_{13} (i.e., $X_1 + X_2 \leq 5$).

For Tableau EXP1.S3, the consistent subset is $\{c_1, c_2, c_3, c_4, (c_8, c_9), c_{11}\}$. Notice that it does not contain constraint c_{13}. This is because $\{X_1 + X_2 \leq 15\} \bigcup S_f$ is inconsistent. Though we assume the property Υ, inconsistent cases can be handled by terminating the C^3 algorithm when there is no improvement in the solution (i.e., no change in the status of any unsatisfied constraints).

Ignoring the constraint c_{13}, observe that the constraints c_{10} and c_{12} are inconsistent. Therefore, at least two consistent subsets are required. Furthermore, $\{c_{10}, c_{12}\} \in S_v$ which implies that the minimum number of experiments ≥ 4. Notice that the C^3 algorithm generates 4 experiments; i.e., 2 feasible points associated with the 2 feasible bags. These feasible points are $\{[5, 14], [5, 18], [20, 15], [25, 15]\}$.

The above discussion for example EXP1 can be easily carried over to the rest of the examples. However, we only discuss the reasoning associated with the computation of the minimum number of experiments in Table C.1 for the remaining examples.

In EXP2, constraint 10 ($X_6 - X_5 \geq 5$) and constraint 26 ($X_6 - X_5 = 0$) are inconsistent. Therefore, the minimum number of experiments is at least 2.

In EXP3, constraint 10 ($X_6 - X_5 \geq 5$), constraint 25 ($X_5 - X_6 \geq 10$) and constraint 26 ($X_6 - X_5 = 0$) are pairwise inconsistent. Therefore, the minimum number of experiments is at least 3.

In EXP4, constraint 9 ($X_5 \leq 25$) and constraint 15 ($X_5 \geq 50$) are inconsistent. Therefore, the minimum number of experiments is at least 2.

In EXP5[13], constraint ROW4 ($-X_2 + X_3 \geq 50000$) and constraint ROW11 ($X_2 \geq 80000$) imply that $X_3 \geq 130000$. But from constraint ROW6, $X_3 \leq 50000$. Hence, the minimum number of experiments is at least 2.

In EXP6[12], constraint ROW5 ($3X_4 - X_5 \leq 2$) and constraint ROW6 ($X_4 \geq 5$) imply that $X_5 \geq 13$. But from constraint ROW4, $X_5 \leq 2$. Hence, the minimum number of experiments is at least 2.

In EXP7[12], $T_{58} \geq 30$, $T_{25} \leq 10$, $T_{35} \leq 10$, $T_{57} \leq 20$ and $T_{58} \leq 30$ imply that $T_{25} + T_{35} \leq 20$ and $T_{57} + T_{58} \geq 30$. But from constraint NODE5, $T_{25} + T_{35} = T_{57} + T_{58}$. Hence, the minimum number of experiments is at least 2.

EXP8 and EXP9 are discussed in chapter 5 as Example 5.1 and Example 5.2 respectively.

In EXP10, observe that constraint 48 ($\sum_{i=1}^{25} X_i \leq 1250$), constraint 49 ($\sum_{i=1}^{25} X_i \geq 2750$) and compound constraint (50, 51) ($\sum_{i=1}^{25} X_i \geq 1750 \wedge \sum_{i=1}^{25} X_i \leq 2250$) are pairwise inconsistent. Also note that $| P_{S_v} | = 2$. Hence, the minimum number of experiments is at least 3.

In EXP11 (see Figure C.2), perturbation constraint 7 is inconsistent with perturbation constraint 5 and perturbation constraint 7 is also inconsistent with pertrubation constraint 6. Therefore, perturbation constraint 7 references a pair of experiments. Furthermore, perturbation constraints 5 and 6 reference at least three

experiments. Therefore, the minimum number of experiments is at least 5.

EXP12 (see Figure C.3) is the result of augmenting EXP11 with the inclusion of a set S_c having two constraints. Observe that the set S_c does not increase the number of experiments computed by the C^3 algorithm.

In EXP13 (see Figure C.4), perturbation constraint 6 and constraint 8 ($X_2 = 18$) are inconsistent. Furthermore, perturbation constraints 5 and 7 reference a pair of experiments. Therefore, the minimum number of experiments at least 4.

In EXP14 (see Figure C.5), constraint 7 ($X_1 + X_2 \geq 30$) and compound perturbation constraint (5,6) are inconsistent. Therefore, the minimum number of experiments is at least 3.

EXP15 is the result of augmenting EXP9 with the inclusion of sets S_f and S_v consisting of two and three constraints respectively. Observe that the set $S_f \bigcup S_v$ does not increase the number of experiments generated by the C^3 algorithm with PMIP/A and PMIP/B.

#	EXAMPLE	Number of Constraints	Number of Variables	$\lvert S_f \rvert$	$\lvert S_c \rvert$	$\lvert S_v \rvert$	Increment Vector ε	$\lvert P_{S_v} \rvert$	Minimum Number of Experiments	Number of Experiments Generated by the C^3 Algorithm	Figure
1	EXP 1	13	2	4	4 (2 &'s)	3	[4, 4, 2]	2	≥ 4	4	C.1
2	EXP 2	30	10	1	29	–	N/A	–	≥ 2	3	–
3	EXP 3	30	10	5	25	–	N/A	–	≥ 3	3	–
4	EXP 4	20	10	5	15	–	N/A	–	≥ 2	3	–
5	EXP 5*	11	8	–	11	–	N/A	–	≥ 2	2	–
6	EXP 6*	9	4	–	9	–	N/A	–	≥ 2	2	–
7	EXP 7*	16	8	–	16	–	N/A	–	≥ 2	2	–
8	EXP 8	5	2	–	5	–	N/A	–	≥ 3	4	5.2
9	EXP 9	5	2	–	5	–	N/A	–	≥ 3	3	5.4
10	EXP 10	96	25	37	18 (37 &'s)	4	[25,25,50,50]	2	≥ 3	3	–
11	EXP 11	7	2	4	–	3	[2, 2, 2]	2	≥ 5	5	C.2
12	EXP 12	9	2	4	2	3	[2, 2, 2]	2	≥ 5	5	C.3
13	EXP 13	8	2	4	1	3	[2, 2, 2]	2	≥ 4	4	C.4
14	EXP 14	8	2	4	2	1 (1 &'s)	[2]	1	≥ 3	3	C.5
15	EXP 15	10	2	2	5	3	[1, 1, 1]	2	≥ 3	3	–

* EXP 5, EXP 6, and EXP 7 are from [11,12]

N/A stands for Not Applicable

n &'s refers to the presence of the n conjunction operators

Table C.1. Summary of the Performance of the C^3 Algorithm

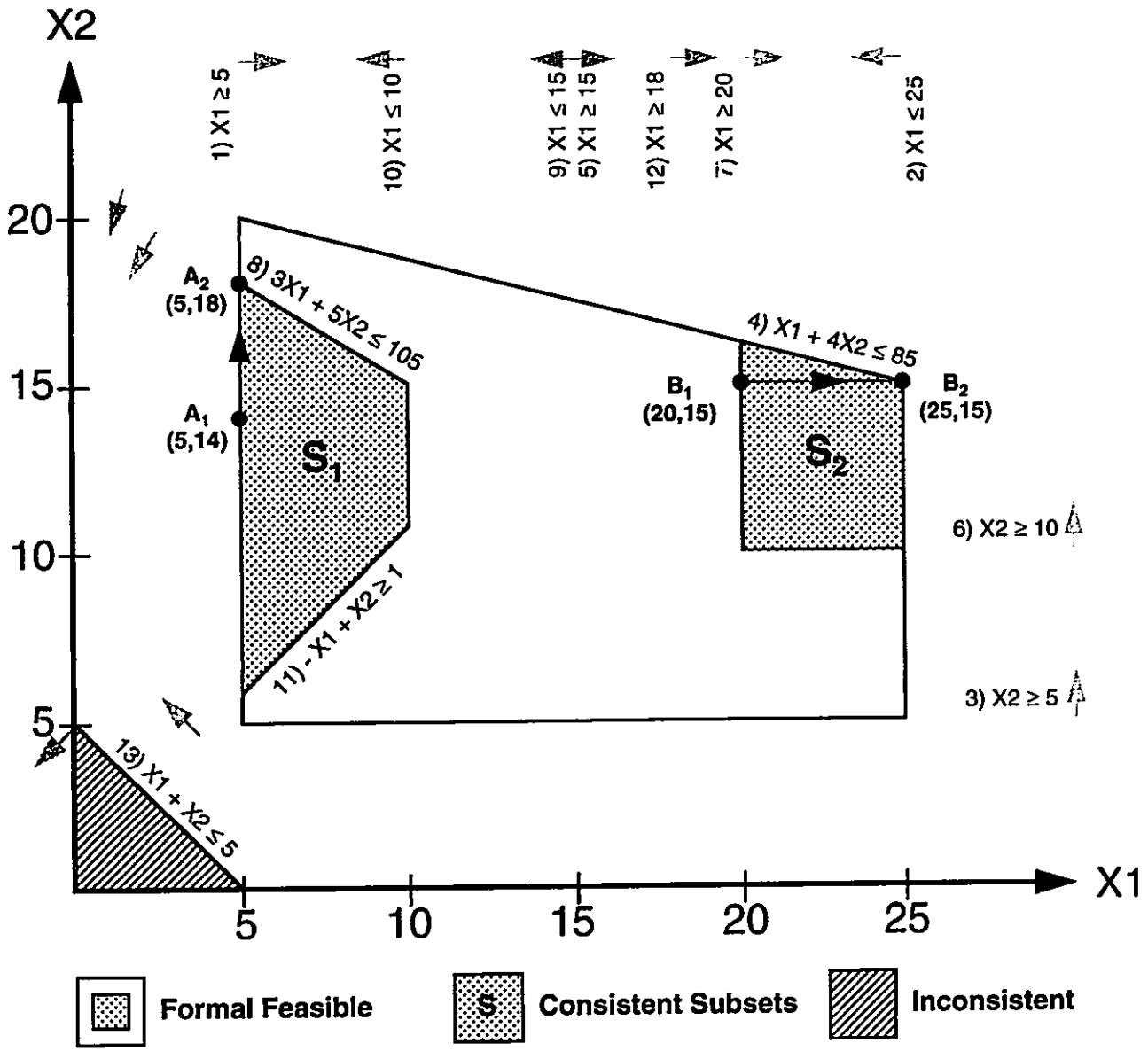

Figure C.1. Constraint Graph for Example 1

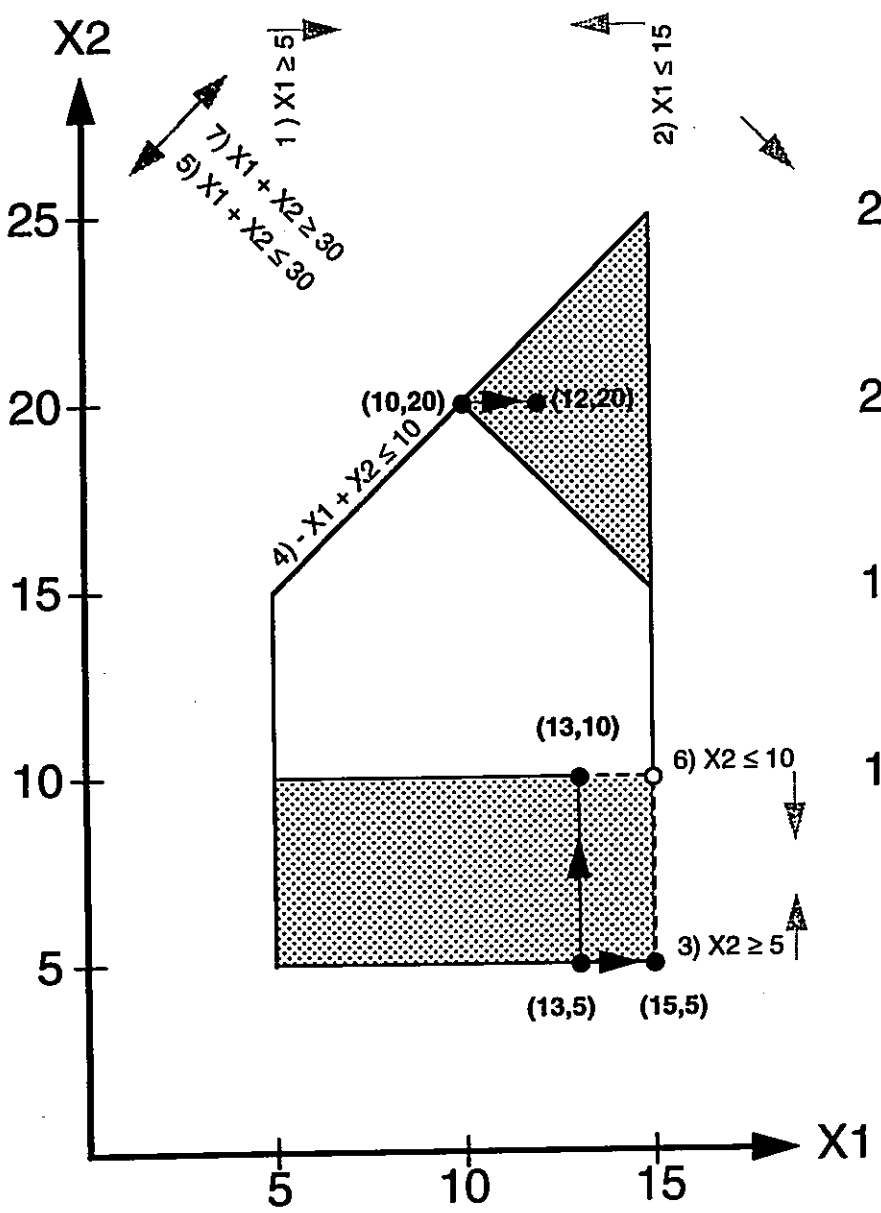

Figure C.2. Constraint Graph for Example 11

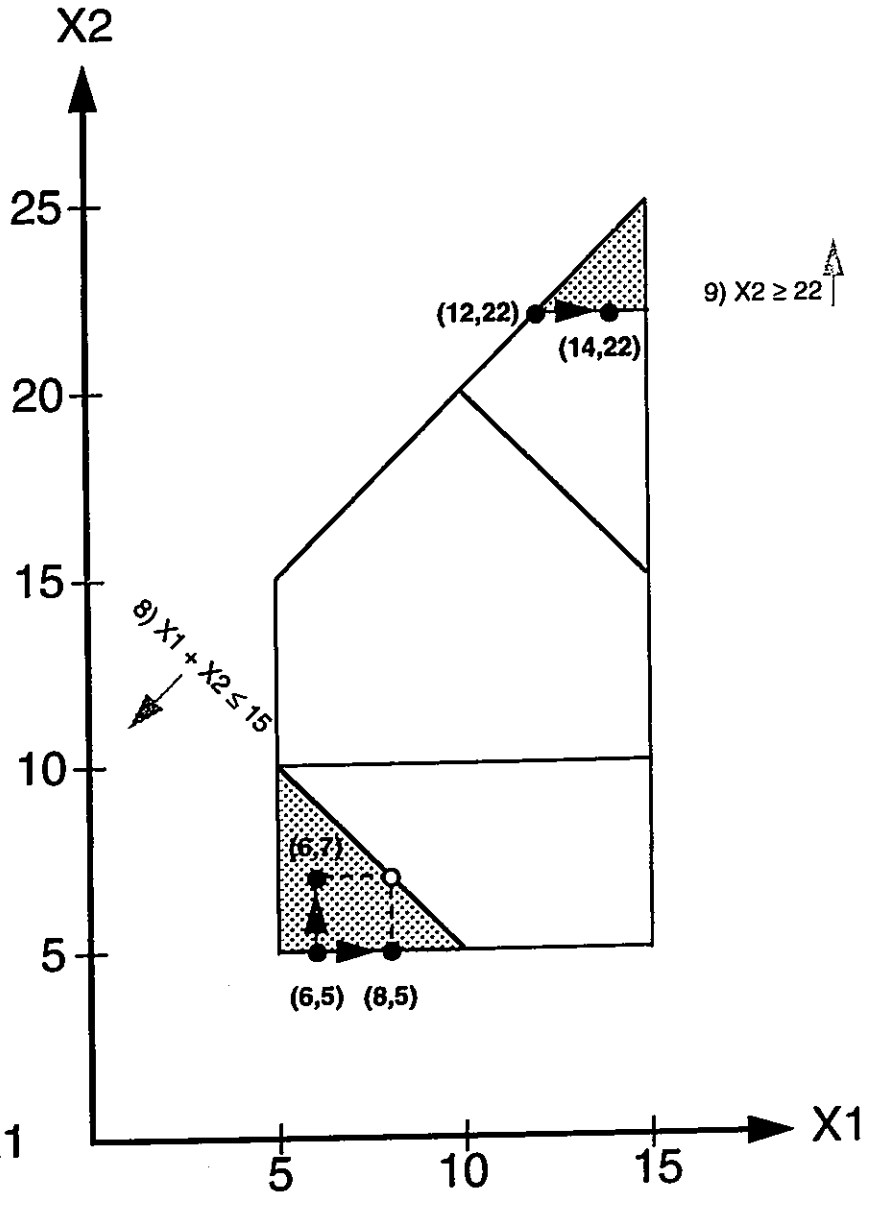

Figure C.3. Constraint Graph for Example 12

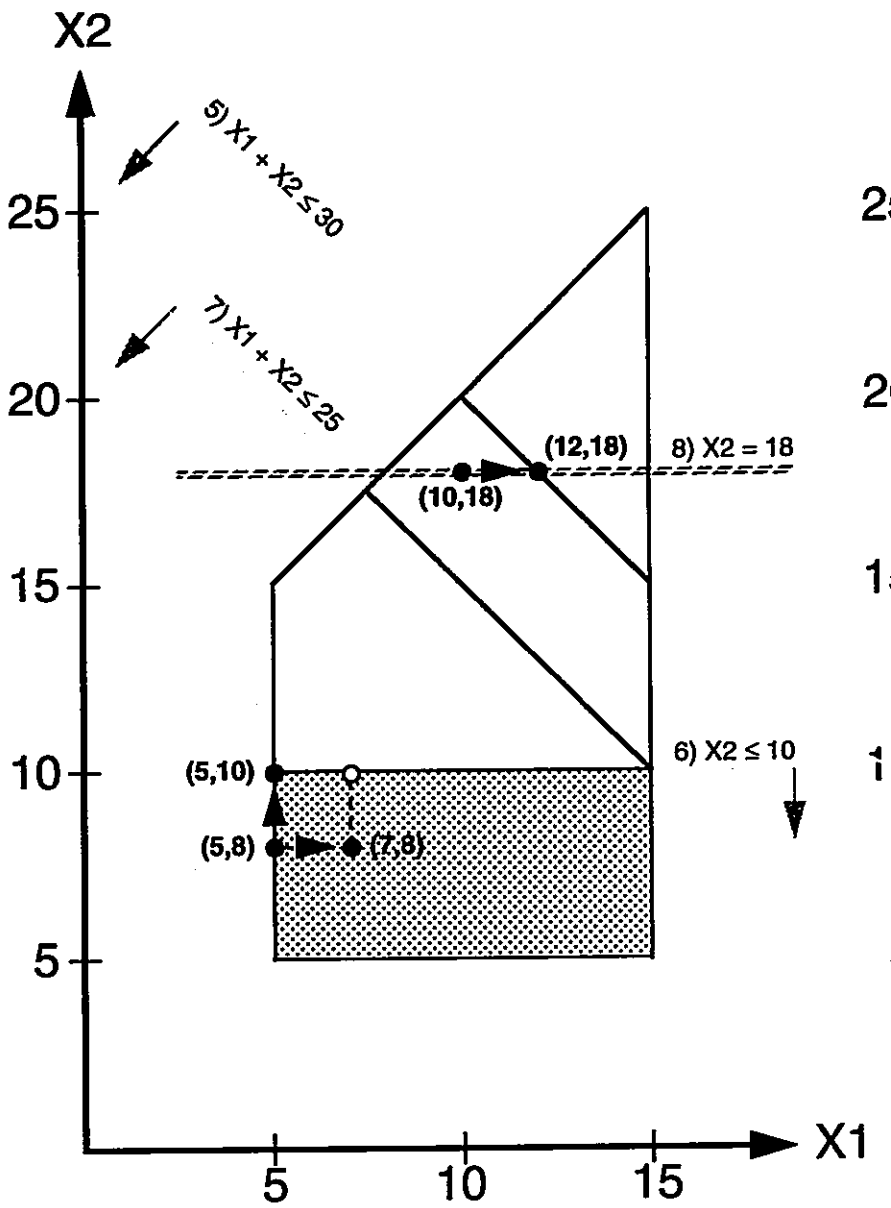

Figure C.4. Constraint Graph for Example 13

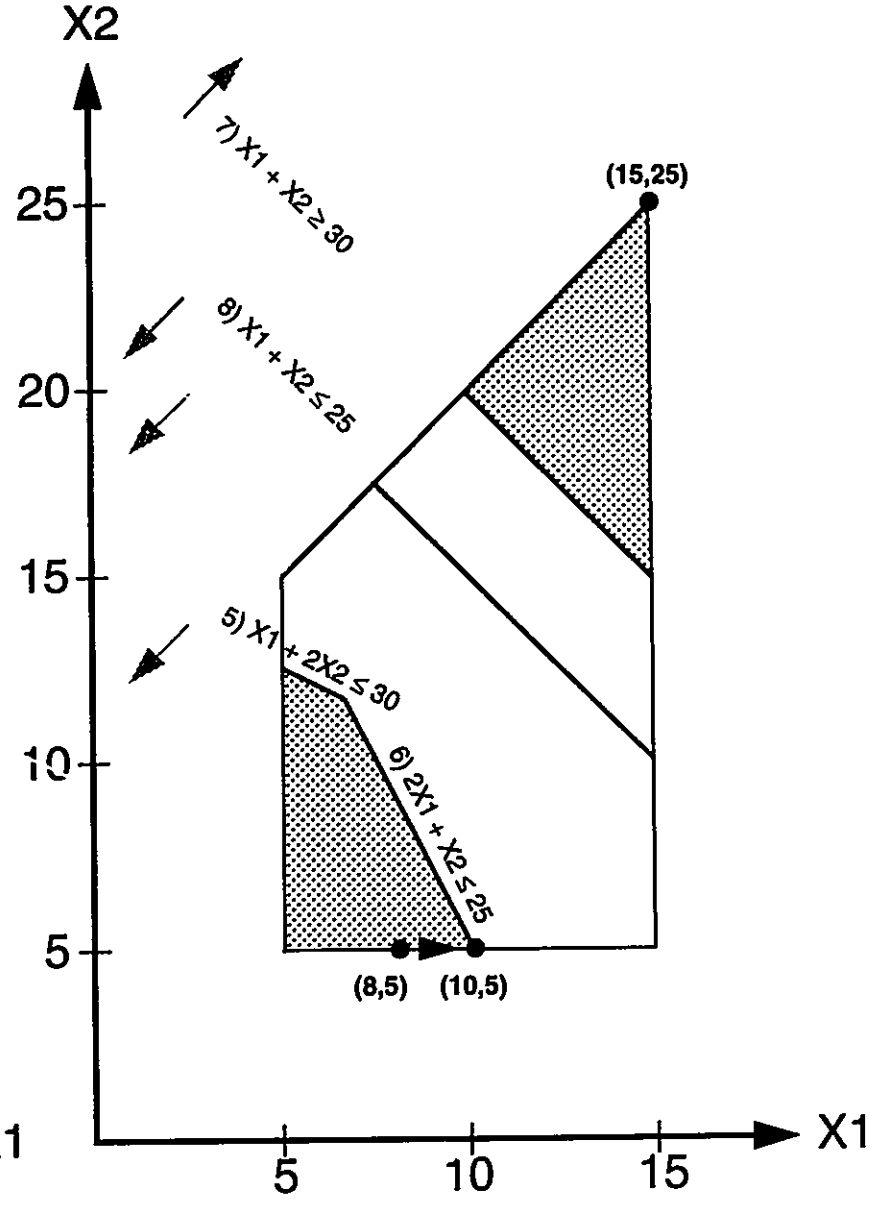

Figure C.5. Constraint Graph for Example 14

EXP1.LP

```
 1)       X1                 >=     5    ! F
 2)       X1                 <=    25    ! F
 3)                   X2     >=     5    ! F
 4)       X1   + 4 X2        <=    85    ! F
 5)       X1                 >=    15
 6)                   X2     >=    10    ! &
 7)       X1                 >=    20
 8)     3 X1   + 5 X2        <=   105    ! &
 9)       X1                 <=    15
10)       X1                 <=    10    ! & CV X2
11)     - X1   +    X2        >=     1    ! & CV X2
12)       X1                 >=    18    ! & CV X1
13)       X1   +    X2        <=     5
```

EXP1.PMIP/B

```
MIN  100 C5  + 100 C6  + 100 C8  + 100 C10 +
     100 C11 + 100 C12 + 100 C13
ST
     X1                                                    >=     5    ! F
     X1             +    E1                                <=    25    ! F
            X2                                            >=     5    ! F
     X1 + 4 X2      +    E1 + 4 E2                         <=    85    ! F

     X1                                 + 10000 C5         >=    15

            X2                          + 10000 C6         >=    10    ! &
     X1                                 + 10000 C6         >=    20

   3 X1 + 5 X2      + 3 E1 + 5 E2      - 10000 C8         <=   105    ! &
     X1             +    E1            - 10000 C8         <=    15

     X1                                 - 10000 C10        <=    10    ! & CV X2
                                  E2    + 10000 C10        >=     4

   - X1 +    X2                        + 10000 C11        >=     1    ! & CV X2
                                  E2    + 10000 C11        >=     4

     X1                                 + 10000 C12        >=    18    ! & CV X1
                                  E1    + 10000 C12        >=     2

     X1 +    X2      +    E1 +    E2    - 10000 C13        <=     5
END
INTEGER 7
```

EXP1.S1

```
MIN  100 C5  + 100 C6  + 100 C8  + 100 C10 +
     100 C11 + 100 C12 + 100 C13

        OBJECTIVE FUNCTION VALUE

        1)      400.00000

  VARIABLE         VALUE          REDUCED COST
        C5          1.000000        100.000000
        C6          1.000000        100.000000
        C8           .000000        100.000000
       C10           .000000        100.000000
       C11           .000000        100.000000
       C12          1.000000        100.000000
       C13          1.000000        100.000000

        X1          5.000000           .000000
        E1           .000000           .000000
        X2         14.000000           .000000
        E2          4.000000           .000000
```

EXP1.S2

```
MIN  100 C5  + 100 C6  +   1 C8  +   1 C10 +
       1 C11 + 100 C12 + 100 C13

        OBJECTIVE FUNCTION VALUE

        1)      103.00000

  VARIABLE         VALUE          REDUCED COST
        C5           .000000        100.000000
        C6           .000000        100.000000
        C8          1.000000          1.000000
       C10          1.000000          1.000000
       C11          1.000000          1.000000
       C12           .000000        100.000000
       C13          1.000000        100.000000

        X1         20.000000           .000000
        E1          5.000000           .000000
        X2         15.000000           .000000
        E2           .000000           .000000
```

EXP1.S3

```
MIN     1 C5  +   1 C6  +   1 C8  +   1 C10 +
        1 C11 +   1 C12 + 100 C13

         OBJECTIVE FUNCTION VALUE

         1)      103.00000

  VARIABLE          VALUE          REDUCED COST
        C5          1.000000          1.000000
        C6          1.000000          1.000000
        C8           .000000          1.000000
       C10           .000000          1.000000
       C11           .000000          1.000000
       C12          1.000000          1.000000
       C13          1.000000        100.000000

        X1          5.000000           .000000
        E1           .000000           .000000
        X2         14.000000           .000000
        E2          4.000000           .000000
```

EXP2.LP

```
 1)  X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10    =  100   ! F

 2)  X1 + X3 + X5 + X7 + X9                              <=   75
 3)  X2 + X4 + X6 + X8 + X10                             <=   75
 4)  X1 + X2 + X3 + X4 + X5                              <=   75
 5)  X6 + X7 + X8 + X9 + X10                             <=   75

 6)  X2 - X1                                             >=    1
 7)  X3 - X2                                             >=    2
 8)  X4 - X3                                             >=    3
 9)  X5 - X4                                             >=    4
10)  X6 - X5                                             >=    5
11)  X7 - X6                                             >=    6
12)  X8 - X7                                             >=    7
13)  X9 - X8                                             >=    8
14)  X10 - X9                                            >=    9

15)  2 X1 - X2                                           >=    0
16)  3 X1 - X3                                           >=    0
17)  4 X1 - X4                                           >=    0
18)  5 X1 - X5                                           >=    0
19)  6 X1 - X6                                           >=    0
20)  7 X1 - X7                                           >=    0
21)  8 X1 - X8                                           >=    0
22)  9 X1 - X9                                           >=    0
23) 10 X1 - X10                                          >=    0

24)  2 X1 - X10                                           =    0
25)  X6 - X5                                             >=   50
26)  X6 - X5                                              =    0
27)  X1 + X10 - X5 - X6                                   =    0
28)  X1 + X2 + X3 + X4 + X5 - X6 - X7 - X8 - X9 - X10     =    0
29)  X1 + X3 + X5 + X7 + X9 - X2 - X4 - X6 - X8 - X10     =    0
30)  X5 + X7                                             >=   75
```

EXP2.PMIP

```
MIN 100 C2  + 100 C3  + 100 C4  + 100 C5  + 100 C6  +
    100 C7  + 100 C8  + 100 C9  + 100 C10 + 100 C11 +
    100 C12 + 100 C13 + 100 C14 + 100 C15 + 100 C16 +
    100 C17 + 100 C18 + 100 C19 + 100 C20 + 100 C21 +
    100 C22 + 100 C23 + 100 C24 + 100 C25 + 100 C26 +
    100 C27 + 100 C28 + 100 C29 + 100 C30
ST
  X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10                    <= 100 ! F
  X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10                    >= 100 ! F
  X1 + X3 + X5 + X7 + X9                                - 10000  C2 <=  75
  X2 + X4 + X6 + X8 + X10                               - 10000  C3 <=  75
  X1 + X2 + X3 + X4 + X5                                - 10000  C4 <=  75
  X6 + X7 + X8 + X9 + X10                               - 10000  C5 <=  75
  X2 - X1                                               + 10000  C6 >=   1
  X3 - X2                                               + 10000  C7 >=   2
  X4 - X3                                               + 10000  C8 >=   3
  X5 - X4                                               + 10000  C9 >=   4
  X6 - X5                                               + 10000 C10 >=   5
  X7 - X6                                               + 10000 C11 >=   6
  X8 - X7                                               + 10000 C12 >=   7
  X9 - X8                                               + 10000 C13 >=   8
  X10 - X9                                              + 10000 C14 >=   9
  2X1 - X2                                              + 10000 C15 >=   0
  3X1 - X3                                              + 10000 C16 >=   0
  4X1 - X4                                              + 10000 C17 >=   0
  5X1 - X5                                              + 10000 C18 >=   0
  6X1 - X6                                              + 10000 C19 >=   0
  7X1 - X7                                              + 10000 C20 >=   0
  8X1 - X8                                              + 10000 C21 >=   0
  9X1 - X9                                              + 10000 C22 >=   0
  10X1 - X10                                            + 10000 C23 >=   0
  2X1 - X10                                             - 10000 C24 <=   0
  2X1 - X10                                             + 10000 C24 >=   0
  X6 - X5                                               + 10000 C25 >=  50
  X6 - X5                                               - 10000 C26 <=   0
  X6 - X5                                               + 10000 C26 >=   0
  X1 + X10 - X5 - X6                                    - 10000 C27 <=   0
  X1 + X10 - X5 - X6                                    + 10000 C27 >=   0
  X1 + X2 + X3 + X4 + X5 - X6 - X7 - X8 - X9 - X10      - 10000 C28 <=   0
  X1 + X2 + X3 + X4 + X5 - X6 - X7 - X8 - X9 - X10      + 10000 C28 >=   0
  X1 + X3 + X5 + X7 + X9 - X2 - X4 - X6 - X8 - X10      - 10000 C29 <=   0
  X1 + X3 + X5 + X7 + X9 - X2 - X4 - X6 - X8 - X10      + 10000 C29 >=   0
  X5 + X7                                               + 10000 C30 >=  75
 END
 INTEGER 29
```

EXP2.S1

```
MIN 100 C2  + 100 C3  + 100 C4  + 100 C5  + 100 C6  +
    100 C7  + 100 C8  + 100 C9  + 100 C10 + 100 C11 +
    100 C12 + 100 C13 + 100 C14 + 100 C15 + 100 C16 +
    100 C17 + 100 C18 + 100 C19 + 100 C20 + 100 C21 +
    100 C22 + 100 C23 + 100 C24 + 100 C25 + 100 C26 +
    100 C27 + 100 C28 + 100 C29 + 100 C30
```

OBJECTIVE FUNCTION VALUE

1) 600.00000

VARIABLE	VALUE	REDUCED COST
C2	.000000	100.000000
C3	.000000	100.000000
C4	.000000	100.000000
C5	.000000	100.000000
C6	.000000	100.000000
C7	.000000	100.000000
C8	.000000	100.000000
C9	.000000	100.000000
C10	1.000000	100.000000
C11	.000000	100.000000
C12	.000000	100.000000
C13	1.000000	100.000000
C14	.000000	100.000000
C15	.000000	100.000000
C16	.000000	100.000000
C17	.000000	100.000000
C18	.000000	100.000000
C19	.000000	100.000000
C20	.000000	100.000000
C21	.000000	100.000000
C22	.000000	100.000000
C23	.000000	100.000000
C24	.000000	100.000000
C25	1.000000	100.000000
C26	1.000000	100.000000
C27	1.000000	100.000000
C28	.000000	100.000000
C29	.000000	100.000000
C30	1.000000	100.000000
X1	4.500000	.000000
X2	5.500000	.000000
X3	7.500000	.000000
X4	10.500000	.000000
X5	22.000000	.000000
X6	.000000	.000000
X7	16.000000	.000000
X8	25.000000	.000000
X9	.000000	.000000
X10	9.000000	.000000

EXP2.S2

```
MIN    1 C2  +   1 C3  +   1 C4  +   1 C5  +   1 C6  +
       1 C7  +   1 C8  +   1 C9  + 100 C10 +   1 C11 +
       1 C12 + 100 C13 +   1 C14 +   1 C15 +   1 C16 +
       1 C17 +   1 C18 +   1 C19 +   1 C20 +   1 C21 +
       1 C22 +   1 C23 +   1 C24 + 100 C25 + 100 C26 +
     100 C27 +   1 C28 +   1 C29 + 100 C30
```

OBJECTIVE FUNCTION VALUE

1) 209.00000

VARIABLE	VALUE	REDUCED COST
C2	1.000000	1.000000
C3	.000000	1.000000
C4	.000000	1.000000
C5	1.000000	1.000000
C6	1.000000	1.000000
C7	.000000	1.000000
C8	1.000000	1.000000
C9	.000000	1.000000
C10	1.000000	100.000000
C11	.000000	1.000000
C12	1.000000	1.000000
C13	.000000	100.000000
C14	1.000000	1.000000
C15	.000000	1.000000
C16	.000000	1.000000
C17	.000000	1.000000
C18	.000000	1.000000
C19	.000000	1.000000
C20	.000000	1.000000
C21	.000000	1.000000
C22	.000000	1.000000
C23	.000000	1.000000
C24	1.000000	1.000000
C25	1.000000	100.000000
C26	.000000	100.000000
C27	.000000	100.000000
C28	1.000000	1.000000
C29	1.000000	1.000000
C30	.000000	100.000000
X1	10.000000	.000000
X2	.000000	.000000
X3	2.000000	.000000
X4	.000000	.000000
X5	5.000000	.000000
X6	5.000000	.000000
X7	70.000000	.000000
X8	.000000	.000000
X9	8.000000	.000000
X10	.000000	.000000

EXP2.S3

```
MIN    1 C2  +    1 C3  +    1 C4  +    1 C5  +    1 C6  +
       1 C7  +    1 C8  +    1 C9  +  100 C10 +    1 C11 +
       1 C12 +    1 C13 +    1 C14 +    1 C15 +    1 C16 +
       1 C17 +    1 C18 +    1 C19 +    1 C20 +    1 C21 +
       1 C22 +    1 C23 +    1 C24 +  100 C25 +    1 C26 +
       1 C27 +    1 C28 +    1 C29 +    1 C30
```

OBJECTIVE FUNCTION VALUE

1) 10.000000

VARIABLE	VALUE	REDUCED COST
C2	.000000	1.000000
C3	.000000	1.000000
C4	.000000	1.000000
C5	.000000	1.000000
C6	1.000000	1.000000
C7	.000000	1.000000
C8	1.000000	1.000000
C9	.000000	1.000000
C10	.000000	100.000000
C11	1.000000	1.000000
C12	.000000	1.000000
C13	1.000000	1.000000
C14	.000000	1.000000
C15	.000000	1.000000
C16	.000000	1.000000
C17	.000000	1.000000
C18	.000000	1.000000
C19	.000000	1.000000
C20	.000000	1.000000
C21	.000000	1.000000
C22	.000000	1.000000
C23	.000000	1.000000
C24	1.000000	1.000000
C25	.000000	100.000000
C26	1.000000	1.000000
C27	1.000000	1.000000
C28	1.000000	1.000000
C29	1.000000	1.000000
C30	1.000000	1.000000
X1	9.000000	.000000
X2	.000000	.000000
X3	12.000000	.000000
X4	.000000	.000000
X5	4.000000	.000000
X6	54.000000	.000000
X7	.000000	.000000
X8	12.000000	.000000
X9	.000000	.000000
X10	9.000000	.000000

EXP3.LP

```
 1)  X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10    =  100   ! F
 2)  X1 + X3 + X5 + X7 + X9                             <=   75   ! F
 3)  X2 + X4 + X6 + X8 + X10                            <=   75   ! F
 4)  X1 + X2 + X3 + X4 + X5                             <=   75   ! F
 5)  X6 + X7 + X8 + X9 + X10                            <=   75   ! F

 6)  X2 - X1                                            >=    1
 7)  X3 - X2                                            >=    2
 8)  X4 - X3                                            >=    3
 9)  X5 - X4                                            >=    4
10)  X6 - X5                                            >=    5
11)  X7 - X6                                            >=    6
12)  X8 - X7                                            >=    7
13)  X9 - X8                                            >=    8
14)  X10 - X9                                           >=    9

15)  2 X1 - X2                                          >=    0
16)  3 X1 - X3                                          >=    0
17)  4 X1 - X4                                          >=    0
18)  5 X1 - X5                                          >=    0
19)  6 X1 - X6                                          >=    0
20)  7 X1 - X7                                          >=    0
21)  8 X1 - X8                                          >=    0
22)  9 X1 - X9                                          >=    0
23) 10 X1 - X10                                         >=    0

24)  2 X1 - X10                                          =    0
25)  X5 - X6                                            >=   10
26)  X6 - X5                                             =    0
27)  X1 + X10 - X5 - X6                                  =    0
28)  X1 + X2 + X3 + X4 + X5 - X6 - X7 - X8 - X9 - X10    =    0
29)  X1 + X3 + X5 + X7 + X9 - X2 - X4 - X6 - X8 - X10    =    0
30)  X5 + X7                                            >=   75
```

EXP3.PMIP

```
MIN 100 C6  + 100 C7  + 100 C8  + 100 C9  + 100 C10 +
    100 C11 + 100 C12 + 100 C13 + 100 C14 + 100 C15 +
    100 C16 + 100 C17 + 100 C18 + 100 C19 + 100 C20 +
    100 C21 + 100 C22 + 100 C23 + 100 C24 + 100 C25 +
    100 C26 + 100 C27 + 100 C28 + 100 C29 + 100 C30
ST
  X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10                    <= 100 ! F
  X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10                    >= 100 ! F
  X1 + X3 + X5 + X7 + X9                                              <=  75 ! F
  X2 + X4 + X6 + X8 + X10                                             <=  75 ! F
  X1 + X2 + X3 + X4 + X5                                              <=  75 ! F
  X6 + X7 + X8 + X9 + X10                                             <=  75 ! F
  X2 - X1                                          + 10000  C6 >=   1
  X3 - X2                                          + 10000  C7 >=   2
  X4 - X3                                          + 10000  C8 >=   3
  X5 - X4                                          + 10000  C9 >=   4
  X6 - X5                                          + 10000 C10 >=   5
  X7 - X6                                          + 10000 C11 >=   6
  X8 - X7                                          + 10000 C12 >=   7
  X9 - X8                                          + 10000 C13 >=   8
  X10 - X9                                         + 10000 C14 >=   9
  2X1 - X2                                         + 10000 C15 >=   0
  3X1 - X3                                         + 10000 C16 >=   0
  4X1 - X4                                         + 10000 C17 >=   0
  5X1 - X5                                         + 10000 C18 >=   0
  6X1 - X6                                         + 10000 C19 >=   0
  7X1 - X7                                         + 10000 C20 >=   0
  8X1 - X8                                         + 10000 C21 >=   0
  9X1 - X9                                         + 10000 C22 >=   0
  10X1 - X10                                       + 10000 C23 >=   0
  2X1 - X10                                        - 10000 C24 <=   0
  2X1 - X10                                        + 10000 C24 >=   0
  X5 - X6                                          + 10000 C25 >=  10
  X6 - X5                                          - 10000 C26 <=   0
  X6 - X5                                          + 10000 C26 >=   0
  X1 + X10 - X5 - X6                               - 10000 C27 <=   0
  X1 + X10 - X5 - X6                               + 10000 C27 >=   0
  X1 + X2 + X3 + X4 + X5 - X6 - X7 - X8 - X9 - X10 - 10000 C28 <=   0
  X1 + X2 + X3 + X4 + X5 - X6 - X7 - X8 - X9 - X10 + 10000 C28 >=   0
  X1 + X3 + X5 + X7 + X9 - X2 - X4 - X6 - X8 - X10 - 10000 C29 <=   0
  X1 + X3 + X5 + X7 + X9 - X2 - X4 - X6 - X8 - X10 + 10000 C29 >=   0
  X5 + X7                                          + 10000 C30 >=  75
END
INTEGER 25
```

EXP3.S1

```
MIN 100 C6  + 100 C7  + 100 C8  + 100 C9  + 100 C10 +
    100 C11 + 100 C12 + 100 C13 + 100 C14 + 100 C15 +
    100 C16 + 100 C17 + 100 C18 + 100 C19 + 100 C20 +
    100 C21 + 100 C22 + 100 C23 + 100 C24 + 100 C25 +
    100 C26 + 100 C27 + 100 C28 + 100 C29 + 100 C30
```

OBJECTIVE FUNCTION VALUE

1) 500.00000

VARIABLE	VALUE	REDUCED COST
C6	1.000000	100.000000
C7	.000000	100.000000
C8	.000000	100.000000
C9	.000000	100.000000
C10	1.000000	100.000000
C11	.000000	100.000000
C12	.000000	100.000000
C13	1.000000	100.000000
C14	.000000	100.000000
C15	.000000	100.000000
C16	.000000	100.000000
C17	.000000	100.000000
C18	.000000	100.000000
C19	.000000	100.000000
C20	.000000	100.000000
C21	.000000	100.000000
C22	.000000	100.000000
C23	.000000	100.000000
C24	.000000	100.000000
C25	.000000	100.000000
C26	1.000000	100.000000
C27	.000000	100.000000
C28	.000000	100.000000
C29	.000000	100.000000
C30	1.000000	100.000000
X1	6.100000	.000000
X2	.000000	.000000
X3	11.300000	.000000
X4	14.300000	.000000
X5	18.300000	.000000
X6	.000000	.000000
X7	11.100000	.000000
X8	23.500000	.000000
X9	3.200000	.000000
X10	12.200000	.000000

EXP3.S2

```
MIN 100 C6  +   1 C7  +   1 C8  +   1 C9  + 100 C10 +
      1 C11 +   1 C12 + 100 C13 +   1 C14 +   1 C15 +
      1 C16 +   1 C17 +   1 C18 +   1 C19 +   1 C20 +
      1 C21 +   1 C22 +   1 C23 +   1 C24 +   1 C25 +
    100 C26 +   1 C27 +   1 C28 +   1 C29 + 100 C30
```

OBJECTIVE FUNCTION VALUE

1) 204.00000

VARIABLE	VALUE	REDUCED COST
C6	.000000	100.000000
C7	.000000	1.000000
C8	.000000	1.000000
C9	1.000000	1.000000
C10	.000000	100.000000
C11	.000000	1.000000
C12	1.000000	1.000000
C13	.000000	100.000000
C14	.000000	1.000000
C15	.000000	1.000000
C16	.000000	1.000000
C17	.000000	1.000000
C18	.000000	1.000000
C19	.000000	1.000000
C20	.000000	1.000000
C21	.000000	1.000000
C22	.000000	1.000000
C23	.000000	1.000000
C24	.000000	1.000000
C25	1.000000	1.000000
C26	1.000000	100.000000
C27	1.000000	1.000000
C28	.000000	1.000000
C29	.000000	1.000000
C30	1.000000	100.000000
X1	8.500000	.000000
X2	9.500000	.000000
X3	13.500000	.000000
X4	16.500000	.000000
X5	2.000000	.000000
X6	7.000000	.000000
X7	18.000000	.000000
X8	.000000	.000000
X9	8.000000	.000000
X10	17.000000	.000000

EXP3.S3

```
MIN    1 C6  +    1 C7  +    1 C8  +    1 C9  +    1 C10 +
       1 C11 +    1 C12 +    1 C13 +    1 C14 +    1 C15 +
       1 C16 +    1 C17 +    1 C18 +    1 C19 +    1 C20 +
       1 C21 +    1 C22 +    1 C23 +    1 C24 +    1 C25 +
     100 C26 +    1 C27 +    1 C28 +    1 C29 +  100 C30
```

OBJECTIVE FUNCTION VALUE

1) 12.000000

VARIABLE	VALUE	REDUCED COST
C6	.000000	1.000000
C7	1.000000	1.000000
C8	.000000	1.000000
C9	1.000000	1.000000
C10	1.000000	1.000000
C11	.000000	1.000000
C12	1.000000	1.000000
C13	1.000000	1.000000
C14	1.000000	1.000000
C15	1.000000	1.000000
C16	.000000	1.000000
C17	1.000000	1.000000
C18	.000000	1.000000
C19	.000000	1.000000
C20	1.000000	1.000000
C21	.000000	1.000000
C22	.000000	1.000000
C23	.000000	1.000000
C24	.000000	1.000000
C25	1.000000	1.000000
C26	.000000	100.000000
C27	.000000	1.000000
C28	1.000000	1.000000
C29	1.000000	1.000000
C30	.000000	100.000000
X1	.000000	.000000
X2	1.000000	.000000
X3	.000000	.000000
X4	24.000000	.000000
X5	.000000	.000000
X6	.000000	.000000
X7	75.000000	.000000
X8	.000000	.000000
X9	.000000	.000000
X10	.000000	.000000

EXP4.LP

```
 1)   X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10    <=   500    ! F
 2)   X1 + 2 X2 + 3 X3 + 4 X4 + 5 X5                      >=   250    ! F
 3)   X6 + 2 X7 + 3 X8 + 4 X9 + 5 X10                     >=   500    ! F
 4)   X1 - X2 - X3                                        >=     0    ! F
 5)   X4 - X5 - X6                                        >=     0    ! F

 6)   X7 + X8 - X9 - X10                                  >=     0
 7)   X1 + X10                                            <=    50
 8)   X2 + X4 + X6 + X8 + X10                             <=    75
 9)   X5                                                  <=    25
10)   X5 + X10                                           >=   200
11)   X1                                                  >=    10
12)   X2                                                  >=    20
13)   X3                                                  >=    30
14)   X4                                                  >=    40
15)   X5                                                  >=    50
16)   X6                                                  >=    60
17)   X7                                                  >=    70
18)   X8                                                  >=    80
19)   X9                                                  >=    90
20)   X10                                                 >=   100
```

EXP4.PMIP

```
MIN 100 C6  + 100 C7  + 100 C8  + 100 C9  + 100 C10 +
    100 C11 + 100 C12 + 100 C13 + 100 C14 + 100 C15 +
    100 C16 + 100 C17 + 100 C18 + 100 C19 + 100 C20
ST
  X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10                <= 500 ! F
  X1 + 2 X2 + 3 X3 + 4 X4 + 5 X5                                  >= 250 ! F
  X6 + 2 X7 + 3 X8 + 4 X9 + 5 X10                                 >= 500 ! F
  X1 - X2 - X3                                                    >=   0 ! F
  X4 - X5 - X6                                                    >=   0 ! F
  X7 + X8 - X9 - X10                          + 10000  C6 >=   0
  X1 + X10                                    - 10000  C7 <=  50
  X2 + X4 + X6 + X8 + X10                     - 10000  C8 <=  75
  X5                                          - 10000  C9 <=  25
  X5 + X10                                    + 10000 C10 >= 200
  X1                                          + 10000 C11 >=  10
  X2                                          + 10000 C12 >=  20
  X3                                          + 10000 C13 >=  30
  X4                                          + 10000 C14 >=  40
  X5                                          + 10000 C15 >=  50
  X6                                          + 10000 C16 >=  60
  X7                                          + 10000 C17 >=  70
  X8                                          + 10000 C18 >=  80
  X9                                          + 10000 C19 >=  90
  X10                                         + 10000 C20 >= 100
END
INTEGER 15
```

EXP4.S1

```
MIN 100 C6  + 100 C7  + 100 C8  + 100 C9  + 100 C10 +
    100 C11 + 100 C12 + 100 C13 + 100 C14 + 100 C15 +
    100 C16 + 100 C17 + 100 C18 + 100 C19 + 100 C20
```

OBJECTIVE FUNCTION VALUE

1) 400.00000

VARIABLE	VALUE	REDUCED COST
C6	.000000	100.000000
C7	.000000	100.000000
C8	1.000000	100.000000
C9	.000000	100.000000
C10	1.000000	100.000000
C11	.000000	100.000000
C12	.000000	100.000000
C13	.000000	100.000000
C14	.000000	100.000000
C15	1.000000	100.000000
C16	.000000	100.000000
C17	.000000	100.000000
C18	.000000	100.000000
C19	.000000	100.000000
C20	1.000000	100.000000
X1	50.000000	.000000
X2	20.000000	.000000
X3	30.000000	.000000
X4	60.000000	.000000
X5	.000000	.000000
X6	60.000000	.000000
X7	70.000000	.000000
X8	80.000000	.000000
X9	90.000000	.000000
X10	.000000	.000000

EXP4.S2

```
MIN    1 C6  +    1 C7  + 100 C8  +    1 C9  + 100 C10 +
       1 C11 +    1 C12 +   1 C13 +    1 C14 + 100 C15 +
       1 C16 +    1 C17 +   1 C18 +    1 C19 +   1 C20
```

OBJECTIVE FUNCTION VALUE

1) 104.00000

VARIABLE	VALUE	REDUCED COST
C6	.000000	1.000000
C7	1.000000	1.000000
C8	1.000000	100.000000
C9	1.000000	1.000000
C10	.000000	100.000000
C11	.000000	1.000000
C12	.000000	1.000000
C13	.000000	1.000000
C14	.000000	1.000000
C15	.000000	100.000000
C16	1.000000	1.000000
C17	.000000	1.000000
C18	.000000	1.000000
C19	1.000000	1.000000
C20	.000000	100.000000
X1	50.000000	.000000
X2	20.000000	.000000
X3	30.000000	.000000
X4	50.000000	.000000
X5	50.000000	.000000
X6	.000000	.000000
X7	70.000000	.000000
X8	80.000000	.000000
X9	.000000	.000000
X10	150.000000	.000000

EXP4.S3

```
MIN    1 C6  +   1 C7  + 100 C8  +   1 C9  +   1 C10 +
       1 C11 +   1 C12 +   1 C13 +   1 C14 +   1 C15 +
       1 C16 +   1 C17 +   1 C18 +   1 C19 +   1 C20
```

OBJECTIVE FUNCTION VALUE

1) 5.0000000

VARIABLE	VALUE	REDUCED COST
C6	.000000	1.000000
C7	.000000	1.000000
C8	.000000	100.000000
C9	.000000	1.000000
C10	1.000000	1.000000
C11	.000000	1.000000
C12	.000000	1.000000
C13	.000000	1.000000
C14	.000000	1.000000
C15	1.000000	1.000000
C16	1.000000	1.000000
C17	.000000	1.000000
C18	1.000000	1.000000
C19	.000000	1.000000
C20	1.000000	1.000000
X1	50.000000	.000000
X2	20.000000	.000000
X3	30.000000	.000000
X4	40.000000	.000000
X5	25.000000	.000000
X6	15.000000	.000000
X7	90.000000	.000000
X8	.000000	.000000
X9	90.000000	.000000
X10	.000000	.000000

EXP5.LP

```
 ROW1)    0.8 X3 + X4 <=    10000
 ROW2)    X1 <=    90000
 ROW3)    2 X6 - X8 <=    10000
 ROW4)  - X2 + X3 >=    50000
 ROW5)  - X2 + X4 >=    87000
 ROW6)    X3 <=    50000
 ROW7)  - 3 X5 + X7 >=    10000
 ROW8)    0.5 X5 + 0.6 X6 <=    300000
 ROW9)    X2 - 0.05 X3 =     5000
ROW10)    X2 - 0.04 X3 - 0.05 X4 =     4500
ROW11)    X2 >=    80000
```

EXP5.PMIP

```
MIN 100 C1  + 100 C2  + 100 C3  + 100 C4  + 100 C5  + 100 C6  +
    100 C7  + 100 C8  + 100 C9  + 100 C10 + 100 C11
ST
  0.8 X3 + X4                        - 1000000   C1   <=   10000
  X1                                 - 1000000   C2   <=   90000
  2 X6 - X8                          - 1000000   C3   <=   10000
- X2 + X3                           + 1000000   C4   >=   50000
- X2 + X4                           + 1000000   C5   >=   87000
  X3                                 - 1000000   C6   <=   50000
- 3 X5 + X7                         + 1000000   C7   >=   10000
  0.5 X5 + 0.6 X6                    - 1000000   C8   <=  300000
  X2 - 0.05 X3                       - 1000000   C9   <=    5000
  X2 - 0.05 X3                       + 1000000   C9   >=    5000
  X2 - 0.04 X3 - 0.05 X4             - 1000000   C10  <=    4500
  X2 - 0.04 X3 - 0.05 X4             + 1000000   C10  >=    4500
  X2                                 + 1000000   C11  >=   80000
END
INTEGER 11
```

EXP5.S1

```
MIN 100 C1  + 100 C2  + 100 C3  + 100 C4  + 100 C5  + 100 C6  +
    100 C7  + 100 C8  + 100 C9  + 100 C10 + 100 C11

        OBJECTIVE FUNCTION VALUE

        1)      300.00000

  VARIABLE          VALUE          REDUCED COST
        C1           .000000        100.000000
        C2           .000000        100.000000
        C3           .000000        100.000000
        C4          1.000000        100.000000
        C5          1.000000        100.000000
        C6           .000000        100.000000
        C7           .000000        100.000000
        C8           .000000        100.000000
        C9           .000000        100.000000
       C10           .000000        100.000000
       C11          1.000000        100.000000

        X3           .000000           .000000
        X4      10000.000000           .000000
        X1           .000000           .000000
        X6           .000000           .000000
        X8           .000000           .000000
        X2       5000.000000           .000000
        X5           .000000           .000000
        X7      10000.000000           .000000
```

EXP5.S2

```
MIN    1 C1  +    1 C2  +    1 C3  + 100 C4  + 100 C5  +    1 C6  +
       1 C7  +    1 C8  +    1 C9  +    1 C10 + 100 C11
```

OBJECTIVE FUNCTION VALUE

1) 4.0000000

VARIABLE	VALUE	REDUCED COST
C1	1.000000	1.000000
C2	.000000	1.000000
C3	.000000	1.000000
C4	.000000	100.000000
C5	.000000	100.000000
C6	1.000000	1.000000
C7	.000000	1.000000
C8	.000000	1.000000
C9	1.000000	1.000000
C10	1.000000	1.000000
C11	.000000	100.000000
X3	1050000.000000	.000000
X4	170000.000000	.000000
X1	.000000	.000000
X6	.000000	.000000
X8	.000000	.000000
X2	80000.000000	.000000
X5	.000000	.000000
X7	10000.000000	.000000

EXP6.LP

```
ROW1)  - 0.5 X1 + X2 >=    0.5
ROW2)    2 X1 - X2 >=    3
ROW3)    3 X1 + X2 <=    6
ROW4)    X5 <=    2
ROW5)    3 X4 - X5 <=    2
ROW6)    X4 >=    5
ROW7)    X1 + X5 <=    10
ROW8)    X1 + 2 X2 + X4 <=    14
ROW9)    X2 + X4 >=    1
```

EXP6.PMIP

```
MIN 100 C1  + 100 C2  + 100 C3  + 100 C5  + 100 C6  +
    100 C7  + 100 C8  + 100 C9
ST
  - 0.5 X1 + X2                + 10000 C1  >=    0.5
    2 X1 - X2                  + 10000 C2  >=    3
    3 X1 + X2                  - 10000 C3  <=    6
    X5                         - 10000 C4  <=    2
    3 X4 - X5                  - 10000 C5  <=    2
    X4                         + 10000 C6  >=    5
    X1 + X5                    - 10000 C7  <=   10
    X1 + 2 X2 + X4             - 10000 C8  <=   14
    X2 + X4                    + 10000 C9  >=    1
END
INTEGER 9
```

EXP6.S1

```
MIN 100 C1  + 100 C2  + 100 C3  + 100 C5  + 100 C6  +
    100 C7  + 100 C8  + 100 C9

        OBJECTIVE FUNCTION VALUE

        1)     200.00000

  VARIABLE         VALUE          REDUCED COST
        C1          1.000000        100.000000
        C2           .000000        100.000000
        C3           .000000        100.000000
        C4           .000000        100.000000
        C5           .000000        100.000000
        C6          1.000000        100.000000
        C7           .000000        100.000000
        C8           .000000        100.000000
        C9           .000000        100.000000

        X1          1.800000           .000000
        X2           .600000           .000000
        X5          2.000000           .000000
        X4          1.333333           .000000
```

EXP6.S2

```
MIN 100 C1  +   1 C2  +   1 C3  +   1 C5  + 100 C6  +
      1 C7  +   1 C8  +   1 C9

        OBJECTIVE FUNCTION VALUE

        1)     2.0000000

  VARIABLE         VALUE          REDUCED COST
        C1           .000000        100.000000
        C2          1.000000          1.000000
        C3           .000000          1.000000
        C4           .000000          1.000000
        C5          1.000000          1.000000
        C6           .000000        100.000000
        C7           .000000          1.000000
        C8           .000000          1.000000
        C9           .000000          1.000000

        X1           .000000           .000000
        X2           .500000           .000000
        X5           .000000           .000000
        X4         13.000000           .000000
```

EXP7.LP

```
   S1)    T14 <=    20
   S2)    T24 + T25 <=    20
   S3)    T35 <=    20
NODE4)    T14 + T24 - T46 - T47 =     0
NODE5)    T25 + T35 - T57 - T58 =     0
   D6)    T46 >=    10
   D7)    T47 + T57 >=    20
   D8)    T58 >=    30

  SUB       T14         30.00000
  SUB       T24         20.00000
  SUB       T25         10.00000
  SUB       T35         10.00000
  SUB       T46         10.00000
  SUB       T47          2.00000
  SUB       T57         20.00000
  SUB       T58         30.00000
```

EXP7.PMIP

```
MIN 100 C1  + 100 C2  + 100 C3  + 100 C4  + 100 C5  + 100 C6  +
    100 C7  + 100 C8  + 100 C9  + 100 C10 + 100 C11 + 100 C12 +
    100 C13 + 100 C14 + 100 C15 + 100 C16
ST
  T14                                 - 10000 C1    <=   20
  T24 + T25                           - 10000 C2    <=   20
  T35                                 - 10000 C3    <=   20
  T14 + T24 - T46 - T47               - 10000 C4    <=    0
  T14 + T24 - T46 - T47               + 10000 C4    >=    0
  T25 + T35 - T57 - T58               - 10000 C5    <=    0
  T25 + T35 - T57 - T58               + 10000 C5    >=    0
  T46                                 + 10000 C6    >=   10
  T47 + T57                           + 10000 C7    >=   20
  T58                                 + 10000 C8    >=   30
  T14                                 - 10000 C9    <=   30
  T24                                 - 10000 C10   <=   20
  T25                                 - 10000 C11   <=   10
  T35                                 - 10000 C12   <=   10
  T46                                 - 10000 C13   <=   10
  T47                                 - 10000 C14   <=    2
  T57                                 - 10000 C15   <=   20
  T58                                 - 10000 C16   <=   30
 END
 INTEGER 16
```

EXP7.S1

```
MIN 100 C1  + 100 C2  + 100 C3  + 100 C4  + 100 C5  + 100 C6  +
    100 C7  + 100 C8  + 100 C9  + 100 C10 + 100 C11 + 100 C12 +
    100 C13 + 100 C14 + 100 C15 + 100 C16

        OBJECTIVE FUNCTION VALUE

        1)    100.0000000

  VARIABLE         VALUE          REDUCED COST
       C1          .000000         100.000000
       C2          .000000         100.000000
       C3          .000000         100.000000
       C4          .000000         100.000000
       C5          .000000         100.000000
       C6          .000000         100.000000
       C7          .000000         100.000000
       C8         1.000000         100.000000
       C9          .000000         100.000000
      C10          .000000         100.000000
      C11          .000000         100.000000
      C12          .000000         100.000000
      C13          .000000         100.000000
      C14          .000000         100.000000
      C15          .000000         100.000000
      C16          .000000         100.000000

      T14         2.000000            .000000
      T24        10.000000            .000000
      T25        10.000000            .000000
      T35        10.000000            .000000
      T46        10.000000            .000000
      T47         2.000000            .000000
      T57        18.000000            .000000
      T58         2.000000            .000000
```

EXP7.S2

```
MIN    1 C1  +   1 C2  +   1 C3  +   1 C4  +   1 C5  +   1 C6  +
       1 C7  + 100 C8  +   1 C9  +   1 C10 +   1 C11 +   1 C12 +
       1 C13 +   1 C14 +   1 C15 +   1 C16
```

OBJECTIVE FUNCTION VALUE

1) 1.0000000

VARIABLE	VALUE	REDUCED COST
C1	.000000	1.000000
C2	.000000	1.000000
C3	.000000	1.000000
C4	.000000	1.000000
C5	1.000000	1.000000
C6	.000000	1.000000
C7	.000000	1.000000
C8	.000000	100.000000
C9	.000000	1.000000
C10	.000000	1.000000
C11	.000000	1.000000
C12	.000000	1.000000
C13	.000000	1.000000
C14	.000000	1.000000
C15	.000000	1.000000
C16	.000000	1.000000
T14	2.000000	.000000
T24	10.000000	.000000
T25	10.000000	.000000
T35	10.000000	.000000
T46	10.000000	.000000
T47	2.000000	.000000
T57	18.000000	.000000
T58	30.000000	.000000

EXP8.LP

```
1)        X1 +     X2  <=   2
2)      6 X1 +   5 X2  <=  30
3)  -     X1 +     X2  >=   3
4)     11 X1 -   3 X2  >=  33
5)      4 X1 -   7 X2  >=  28
```

EXP8.PMIP

```
MIN 100 C1 + 100 C2 + 100 C3 + 100 C4 + 100 C5
ST
         X1 +     X2 - 10000 C1  <=   2
       6 X1 +   5 X2 - 10000 C2  <=  30
   -     X1 +     X2 + 10000 C3  >=   3
      11 X1 -   3 X2 + 10000 C4  >=  33
       4 X1 -   7 X2 + 10000 C5  >=  28
END
INTEGER 5
```

EXP8.S1

```
MIN 100 C1 + 100 C2 + 100 C3 + 100 C4 + 100 C5

        OBJECTIVE FUNCTION VALUE

        1)      300.00000

  VARIABLE          VALUE          REDUCED COST
        C1          1.000000         100.000000
        C2           .000000         100.000000
        C3          1.000000         100.000000
        C4           .000000         100.000000
        C5          1.000000         100.000000

        X1          3.493151            .000000
        X2          1.808219            .000000
```

EXP8.S2

```
MIN 100 C1 +   1 C2 + 100 C3 +   1 C4 + 100 C5

        OBJECTIVE FUNCTION VALUE

        1)      201.00000

  VARIABLE          VALUE          REDUCED COST
        C1          1.000000         100.000000
        C2          1.000000           1.000000
        C3          1.000000         100.000000
        C4           .000000           1.000000
        C5           .000000         100.000000

        X1          7.000000            .000000
        X2           .000000            .000000
```

EXP8.S3

```
MIN 100 C1 +   1 C2 + 100 C3 +   1 C4 +   1 C5

        OBJECTIVE FUNCTION VALUE

        1)      102.00000

  VARIABLE          VALUE          REDUCED COST
        C1           .000000        100.000000
        C2           .000000          1.000000
        C3          1.000000        100.000000
        C4          1.000000          1.000000
        C5          1.000000          1.000000

        X1           .000000           .000000
        X2          2.000000           .000000
```

EXP8.S4

```
MIN   1 C1 +   1 C2 + 100 C3 +   1 C4 +   1 C5

        OBJECTIVE FUNCTION VALUE

        1)      3.0000000

  VARIABLE          VALUE          REDUCED COST
        C1          1.000000          1.000000
        C2           .000000          1.000000
        C3           .000000        100.000000
        C4          1.000000          1.000000
        C5          1.000000          1.000000

        X1           .000000           .000000
        X2          6.000000           .000000
```

EXP9.LP

```
1)      4 X1 +   3 X2   <=  12
2)   -  2 X1 +      X2   >=   2
3)   -     X1 +      X2   >=   6
4)     16 X1 + 17 X2   >= 272
5)      2 X1 -   7 X2   >=  14
```

EXP9.PMIP

```
MIN 100 C1 + 100 C2 + 100 C3 + 100 C4 + 100 C5
ST
       4 X1 +    3 X2 - 10000 C1   <=  12
    -  2 X1 +       X2 + 10000 C2   >=   2
    -     X1 +       X2 + 10000 C3   >=   6
      16 X1 +   17 X2 + 10000 C4   >= 272
       2 X1 -    7 X2 + 10000 C5   >=  14
END
INTEGER 5
```

EXP9.S1

```
MIN 100 C1 + 100 C2 + 100 C3 + 100 C4 + 100 C5

        OBJECTIVE FUNCTION VALUE

        1)      200.00000

  VARIABLE         VALUE          REDUCED COST
        C1          1.000000        100.000000
        C2           .000000        100.000000
        C3           .000000        100.000000
        C4           .000000        100.000000
        C5          1.000000        100.000000

        X1           .000000           .000000
        X2         16.000000           .000000
```

EXP9.S2

```
MIN 100 C1 +   1 C2 +   1 C3 +   1 C4 + 100 C5

        OBJECTIVE FUNCTION VALUE

        1)      102.00000

  VARIABLE         VALUE          REDUCED COST
        C1           .000000        100.000000
        C2           .000000          1.000000
        C3          1.000000          1.000000
        C4          1.000000          1.000000
        C5          1.000000        100.000000

        X1           .000000           .000000
        X2          2.000000           .000000
```

EXP9.S3

```
MIN   1 C1 +   1 C2 +   1 C3 +   1 C4 + 100 C5

        OBJECTIVE FUNCTION VALUE

        1)      3.0000000

  VARIABLE          VALUE          REDUCED COST
        C1          1.000000          1.000000
        C2          1.000000          1.000000
        C3          1.000000          1.000000
        C4           .000000          1.000000
        C5           .000000        100.000000

        X1         14.671230           .000000
        X2          2.191781           .000000
```

EXP10.LP

```
 1) X1  + X2  + X3  + X4  + X5  +
    X6  + X7  + X8  + X9  + X10 +
    X11 + X12 + X13 + X14 + X15 +
    X16 + X17 + X18 + X19 + X20 +
    X21 + X22 + X23 + X24 + X25          >= 1000    ! F

 2) X1  + X2  + X3  + X4  + X5  +
    X6  + X7  + X8  + X9  + X10 +
    X11 + X12 + X13 + X14 + X15 +
    X16 + X17 + X18 + X19 + X20 +
    X21 + X22 + X23 + X24 + X25          <= 3000    ! F

 3) X1  + X2  + X3  + X4  + X5           >=  100    ! F
 4) X6  + X7  + X8  + X9  + X10          >=  100    ! F
 5) X11 + X12 + X13 + X14 + X15          >=  100    ! F
 6) X16 + X17 + X18 + X19 + X20          >=  100    ! F
 7) X21 + X22 + X23 + X24 + X25          >=  100    ! F

 8) X1  + X6  + X11 + X16 + X21          >=  100    ! F
 9) X2  + X7  + X12 + X17 + X22          >=  100    ! F
10) X3  + X8  + X13 + X18 + X23          >=  100    ! F
11) X4  + X9  + X14 + X19 + X24          >=  100    ! F
12) X5  + X10 + X15 + X20 + X25          >=  100    ! F

13) X1                                   >=   10    ! F
14) X2                                   >=   10    ! F
15) X3                                   >=   10    ! F
16) X4                                   >=   10    ! F
17) X5                                   >=   10    ! F
18) X6                                   >=   10    ! F
19) X7                                   >=   10    ! F
20) X8                                   >=   10    ! F
21) X9                                   >=   10    ! F
22) X10                                  >=   10    ! F
23) X11                                  >=   10    ! F
24) X12                                  >=   10    ! F
25) X13                                  >=   10    ! F
26) X14                                  >=   10    ! F
27) X15                                  >=   10    ! F
28) X16                                  >=   10    ! F
29) X17                                  >=   10    ! F
30) X18                                  >=   10    ! F
31) X19                                  >=   10    ! F
32) X20                                  >=   10    ! F
33) X21                                  >=   10    ! F
34) X22                                  >=   10    ! F
35) X23                                  >=   10    ! F
36) X24                                  >=   10    ! F
37) X25                                  >=   10    ! F
```

```
38) X1  + X2  + X3  + X4  + X5                    >=  500
39) X6  + X7  + X8  + X9  + X10                   >=  500
40) X11 + X12 + X13 + X14 + X15                   >=  500
41) X16 + X17 + X18 + X19 + X20                   >=  500
42) X21 + X22 + X23 + X24 + X25                   >=  500

43) X1  + X6  + X11 + X16 + X21                   >=  500
44) X2  + X7  + X12 + X17 + X22                   >=  500
45) X3  + X8  + X13 + X18 + X23                   >=  500
46) X4  + X9  + X14 + X19 + X24                   >=  500
47) X5  + X10 + X15 + X20 + X25                   >=  500

48) X1  + X2  + X3  + X4  + X5  +
    X6  + X7  + X8  + X9  + X10 +
    X11 + X12 + X13 + X14 + X15 +
    X16 + X17 + X18 + X19 + X20 +
    X21 + X22 + X23 + X24 + X25                   <= 1250

49) X1  + X2  + X3  + X4  + X5  +
    X6  + X7  + X8  + X9  + X10 +
    X11 + X12 + X13 + X14 + X15 +
    X16 + X17 + X18 + X19 + X20 +
    X21 + X22 + X23 + X24 + X25                   >= 2750

50) X1  + X2  + X3  + X4  + X5  +
    X6  + X7  + X8  + X9  + X10 +
    X11 + X12 + X13 + X14 + X15 +
    X16 + X17 + X18 + X19 + X20 +
    X21 + X22 + X23 + X24 + X25                   >= 1750   ! &

51) X1  + X2  + X3  + X4  + X5  +
    X6  + X7  + X8  + X9  + X10 +
    X11 + X12 + X13 + X14 + X15 +
    X16 + X17 + X18 + X19 + X20 +
    X21 + X22 + X23 + X24 + X25                   <= 2250

52) X6  + X7  + X8  + X9  + X10 -
    X1  - X2  - X3  - X4  - X5                    >=  100   ! &
53) X11 + X12 + X13 + X14 + X15 -
    X6  - X7  - X8  - X9  - X10                   >=  100   ! &
54) X16 + X17 + X18 + X19 + X20 -
    X11 - X12 - X13 - X14 - X15                   >=  100   ! &
55) X21 + X22 + X23 + X24 + X25 -
    X16 - X17 - X18 - X19 - X20                   >=  100

56) X2  + X7  + X12 + X17 + X22 -
    X1  - X6  - X11 - X16 - X21                   >=  100   ! &
57) X3  + X8  + X13 + X18 + X23 -
    X2  - X7  - X12 - X17 - X22                   >=  100   ! &
58) X4  + X9  + X14 + X19 + X24 -
    X3  - X8  - X13 - X18 - X23                   >=  100   ! &
59) X5  + X10 + X15 + X20 + X25 -
    X4  - X9  - X14 - X19 - X24                   >=  100
```

```
60) X1  - X6                                    >=    0   ! &
61) X6  - X11                                   >=    0   ! &
62) X11 - X16                                   >=    0   ! &
63) X16 - X21                                   >=    0

64) X10 - X5                                    >=    0   ! &
65) X15 - X10                                   >=    0   ! &
66) X20 - X15                                   >=    0   ! &
67) X25 - X20                                   >=    0

68) X1                                          >=   25   ! &
69) X2                                          >=   45   ! &
70) X3                                          >=   50   ! &
71) X4                                          >=   55   ! &
72) X5                                          >=   75   ! &
73) X6                                          >=   30   ! &
74) X7                                          >=   40   ! &
75) X8                                          >=   50   ! &
76) X9                                          >=   60   ! &
77) X10                                         >=   70   ! &
78) X11                                         >=   50   ! &
79) X12                                         >=   50   ! &
80) X13                                         >=   50   ! &
81) X14                                         >=   50   ! &
82) X15                                         >=   50   ! &
83) X16                                         >=   30   ! &
84) X17                                         >=   40   ! &
85) X18                                         >=   50   ! &
86) X19                                         >=   60   ! &
87) X20                                         >=   70   ! &
88) X21                                         >=   25   ! &
89) X22                                         >=   45   ! &
90) X23                                         >=   50   ! &
91) X24                                         >=   55   ! &
92) X25                                         >=   75

93) X5  + X10 + X15 + X20 + X25                 <=  750   ! CV X1
94) X21 + X22 + X23 + X24 + X25                 <=  750   ! CV X1
95) X1  + X6  + X11 + X16 + X21                 >=  250   ! CV X25
96) X1  + X2  + X3  + X4  + X5                  >=  250   ! CV X25
```

EXP10.PMIP/B

```
MIN  100 C38 + 100 C39 + 100 C40 + 100 C41 + 100 C42 +
     100 C43 + 100 C44 + 100 C45 + 100 C46 + 100 C47 +
     100 C48 + 100 C49 + 100 C50 + 100 C52 + 100 C56 +
     100 C60 + 100 C64 + 100 C68 + 100 C93 + 100 C94 +
     100 C95 + 100 C96
ST
  1) X1  + X2  + X3  + X4  + X5  +
     X6  + X7  + X8  + X9  + X10 +
     X11 + X12 + X13 + X14 + X15 +
     X16 + X17 + X18 + X19 + X20 +
     X21 + X22 + X23 + X24 + X25          >= 1000    ! F
  2) X1  + X2  + X3  + X4  + X5  +
     X6  + X7  + X8  + X9  + X10 +
     X11 + X12 + X13 + X14 + X15 +
     X16 + X17 + X18 + X19 + X20 +
     X21 + X22 + X23 + X24 + X25 +
                       E1  + E25          <= 3000    ! F
  3) X1  + X2  + X3  + X4  + X5           >=  100    ! F
  4) X6  + X7  + X8  + X9  + X10          >=  100    ! F
  5) X11 + X12 + X13 + X14 + X15          >=  100    ! F
  6) X16 + X17 + X18 + X19 + X20          >=  100    ! F
  7) X21 + X22 + X23 + X24 + X25          >=  100    ! F
  8) X1  + X6  + X11 + X16 + X21          >=  100    ! F
  9) X2  + X7  + X12 + X17 + X22          >=  100    ! F
 10) X3  + X8  + X13 + X18 + X23          >=  100    ! F
 11) X4  + X9  + X14 + X19 + X24          >=  100    ! F
 12) X5  + X10 + X15 + X20 + X25          >=  100    ! F
 13) X1                                   >=   10    ! F
 14) X2                                   >=   10    ! F
 15) X3                                   >=   10    ! F
 16) X4                                   >=   10    ! F
 17) X5                                   >=   10    ! F
 18) X6                                   >=   10    ! F
 19) X7                                   >=   10    ! F
 20) X8                                   >=   10    ! F
 21) X9                                   >=   10    ! F
 22) X10                                  >=   10    ! F
 23) X11                                  >=   10    ! F
 24) X12                                  >=   10    ! F
 25) X13                                  >=   10    ! F
 26) X14                                  >=   10    ! F
 27) X15                                  >=   10    ! F
 28) X16                                  >=   10    ! F
 29) X17                                  >=   10    ! F
 30) X18                                  >=   10    ! F
 31) X19                                  >=   10    ! F
 32) X20                                  >=   10    ! F
 33) X21                                  >=   10    ! F
 34) X22                                  >=   10    ! F
 35) X23                                  >=   10    ! F
 36) X24                                  >=   10    ! F
 37) X25                                  >=   10    ! F
```

```
38) X1  + X2  + X3  + X4  + X5  + 10000 C38      >=  500
39) X6  + X7  + X8  + X9  + X10 + 10000 C39      >=  500
40) X11 + X12 + X13 + X14 + X15 + 10000 C40      >=  500
41) X16 + X17 + X18 + X19 + X20 + 10000 C41      >=  500
42) X21 + X22 + X23 + X24 + X25 + 10000 C42      >=  500
43) X1  + X6  + X11 + X16 + X21 + 10000 C43      >=  500
44) X2  + X7  + X12 + X17 + X22 + 10000 C44      >=  500
45) X3  + X8  + X13 + X18 + X23 + 10000 C45      >=  500
46) X4  + X9  + X14 + X19 + X24 + 10000 C46      >=  500
47) X5  + X10 + X15 + X20 + X25 + 10000 C47      >=  500
48) X1  + X2  + X3  + X4  + X5  +
    X6  + X7  + X8  + X9  + X10 +
    X11 + X12 + X13 + X14 + X15 +
    X16 + X17 + X18 + X19 + X20 +
    X21 + X22 + X23 + X24 + X25 +
                      E1  + E25 - 10000 C48      <= 1250
49) X1  + X2  + X3  + X4  + X5  +
    X6  + X7  + X8  + X9  + X10 +
    X11 + X12 + X13 + X14 + X15 +
    X16 + X17 + X18 + X19 + X20 +
    X21 + X22 + X23 + X24 + X25 + 10000 C49      >= 2750
50) X1  + X2  + X3  + X4  + X5  +
    X6  + X7  + X8  + X9  + X10 +
    X11 + X12 + X13 + X14 + X15 +
    X16 + X17 + X18 + X19 + X20 +
    X21 + X22 + X23 + X24 + X25 + 10000 C50      >= 1750
51) X1  + X2  + X3  + X4  + X5  +
    X6  + X7  + X8  + X9  + X10 +
    X11 + X12 + X13 + X14 + X15 +
    X16 + X17 + X18 + X19 + X20 +
    X21 + X22 + X23 + X24 + X25 +
                      E1  + E25 - 10000 C50      <= 2250
52) X6  + X7  + X8  + X9  + X10 - E1 -
    X1  - X2  - X3  - X4  - X5  + 10000 C52      >=  100   ! &
53) X11 + X12 + X13 + X14 + X15 -
    X6  - X7  - X8  - X9  - X10 + 10000 C52      >=  100   ! &
54) X16 + X17 + X18 + X19 + X20 -
    X11 - X12 - X13 - X14 - X15 + 10000 C52      >=  100   ! &
55) X21 + X22 + X23 + X24 + X25 -
    X16 - X17 - X18 - X19 - X20 + 10000 C52      >=  100
56) X2  + X7  + X12 + X17 + X22 - E1 -
    X1  - X6  - X11 - X16 - X21 + 10000 C56      >=  100   ! &
57) X3  + X8  + X13 + X18 + X23 -
    X2  - X7  - X12 - X17 - X22 + 10000 C56      >=  100   ! &
58) X4  + X9  + X14 + X19 + X24 -
    X3  - X8  - X13 - X18 - X23 + 10000 C56      >=  100   ! &
59) X5  + X10 + X15 + X20 + X25 -
    X4  - X9  - X14 - X19 - X24 + 10000 C56      >=  100
60) X1  - X6                    + 10000 C60      >=    0   ! &
61) X6  - X11                   + 10000 C60      >=    0   ! &
62) X11 - X16                   + 10000 C60      >=    0   ! &
63) X16 - X21                   + 10000 C60      >=    0
64) X10 - X5                    + 10000 C64      >=    0   ! &
```

```
65) X15 - X10                                + 10000 C64      >=     0    ! &
66) X20 - X15                                + 10000 C64      >=     0    ! &
67) X25 - X20                                + 10000 C64      >=     0
68) X1                                       + 10000 C68      >=    25    ! &
69) X2                                       + 10000 C68      >=    45    ! &
70) X3                                       + 10000 C68      >=    50    ! &
71) X4                                       + 10000 C68      >=    55    ! &
72) X5                                       + 10000 C68      >=    75    ! &
73) X6                                       + 10000 C68      >=    30    ! &
74) X7                                       + 10000 C68      >=    40    ! &
75) X8                                       + 10000 C68      >=    50    ! &
76) X9                                       + 10000 C68      >=    60    ! &
77) X10                                      + 10000 C68      >=    70    ! &
78) X11                                      + 10000 C68      >=    50    ! &
79) X12                                      + 10000 C68      >=    50    ! &
80) X13                                      + 10000 C68      >=    50    ! &
81) X14                                      + 10000 C68      >=    50    ! &
82) X15                                      + 10000 C68      >=    50    ! &
83) X16                                      + 10000 C68      >=    30    ! &
84) X17                                      + 10000 C68      >=    40    ! &
85) X18                                      + 10000 C68      >=    50    ! &
86) X19                                      + 10000 C68      >=    60    ! &
87) X20                                      + 10000 C68      >=    70    ! &
88) X21                                      + 10000 C68      >=    25    ! &
89) X22                                      + 10000 C68      >=    45    ! &
90) X23                                      + 10000 C68      >=    50    ! &
91) X24                                      + 10000 C68      >=    55    ! &
92) X25                                      + 10000 C68      >=    75    ! &

93) X5  + X10 + X15 + X20 + X25 - 10000 C93      <=   750    ! & CV X1
                            E1  + 10000 C93      >=    25

94) X21 + X22 + X23 + X24 + X25 - 10000 C94      <=   750    ! & CV X1
                            E1  + 10000 C94      >=    25

95) X1  + X6  + X11 + X16 + X21 + 10000 C95      >=   250    ! & CV X25
                            E25 + 10000 C95      >=    50

96) X1  + X2  + X3  + X4  + X5  + 10000 C96      >=   250    ! & CV X25
                            E25 + 10000 C96      >=    50

END
INTEGER 22
```

EXP10.S1

```
MIN   100 C38 + 100 C39 + 100 C40 + 100 C41 + 100 C42 +
      100 C43 + 100 C44 + 100 C45 + 100 C46 + 100 C47 +
      100 C48 + 100 C49 + 100 C50 + 100 C52 + 100 C56 +
      100 C60 + 100 C64 + 100 C68 + 100 C93 + 100 C94 +
      100 C95 + 100 C96

         OBJECTIVE FUNCTION VALUE

         1)      400.00000

   VARIABLE          VALUE          REDUCED COST
        C38           .000000         100.000000
        C39           .000000         100.000000
        C40           .000000         100.000000
        C41           .000000         100.000000
        C42           .000000         100.000000
        C43           .000000         100.000000
        C44           .000000         100.000000
        C45           .000000         100.000000
        C46           .000000         100.000000
        C47           .000000         100.000000
        C48          1.000000         100.000000
        C49           .000000         100.000000
        C50          1.000000         100.000000
        C52          1.000000         100.000000
        C56          1.000000         100.000000
        C60           .000000         100.000000
        C64           .000000         100.000000
        C68           .000000         100.000000
        C93           .000000         100.000000
        C94           .000000         100.000000
        C95           .000000         100.000000
        C96           .000000         100.000000
```

X1	100.000000	.000000
X2	45.000000	.000000
X3	50.000000	.000000
X4	225.000000	.000000
X5	100.000000	.000000
X6	100.000000	.000000
X7	40.000000	.000000
X8	50.000000	.000000
X9	210.000000	.000000
X10	100.000000	.000000
X11	100.000000	.000000
X12	180.000000	.000000
X13	300.000000	.000000
X14	50.000000	.000000
X15	100.000000	.000000
X16	100.000000	.000000
X17	40.000000	.000000
X18	50.000000	.000000
X19	210.000000	.000000
X20	100.000000	.000000
X21	100.000000	.000000
X22	195.000000	.000000
X23	50.000000	.000000
X24	55.000000	.000000
X25	100.000000	.000000
E1	25.000000	.000000
E25	50.000000	.000000

EXP10.S2

```
MIN     1 C38 +    1 C39 +    1 C40 +    1 C41 +    1 C42 +
        1 C43 +    1 C44 +    1 C45 +    1 C46 +    1 C47 +
      100 C48 +    1 C49 +  100 C50 +  100 C52 +  100 C56 +
        1 C60 +    1 C64 +    1 C68 +    1 C93 +    1 C94 +
        1 C95 +    1 C96
```

OBJECTIVE FUNCTION VALUE

1) 108.00000

VARIABLE	VALUE	REDUCED COST
C38	1.000000	1.000000
C39	1.000000	1.000000
C40	.000000	1.000000
C41	.000000	1.000000
C42	.000000	1.000000
C43	1.000000	1.000000
C44	1.000000	1.000000
C45	.000000	1.000000
C46	.000000	1.000000
C47	.000000	1.000000
C48	1.000000	100.000000
C49	1.000000	1.000000
C50	.000000	100.000000
C52	.000000	100.000000
C56	.000000	100.000000
C60	.000000	1.000000
C64	.000000	1.000000
C68	1.000000	1.000000
C93	.000000	1.000000
C94	.000000	1.000000
C95	1.000000	1.000000
C96	1.000000	1.000000

X1	60.000000	.000000
X2	10.000000	.000000
X3	10.000000	.000000
X4	10.000000	.000000
X5	10.000000	.000000
X6	40.000000	.000000
X7	10.000000	.000000
X8	85.000000	.000000
X9	110.000000	.000000
X10	10.000000	.000000
X11	10.000000	.000000
X12	10.000000	.000000
X13	10.000000	.000000
X14	460.000000	.000000
X15	10.000000	.000000
X16	10.000000	.000000
X17	185.000000	.000000
X18	385.000000	.000000
X19	10.000000	.000000
X20	10.000000	.000000
X21	10.000000	.000000
X22	10.000000	.000000
X23	10.000000	.000000
X24	10.000000	.000000
X25	660.000000	.000000
E1	25.000000	.000000
E25	.000000	.000000

EXP10.S3

```
MIN     1 C38 +    1 C39 +    1 C40 +    1 C41 +    1 C42 +
        1 C43 +    1 C44 +    1 C45 +    1 C46 +    1 C47 +
      100 C48 +    1 C49 +    1 C50 +    1 C52 +    1 C56 +
        1 C60 +    1 C64 +    1 C68 +    1 C93 +    1 C94 +
        1 C95 +    1 C96
```

OBJECTIVE FUNCTION VALUE

1) 13.000000

VARIABLE	VALUE	REDUCED COST
C38	1.000000	1.000000
C39	1.000000	1.000000
C40	.000000	1.000000
C41	1.000000	1.000000
C42	1.000000	1.000000
C43	1.000000	1.000000
C44	1.000000	1.000000
C45	1.000000	1.000000
C46	.000000	1.000000
C47	1.000000	1.000000
C48	.000000	100.000000
C49	1.000000	1.000000
C50	1.000000	1.000000
C52	1.000000	1.000000
C56	1.000000	1.000000
C60	.000000	1.000000
C64	.000000	1.000000
C68	1.000000	1.000000
C93	.000000	1.000000
C94	.000000	1.000000
C95	.000000	1.000000
C96	.000000	1.000000

X1	60.000000	.000000
X2	60.000000	.000000
X3	60.000000	.000000
X4	60.000000	.000000
X5	60.000000	.000000
X6	60.000000	.000000
X7	10.000000	.000000
X8	10.000000	.000000
X9	10.000000	.000000
X10	10.000000	.000000
X11	60.000000	.000000
X12	10.000000	.000000
X13	10.000000	.000000
X14	410.000000	.000000
X15	10.000000	.000000
X16	35.000000	.000000
X17	10.000000	.000000
X18	10.000000	.000000
X19	10.000000	.000000
X20	35.000000	.000000
X21	35.000000	.000000
X22	10.000000	.000000
X23	10.000000	.000000
X24	10.000000	.000000
X25	35.000000	.000000
E1	25.000000	.000000
E25	50.000000	.000000

EXP11.LP

```
 1)        X1                    >=     5    ! F
 2)        X1                    <=    15    ! F
 3)                        X2    >=     5    ! F
 4)     - X1   +           X2    <=    10    ! F
 5)        X1   +           X2    <=    30    ! & CV X1
 6)                        X2    <=    10    ! & CV X2
 7)        X1   +           X2    >=    30    ! & CV X1
```

EXP11.PMIP/B

```
MIN  100 C5  + 100 C6  + 100 C7
ST
      X1                                                  >=     5    ! F
      X1                 +    E1                         <=    15    ! F
                  X2                                     >=     5    ! F
    - X1  +       X2                +    E2              <=    10    ! F

      X1  +       X2         +    E1          - 10000 C5  <=    30    ! & CV X1
                             E1              + 10000 C5  >=     2

                  X2                +    E2    - 10000 C6  <=    10    ! & CV X2
                                     E2      + 10000 C6  >=     2

      X1  +       X2                         + 10000 C7  >=    30    ! & CV X1
                             E1              + 10000 C7  >=     2
END
INTEGER 3
```

EXP11.S1

```
MIN   100 C5   + 100 C6   + 100 C7

         OBJECTIVE FUNCTION VALUE

         1)      100.00000

  VARIABLE         VALUE          REDUCED COST
        C5          .000000        100.000000
        C6          .000000        100.000000
        C7         1.000000        100.000000

        X1        13.000000           .000000
        E1         2.000000           .000000
        X2         5.000000           .000000
        E2         5.000000           .000000
```

EXP11.S2

```
MIN     1 C5   +    1 C6   + 100 C7

         OBJECTIVE FUNCTION VALUE

         1)      2.0000000

  VARIABLE         VALUE          REDUCED COST
        C5         1.000000          1.000000
        C6         1.000000          1.000000
        C7          .000000        100.000000

        X1        10.000000           .000000
        E1         2.000000           .000000
        X2        20.000000           .000000
        E2          .000000           .000000
```

EXP12.LP

```
 1)       X1                 >=     5    ! F
 2)       X1                 <=    15    ! F
 3)                    X2    >=     5    ! F
 4)     - X1   +       X2    <=    10    ! F
 5)       X1   +       X2    <=    30    ! & CV X1
 6)                    X2    <=    10    ! & CV X2
 7)       X1   +       X2    >=    30    ! & CV X1
 8)       X1   +       X2    <=    15
 9)                    X2    >=    22
```

EXP12.PMIP/B

```
MIN  100 C5  + 100 C6  + 100 C7  + 100 C8  + 100 C9
ST
     X1                                                >=     5    ! F
     X1                 +    E1                        <=    15    ! F
                X2                                     >=     5    ! F
   - X1 +       X2                 +    E2             <=    10    ! F

     X1 +       X2      +    E1           - 10000 C5   <=    30    ! & CV X1
                             E1           + 10000 C5   >=     2

                X2                 +    E2   - 10000 C6   <=    10    ! & CV X2
                                        E2   + 10000 C6   >=     2

     X1 +       X2                       + 10000 C7   >=    30    ! & CV X1
                             E1           + 10000 C7   >=     2

     X1 +       X2      +    E1 +    E2   - 10000 C8   <=    15
                X2                       + 10000 C9   >=    22

END
INTEGER 5
```

EXP12.S1

```
MIN  100 C5  + 100 C6  + 100 C7  + 100 C8  + 100 C9

        OBJECTIVE FUNCTION VALUE

        1)      200.00000

  VARIABLE         VALUE          REDUCED COST
        C5          .000000        100.000000
        C6          .000000        100.000000
        C7         1.000000        100.000000
        C8          .000000        100.000000
        C9         1.000000        100.000000

        X1         6.000000           .000000
        E1         2.000000           .000000
        X2         5.000000           .000000
        E2         2.000000           .000000
```

EXP12.S2

```
MIN    1 C5  +   1 C6  + 100 C7  +   1 C8  + 100 C9

        OBJECTIVE FUNCTION VALUE

        1)      3.0000000

  VARIABLE         VALUE          REDUCED COST
        C5         1.000000           1.000000
        C6         1.000000           1.000000
        C7          .000000         100.000000
        C8         1.000000           1.000000
        C9          .000000         100.000000

        X1        12.000000           .000000
        E1         2.000000           .000000
        X2        22.000000           .000000
        E2          .000000           .000000
```

EXP13.LP

```
 1)       X1                    >=     5    ! F
 2)       X1                    <=    15    ! F
 3)                       X2    >=     5    ! F
 4)     - X1    +         X2    <=    10    ! F
 5)       X1    +         X2    <=    30    ! & CV X1
 6)                       X2    <=    10    ! & CV X2
 7)       X1    +         X2    <=    25    ! & CV X1
 8)                       X2     =    18
```

EXP13.PMIP/B

```
MIN  100 C5  + 100 C6  + 100 C7  + 100 C8
ST
     X1                                                    >=     5    ! F
     X1                  +    E1                           <=    15    ! F
               X2                                          >=     5    ! F
   - X1  +     X2                  +    E2                 <=    10    ! F

     X1  +     X2        +    E1              - 10000 C5   <=    30    ! & CV X1
                         E1                   + 10000 C5   >=     2

               X2                  +    E2     - 10000 C6   <=    10    ! & CV X2
                                    E2     + 10000 C6   >=     2

     X1  +     X2        +    E1              - 10000 C7   <=    25    ! & CV X1
                         E1                   + 10000 C7   >=     2

               X2                  +    E2     - 10000 C8   <=    18    ! &
               X2                                + 10000 C8   >=    18
END
INTEGER 4
```

EXP13.S1

```
MIN  100 C5  + 100 C6  + 100 C7  + 100 C8

        OBJECTIVE FUNCTION VALUE

        1)     100.00000

  VARIABLE         VALUE          REDUCED COST
        C5           .000000        100.000000
        C6           .000000        100.000000
        C7           .000000        100.000000
        C8          1.000000        100.000000

        X1          5.000000           .000000
        E1          2.000000           .000000
        X2          8.000000           .000000
        E2          2.000000           .000000
```

EXP13.S2

```
MIN    1 C5  +   1 C6  +   1 C7  + 100 C8

        OBJECTIVE FUNCTION VALUE

        1)     2.0000000

  VARIABLE         VALUE          REDUCED COST
        C5           .000000          1.000000
        C6          1.000000          1.000000
        C7          1.000000          1.000000
        C8           .000000        100.000000

        X1         10.000000           .000000
        E1          2.000000           .000000
        X2         18.000000           .000000
        E2           .000000           .000000
```

EXP14.LP

```
 1)       X1                    >=     5    ! F
 2)       X1                    <=    15    ! F
 3)                  X2         >=     5    ! F
 4)     - X1   +     X2         <=    10    ! F
 5)       X1   + 2 X2           <=    30    ! &
 6)     2 X1   +     X2         <=    25    ! & CV X1
 7)       X1   +     X2         >=    30
 8)       X1   +     X2         <=    25
```

EXP14.PMIP/B

```
MIN  100 C5  + 100 C7  + 100 C8
ST
     X1                                          >=     5    ! F
     X1              +    E1                     <=    15    ! F
               X2                                >=     5    ! F
   - X1 +    X2                                  <=    10    ! F

     X1 + 2 X2     +    E1          - 10000 C5   <=    30    ! &
   2 X1 +   X2     + 2  E1          - 10000 C5   <=    25    ! & CV X1
                        E1          + 10000 C5   >=     2

     X1 +   X2                     + 10000 C7   >=    30
     X1 +   X2       +   E1        - 10000 C8   <=    25
END
INTEGER 3
```

EXP14.S1

```
MIN  100 C5  + 100 C7  + 100 C8

        OBJECTIVE FUNCTION VALUE

        1)      100.00000

  VARIABLE          VALUE          REDUCED COST
        C5           .000000        100.000000
        C7          1.000000        100.000000
        C8           .000000        100.000000

        X1          8.000000           .000000
        E1          2.000000           .000000
        X2          5.000000           .000000
```

EXP14.S2

```
MIN    1 C5  + 100 C7  +   1 C8

        OBJECTIVE FUNCTION VALUE

        1)      2.0000000

  VARIABLE          VALUE          REDUCED COST
        C5          1.000000          1.000000
        C7           .000000          1.000000
        C8          1.000000        100.000000

        X1         15.000000           .000000
        E1           .000000           .000000
        X2         25.000000           .000000
```

EXP15.LP

```
 1)     2 X1                    <=    41    ! F
 2)                 3 X2        <=    65    ! F
 3)     4 X1   +    3 X2        <=    12
 4)  -  2 X1   +      X2        >=     2
 5)  -    X1   +      X2        >=     6
 6)    16 X1   +   17 X2        >=   272
 7)     2 X1   -    7 X2        >=    14
 8)     2 X1   -    5 X2        <=    16    ! & CV X1
 9)     9 X1                    <=    46    ! & CV X2
10)     5 X1   +    8 X2        <=    45    ! & CV X1
```

EXP15.PMIP/A

```
MIN   100 C3   + 100 C4   + 100 C5   + 100 C6   +
      100 C7   + 100 C8   + 100 C9   + 100 C10
ST
    2 X1            +     W1                        <=    41    ! F
                          W1             + 10000 C8 >=     2
                          W1             + 10000 C10>=     2

              3 X2  +     W2                        <=    65    ! F
                          W2             + 10000 C9 >=     3

    4 X1  +   3 X2  +     W3             - 10000 C3 <=    12
                          W3             + 10000 C8 >=     4
                          W3             + 10000 C10>=     4
                          W3             + 10000 C9 >=     3

-   2 X1  +     X2  -     W4             + 10000 C4 >=     2
                          W4             + 10000 C8 >=     2
                          W4             + 10000 C10>=     2

-     X1  +     X2  -     W5             + 10000 C5 >=     6
                          W5             + 10000 C8 >=     1
                          W5             + 10000 C10>=     1

   16 X1  + 17 X2                        + 10000 C6 >=   272

    2 X1  -   7 X2  -     W7             + 10000 C7 >=    14
                          W7             + 10000 C9 >=     7

    2 X1  -   5 X2  +   2 W8             - 10000 C8 <=    16    ! & CV X1
                          W8             + 10000 C8 >=     2

    9 X1                                  - 10000 C9 <=    46    ! & CV X2

    5 X1  +   8 X2  +     W10            - 10000 C10<=    45    ! & CV X1
                          W10            + 10000 C10>=     5

END
INTEGER 8
```

EXP15.S1/A

```
MIN   100 C3   + 100 C4   + 100 C5   + 100 C6   +
      100 C7   + 100 C8   + 100 C9   + 100 C10
```

OBJECTIVE FUNCTION VALUE

1) 300.00000

VARIABLE	VALUE	REDUCED COST
C3	1.000000	100.000000
C4	.000000	100.000000
C5	.000000	100.000000
C6	.000000	100.000000
C7.	1.000000	100.000000
C8	.000000	100.000000
C9	.000000	100.000000
C10	1.000000	100.000000
X1	5.111111	.000000
X2	20.666670	.000000
W1	2.000000	.000000
W2	3.000000	.000000
W3	4.000000	.000000
W4	2.000000	.000000
W5	1.000000	.000000
W7	7.000000	.000000
W8	2.000000	.000000
W10	9854.111000	.000000

EXP15.S2/A

```
MIN   100 C3  +    1 C4  +    1 C5  +    1 C6  +
      100 C7  +    1 C8  +    1 C9  + 100 C10
```

OBJECTIVE FUNCTION VALUE

1) 103.00000

VARIABLE	VALUE	REDUCED COST
C3	.000000	100.000000
C4	1.000000	1.000000
C5	1.000000	1.000000
C6	1.000000	1.000000
C7.	1.000000	100.000000
C8	.000000	1.000000
C9	.000000	1.000000
C10	.000000	100.000000
X1	2.000000	.000000
X2	.000000	.000000
W1	2.000000	.000000
W2	3.000000	.000000
W3	4.000000	.000000
W4	2.000000	.000000
W5	1.000000	.000000
W7	7.000000	.000000
W8	2.000000	.000000
W10	35.000000	.000000

EXP15.S3/A

```
MIN     1 C3   +    1 C4   +    1 C5   +    1 C6   +
      100 C7   +    1 C8   +    1 C9   +    1 C10
```

OBJECTIVE FUNCTION VALUE

1) 5.0000000

VARIABLE	VALUE	REDUCED COST
C3	1.000000	1.000000
C4	1.000000	1.000000
C5	1.000000	1.000000
C6	1.000000	1.000000
C7.	.000000	100.000000
C8	.000000	1.000000
C9	1.000000	1.000000
C10	.000000	1.000000
X1	7.000000	.000000
X2	.000000	.000000
W1	2.000000	.000000
W2	.000000	.000000
W3	4.000000	.000000
W4	2.000000	.000000
W5	1.000000	.000000
W7	.000000	.000000
W8	2.000000	.000000
W10	10.000000	.000000

EXP15.LP

```
 1)     2 X1                    <=    41    ! F
 2)                 3 X2        <=    65    ! F
 3)     4 X1   +    3 X2        <=    12
 4)  -  2 X1   +      X2        >=     2
 5)  -    X1   +      X2        >=     6
 6)    16 X1   +   17 X2        >=   272
 7)     2 X1   -    7 X2        >=    14
 8)     2 X1   -    5 X2        <=    16    ! & CV X1
 9)     9 X1                    <=    46    ! & CV X2
10)     5 X1   +    8 X2        <=    45    ! & CV X1
```

EXP15.PMIP/B

```
MIN   100 C3   + 100 C4   + 100 C5   + 100 C6   +
      100 C7   + 100 C8   + 100 C9   + 100 C10
ST
    2 X1              +  2 E1                                  <=    41    ! F
                3 X2              +  3 E2                      <=    65    ! F

    4 X1 +   3 X2 +   4 E1 +   3 E2   - 10000 C3   <=    12
-   2 X1 +     X2 -   2 E1           + 10000 C4   >=     2
-     X1 +     X2 -     E1           + 10000 C5   >=     6
   16 X1 + 17 X2                     + 10000 C6   >=   272
    2 X1 -   7 X2             -  7 E2 + 10000 C7   >=    14

    2X1   -   5 X2 +   2 E1           - 10000 C8   <=    16    ! & CV X1
                       E1           + 10000 C8   >=     1

    9X1                             - 10000 C9   <=    46    ! & CV X2
                                 E2 + 10000 C9   >=     1

    5X1   +   8 X2 +   5 E1           - 10000 C10  <=    45    ! & CV X1
                       E1           + 10000 C10  >=     1

END
INTEGER 8
```

EXP15.S1/B

```
MIN   100 C3   + 100 C4   + 100 C5   + 100 C6. +
      100 C7   + 100 C8   + 100 C9   + 100 C10

         OBJECTIVE FUNCTION VALUE

         1)      300.00000

  VARIABLE          VALUE          REDUCED COST
        C3          1.000000         100.000000
        C4           .000000         100.000000
        C5           .000000         100.000000
        C6           .000000         100.000000
        C7.         1.000000         100.000000
        C8           .000000         100.000000
        C9           .000000         100.000000
       C10          1.000000         100.000000

        X1          5.111111            .000000
        E1          4.222222            .000000
        X2         20.666670            .000000
        E2          1.000000            .000000
```

EXP15.S2/B

```
MIN   100 C3   +   1 C4   +   1 C5   +   1 C6   +
      100 C7   +   1 C8   +   1 C9   + 100 C10

         OBJECTIVE FUNCTION VALUE

         1)      103.00000

  VARIABLE          VALUE          REDUCED COST
        C3           .000000         100.000000
        C4          1.000000           1.000000
        C5          1.000000           1.000000
        C6          1.000000           1.000000
        C7.         1.000000         100.000000
        C8           .000000           1.000000
        C9           .000000           1.000000
       C10           .000000         100.000000

        X1          1.250000            .000000
        E1          1.000000            .000000
        X2           .000000            .000000
        E2          1.000000            .000000
```

EXP15.S3/B

```
MIN     1 C3  +    1 C4  +    1 C5  +    1 C6  +
      100 C7  +    1 C8  +    1 C9  +    1 C10
```

OBJECTIVE FUNCTION VALUE

1) 5.0000000

VARIABLE	VALUE	REDUCED COST
C3	1.000000	1.000000
C4	1.000000	1.000000
C5	1.000000	1.000000
C6	1.000000	1.000000
C7.	.000000	100.000000
C8	.000000	1.000000
C9	1.000000	1.000000
C10	.000000	1.000000
X1	7.000000	.000000
E1	1.000000	.000000
X2	.000000	.000000
E2	.000000	.000000

Bibliography

[1] Abelson, R. P. (1968), "Simulation of Social Behaviour," The Handbook of Social Psychology, Vol. 2: Research Methods, G. Lindzey and E. Aronson (Eds.), Addison-Wesley.

[2] Asher, H. B. (1983), Causal Modeling, Beverly Hills, CA: Sage Publications.

[3] Balci, O. (1988), "Credibility Assessment of Simulation Results: The State of the Art," Methodology and Validation, O. Balci (Ed.), SCS, Vol. 19, No. 1, San Diego, pp. 19-25.

[4] Balci, O. (1990), "Guidelines for Successful Simulation Studies," in: Proc. of 1990 Winter Simulation Conference, O. Balci, R. P. Sadowski and R. E. Nance (Eds.), New Orleans, Louisiana, pp. 25-32.

[5] Balci, O. and R. G. Sargent (1984), "A Bibliography on the Credibility, Assessment and Validation of Simulation and Mathematical Models," Simuletter, Vol. 15, No. 3, pp. 15-27.

[6] Banks, J., D. Gerstein and S. P. Searles (1988), "Modeling Processes, Validation, and Verification of Complex Simulations: A Survey," Methodology and Validation, O. Balci (Ed.), SCS, Vol. 19, No. 1, San Diego, pp. 13-18.

[7] Banks, J., D. Gerstein and S. P. Searles (1990), "Verification and Validation of Large Scale Simulation Models," Proc. UKSC Conference on Computer Simulation, Brighton, England.

[8] Barlas, Y. (1990), "An Autocorrelation Function Test for Output Validation," Simulation, Vol. 55, No. 1, pp. 7-16.

[9] Birta, L. G., O. Abou-Rabia, T. I. Ören, D. King and R. Wendt (1991), "Reverse Engineering in the Simulation Life Cycle," Systems Analysis Modeling Simulation, Vol. 9, No. 1, pp. 69-84.

[10] Brunessaux, L., J. P. Vaudet, S. Petitjean and M. N. Jullion (1990), "An Attempt to Improve the Knowledge-Based Systems Validation: The VALID Project," The Journal of Knowledge Engineering, Vol. 3, No. 3, pp. 1-8.

[11] CCITT (1983), General Network Planning, The International Telegraph and Telephone Consultative Committee, Geneva.

[12] Chinneck, J. W. Personal Communication.

[13] Chinneck, J. W. and E. W. Dravnieks (1991), "Locating Minimal Infeasible Constraint Sets in Linear Programs," ORSA Journal of Computing, Vol. 3, No. 2, pp. 157-168.

[14] Chvatal, V. (1979), "A Greedy Heuristic for the Set-Covering Problem," Mathematics of Operations Research, Vol. 4, No. 3, August, pp. 233-235.

[15] Clarke, L. A. (1976), "A System to Generate Test Data and Symbolically Execute Programs," IEEE Trans. Software Engineering, Vol. SE-2, No. 3, September, pp. 215-222.

[16] Colby, K. M. (1973), "Simulations of Belief Systems," Computer Models of Thought and Language, R. C. Shank and K. M. Colby (Eds.), Freeman.

[17] COMNET II.5 (1991), User's Manual, Release 4, CACI Products Company, La Jolla,CA.

[18] Cormen, T. H., C. E. Leiserson and R. L. Rivest (1989), Introduction to Algorithms, The MIT Press, Cambridge, Massachusetts and McGraw-Hill Book Company, New York.

[19] Cornuejols, G., M. L. Fisher and G. L. Nemhauser (1977), "Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms," Management Science, vol. 23, no. 8, April, pp. 789-810.

[20] Czaja, L. (1988), "Cause-Effect Structures," Information Processing Letters, Vol. 26, pp. 313-319.

[21] Davis, R. (1984), "Diagnostic Reasoning Based on Structure and Behaviour," Artificial Intelligence, Vol. 24, No. 3, pp. 347-410.

[22] Deason, W. H., D. B. Brown, K. Chang and J. H. CrossII (1991), "A Rule-Based Software Test Data Generator," IEEE Trans. on Knowledge and Data Engineering, Vol. 3, No. 1, pp. 108-117.

[23] deKleer, J. and J. S. Brown (1984), "A Qualitative Physics Based on Confluences," Artificial Intelligence, vol. 24, pp. 7-83.

[24] DeMillo, R. A., R. J. Lipton and F. G. Sayward (1978), "Hints on Test Data Selection: Help for the Practicing Programmer," IEEE Computer Magazine, Vol. 11, No. 4, pp. 34-41.

[25] DeMillo, R. A. and A. J. Offutt (1991), "Constraint-Based Automatic Test Data Generation," IEEE Trans. Software Engineering, Vol. SE-17, No. 9, Setpember, pp. 900-910.

[26] Deng, D. and J. O. Jenkins (1989), "Artificial Intelligence Validation of Simulation Model," Advances in AI and Simulation, pp. 80-84.

[27] Deslandres, V. and H. Pierreval (1991), "An Expert System Prototype Assisting the Statistical Validation of Simulation Models," Simulation, Vol. 56, No. 2, pp. 79-89.

[28] Dijkstra, E. W. (1959), "A Note on Two Problems in Connexion with Graphs," Numerische Mathematik, Vol. 1, pp. 269-271.

[29] Dobson, G. (1982), "Worst-Case Analysis of Greedy Heuristics for Integer Programming with Nonnegative Data," Mathematics of Operations Research, Vol. 7, No. 4, November, pp. 515-531.

[30] Edmonds, J. (1971), "Matroids and the Greedy Algorithm," Mathematical Programming, Vol. 1, pp. 127-136.

[31] Elzas, M. S., T. I. Ören and B. P. Zeigler (1986), Modelling and Simulation Methodology in the Artificial Intelligence Era, North-Holland, Amsterdam.

[32] Fidelak, M. (1989), "Verification of Production Systems," Computing and Information, R. Janicki and W. W. Koczkodaj(Eds.), Elsivier Science, North-Holland, pp. 371-378.

[33] Findler, N. V. and N. M. Mazur (1990), "A System for Automatic Model Verification and Validation," Transactions of the Society for Computer Simulation, Vol. 6, No. 3, pp. 153-172.

[34] Forbus, K. D. (1984),"Qualitative Process Theory," Artificial Intelligence, vol. 24, pp. 85-168.

[35] Forrester, J. W. (1968), Principles of Systems, Cambridge, MA: Wright-Allen.

[36] Friedman, L. W. and H. H. Friedman (1985), "Validating the Simulation Metamodel: Some Practical Approaches," Simulation, Vol. 45, No. 3, pp. 144-146.

[37] Futo, I. and T. Gergely (1987), "Logic Programming in Simulation," Trans. of the Society for Computer Simulation, pp. 195-216.

[38] Garey, M. R. and D. S. Johnson (1979), Computers and Intractibility - A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, San Francisco.

[39] Garey, M. R. and D. S. Johnson (1976), "Approximation Algorithms for Combinatorial Problems: an Annotated Bibliography," in J.F.Traub (ed.), Algorithms and Complexity: New Directions and Recent Results, Academic Press, New York, pp. 41-52.

[40] Garzia, R. F. and M. R. Garzia (Eds) (1990), Network Modeling, Simulation and Analysis, Marcel Dekker.

[41] Genrich, H. J. (1987), "Predicate/Transition Nets," Lecture Notes on Computer Science, No. 254, Springer-Verlag, NY, pp. 207-247.

[42] Ginsberg, A., S. M. Weiss and P. Politakis (1988), "Automatic Knowledge Base Refinement for Classification Systems," Artificial Intelligence, Vol. 35, pp. 197-226.

[43] Giordana, A. and L. Saitta (1985), "Modeling Production Rules by Means of Predicate Transition Networks," Information Science, Vol. 35, pp. 1-41.

[44] Gonzales, A. J. and D. D. Dankel (1993), The Engineering of Knowledge-based Systems: Theory and Practice, Prentice Hall, Englewood Cliffs, NJ.

[45] Goodenough, J. B. and S. L. Gerhart (1975), "Toward a Theory of Test Data Selection," IEEE Trans. on Software Engineering, Vol. SE-1, No. 2, pp. 156-173.

[46] Hebuterne, G. (1987), Traffic Flow in Switching Systems, Artech House Inc.

[47] Herskovitz, P. J. (1991), "A Theoretical Framework for Simulation Validation: Popper's Falsification," Int. J. of Modelling and Simulation, Vol. 11, No. 2, pp. 56-58.

[48] Horowitz, E. and S. Sahni (1978), Fundamentals of Computer Algorithms, Computer Science Press Inc., Maryland.

[49] Howden, W. E. (1986), "A Functional Approach to Program Testing and Analysis," IEEE Trans. on Software Engineering, Vol. SE-12, No. 10, pp. 997-1005.

[50] Howden, W. E. (1987), Functional Program Testing and Analysis, New York, McGraw-Hill.

[51] Hudlická, E. (1988), "Construction and Use of a Causal Model for Diagnosis," International Journal of Intelligent Systems, Vol. 3, pp. 315-349.

[52] Ibarra, O. H. and C. E. Kim (1975), "Fast Approximation Algorithms for the Knapsack and Sum of Subset Problems," J. ACM, Vol. 22, No. 4, October, pp. 463-468.

[53] Johnson, D. S. (1974), "Approximation Algorithms for Combinatorial Problems," J. Computer and System Science, Vol. 9, pp. 256-278.

[54] Kelton, W. D. (1988), "Designing Computer Simulation Experiments," Proc. WSC, M. Abrams, P. Haigh and J. Comfort (Eds.), pp. 15-18.

[55] Klahr, P. and W. S. Fought (1980), "Knowledge-Based Simulation," Proc. First Annual Conference of the AAAI, Stanford, pp. 181-183.

[56] Klir, G. J. (1991), Facets of Systems Science, Plenum Press, New York.

[57] Korte, B. (1979), "Approximative Algorithms for Discrete Optimization Problems," Annals of Discrete Mathematics, Vol. 4, pp. 85-120.

[58] Kruskal, J. B. Jr. (1958), "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," Proc. Americal Mathematical Society, Vol. 7, No. 1, pp. 48-50.

[59] Kuipers, B. J. (1987), "Qualitative Simulation as Casual Explanation," IEEE Tr. Systems, Man and Cybernetics, Vol. SMC-17, No. 13, pp. 432-444.

[60] Kuipers, B. J. and D. Berleant (1988), "Using Incomplete Quantitative Knowledge in Qualitative Reasoning," Proc. of the seventh National Conf. on Artificial Intelligence, AAAI-88, Vol. 1, pp. 324-329.

[61] Law, A. M. and C. S. Larmey (1984), An Introduction to Simulation using SIMSCRIPTII.5, Los Angeles, CA.

[62] Lendaris, G. G. (1980), "Structural Modeling - a Tutorial guide," IEEE Tr. Systems, Man and Cybernetics, Vol. SMC-10, No. 12, pp. 807-840.

[63] Lovasz, L. (1975), "On the Ratio of Optimal Integral and Fractional Covers," Discrete Mathematics, Vol. 13, pp. 383-390.

[64] Maes, R. and J. E. M. Van Dijk (1988), "On the Role of Ambiguity and Incompleteness in the Design of Decision Tables and Rule-Based Systems," The Computer Journal, Vol. 31, No. 6, pp. 481-489.

[65] Marathe, H., T-K. Ma and C-C. Liu (1989), "An Algorithm for Identification of Relations Among Rules," Proc. IEEE Int. Workshop on Tools for AI, October 23-25, pp. 360-367.

[66] McGuire, J. G. (1990), "Uncovering Redundancy and Rule-Inconsistency in Knowledge Bases via Deduction," Proc. IEEE conference, pp. 57-67.

[67] Miller, E. F. and W. E. Howden (1979), Software Testing and Validation Techniques, IEEE Computer Society.

[68] Miller, E. F. Jr. (1984), "Software Technology: An Overview," in: Handbook of Software Engineering, C. V. Rick and C. V. Ramamoorthy (Eds.), Van Nostrand Reinhold Company Inc. New York, pp. 359-379.

[69] Misra, J. (1986), "Distributed Discrete-Event Simulation," Computing Survey, Vol. 18, No. 1, pp. 39-65 .

[70] Montan, V. and R. V. Reddy (1989), "An Expert Systems Approach to the Analysis of Discrete Event Simulations," Advances in AI and Simulation, pp. 189-193.

[71] Murray, K. J. and S. V. Sheppard (1988), "Knowledge Based Simulation Model Specification," Simulation, Vol. 50, No. 3, pp. 112-119.

[72] Myers, G. J. (1976), Software Reliability: Principles and Practices, New York, Wiley-Interscience.

[73] Myers, G. J. (1979), The Art of Software Testing, John Wiley and Sons.

[74] Nance, R. E. and C. M. Overstreet (1988), "Diagnostic Assistance Using Digraph Representations of Discrete Event Simulation Model Specifications," Transactions of the SCS, Vol. 4, No. 1, pp. 33-57.

[75] Naylor, T. H. and J. M. Finger (1967), "Verification of Computer Simulation Models," Management Science, Vol. 14, No. 2, pp. B92-B101.

[76] Nguyen, T. A. et. al. (1985), "Checking an Expert Systems Knowledge Base for Consistency and Completeness," IJCAI-85, pp. 375-378.

[77] Offut, A. J. and E. J. Seaman (1990), "Using Symbolic Execution to Aid Automatic Test Data Generation," Proc. IEEE Conf., pp. 12-21.

[78] Ostrand, T. J. and M. J. Balcer (1988), "The Category-Partition Method for Specifying and Generating Functional Tests," Communications of the ACM, Vol. 31, No. 6, pp. 676-686.

[79] Ören, T. I. (1984), "GEST - A Modelling and Simulation Language Based on System Theoretic Concepts," Simulation and Model-Based Methodologies: An Integrative View, T. I. Ören, B. P. Zeigler and M. S. Elsas (Eds.), Springer-Verlag, New York, pp. 281-335.

[80] Ören, T. I. (1987), "Quality Assurance Paradigms for Artificial Intelligence in Modelling and Simulation," Simulation, Vol. 48, No. 4, pp. 149-151.

[81] Ören, T. I. (1989), "Bases for Advanced Simulation: Paradigms for the Future," Modelling and Simulation Methodology: Knowledge Systems' Paradigms, M. S. Elzas, T. I. Ören and B. P. Zeigler (eds.), North-Holland, Amsterdam, pp. 29-43.

[82] Ören, T. I. (1992), "Advances in Knowledge-Based Simulation Systems," in: Proc. of the Symp. on Advances in Simulation '92, A. R. Kaylan and T. I. Ören (eds.), Istanbul, Turkey, pp. 17-24.

[83] Ören, T. I. (1992), "Simulation Environments: Challenges for Advancement," in: Proc. 2nd Int. Conf. System Simulation and Scientific Computing, Beijing, China, pp. 8-12.

[84] Ören, T. I. (1993), "Three Simulation Experimentation Environments: SIMAD, SIMGEST, and E/SLAM," in: Proc. European Simulation Symp., Delft, Netherlands, A. Verbraeck and E. J. H. Kerckhoffs (Eds.), SCSI, San Diago, California, pp. 627-632.

[85] Ören, T. I. (1994), "Artificial Intelligence in Simulation," To appear in Annals of Operations Research.

[86] Ören, T. I., M. S. Elzas and G. Sheng(1985), "Model Reliability and Software Quality Assurance in Simulation of Nuclear Waste Management Systems," Waste Management, R. G. Post (Ed.), Vol. 1, pp. 381-396.

[87] Ören, T. I. and B. P. Zeigler (1979), "Concepts for Advanced Simulation Methodologies," Simulation, Vol. 32, No. 3, pp. 69-82.

[88] Papdimitriou, C. H. and K. Steiglitz (1982), Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall.

[89] Peterson, J. L. (1981), Petri Net Theory and the Modeling of Systems, Prentice-Hall, Englewood Cliffs, N.J.

[90] Pierreval, H. (1992), "Rule-based Simulation Metamodels," European Journal of Operational Research, Vol. 61, pp. 6-17.

[91] Prim, R. C. (1957), "Shortest Connection Networks and Some Generalizations," Bell System Technical Journal, Vol. 36, pp. 1389-1401.

[92] Ramamoorthy, C. V., S-B. F. Ho. and W. T. Chen (1976), "On the Automated Generation of Program Test Data ," IEEE Trans. Software Engineering, Vol. SE-2, No. 4, December, pp. 293-300.

[93] Rao, M. J. and R. G. Sargent (1988), "An Experimental Advisory System for Operational Validity," AI and Simulation, pp. 245-250.

[94] Reddy, Y. V., M. S. Fox and N. Husain (1988), Knowledge Based Simulation: An Artificial Intelligence Approach to System Modeling and Automating the Simulation Life Cycle, Technical Report, CM-RI-TR-88-5, Intelligent Systems

Laboratory, The Robotics Institute, Carnegie-Mellon University, Pittsburhg, Pennsylvania.

[95] Reisig, W. (1985), Petri Nets: An Introduction, EATCS Monographs on Theoretical Computer Science, Vol. 4, Springer-Verlag, NY.

[96] Rieger, C. and M. Grinberg (1978), "A System of Cause-Effect Representation and Simulation for Computer Aided-Design," Artificial Intelligence and Pattern Recognition in Computer Aided Design, Latombe (Ed.), IFIP, North-Holland Publishing Company, pp. 299-333.

[97] Rogeberg, T. (1982), "TETRASIM: A Program System for the Simulation of Telephone Networks," Computer Networks and Simulation II, S. Schoemaker (Ed.), North Holland Publishing Company.

[98] Rosenkrantz, D. J., R. E. Stearns and P. M. Lewis II (1977), "An Analysis of Several Heuristics for the Traveling Salesman Problem," SIAM J. Comput., Vol. 6, No. 3, September, pp. 563-581.

[99] Sahni, S. and T. Gonzalez (1976), "P-Complete Approximation Problems," J. ACM, Vol. 23, No. 3, July, pp. 555-565.

[100] Sahni, S. (1977), "General Techniques for Combinatorial Approximation," Operations Research, Vol. 25, No. 6, November-December, pp. 920-936.

[101] Sahni, S. and E. Horowitz (1978), "Combinatorial Problems: Reducibility and Approximation," Operations Research, Vol. 26, No. 5, September-October, pp. 718-759.

[102] Saradhi, M. (1992), 'System Modeling and Description," ACM SIGSOFT Software Engineering Notes, Vol. 17, No.2, April, pp. 57-63.

[103] Sargent, R. G. (1984), "Simulation and Model Validation," in: Simulation and Model-Based Methodologies: An Integrative View, T. I. Ören, B. P. Zeigler and M. S. Elzas (Eds.), New York: Springer-Verlag, pp. 537-555.

[104] Sargent, R. G. (1984), "A Tutorial on Verification and Validation of Simulation Models," in: Proc. of 1984 Winter Simulation Conference, S. Sheppard, U. Pooch, and D. Pegden (Eds.), Dallas, Texas, pp. 115-121.

[105] Sargent, R. G. (1986), "An Exploration of the Possibilities for Expert Aids in Model Validation," Modeling and Simulation Methodology in Artificial Intelligence Era, M. S. Elzas, T. I. Ören and B. P. Zeigler (Eds.), North-Holland, pp. 279-297.

[106] Sargent, R. G. (1987), "An Overview of Verification and Validation of Simulation Models," Proc. 1987 WSC, A. Thesen, H. Grant and W. D. Kelton (Eds.), pp. 33-39.

[107] Schrage, L. (1984), Linear, Integer, and, Quadratic Programming with LINDO, Palo Alto, CA: Scientific.

[108] Schruben, L. W. (1990), "Establishing the Credibility of Simulations," Simulation, Vol. 34, No. 3, pp. 101-105.

[109] Shannon, R. E. (1975), Systems Simulation: The Art and the Science, Prentice Hall.

[110] Sharda, R. (1992), "Linear Programming Software for Personal Computers: 1992 Survey," OR/MS Today, June, pp. 44-60.

[111] Sheng, G., M. S. Elzas, T. I. Ören and B. T. Cronhjort (1993), "Model Validation: A Systemic and Systematic Approach," To Appear in Reliability Engineering and System Safety Journal.

[112] Simon, H. A. (1977), "On the Definition of the Causal Relation," Models and Discovery, H.A.Simon, Boston, MA: Reidel, pp. 86-91.

[113] Suwa, M., A. C. Scott and E. H. Shortliffe (1984), "Completeness and Consistency in a Rule-Based Expert System," Rule-based Expert Systems, Reading, MA: Addison-Wesley, pp. 159-170.

[114] Turing, A. M. (1950), "Computing Machinery and Intelligence," Mind, Vol. 59, October, pp. 433-460.

[115] Umeda, T. et al. (1988), "A Graphical Approach to Cause and Effect Analysis of Chemical Processing Systems," Chemical Engineering Science, Vol. 35, pp. 2379-2388.

[116] Wein, A. S. and A. Sathaye (1990), "Validating Complex Computer System Availability Models," IEEE Transactions on Reliability, Vol. 39, No. 4, pp. 468-479.

[117] Weiss, S. M. et al. (1978), "A Model-Based Method for Computer Aided Medical Decision Making," Artificial Intelligence, vol. 11, pp. 145-172.

[118] White, M. (1990), "Knowledge Representation - Petri Nets," Knowledgebase, IAKE, Vol. 3, No. 8, September/October, p. 5.

[119] Wymore, A. W. (1993), Model-based Systems Engineering, CRC Press, Ann Arbor.

[120] Youngblood, S. M. (1993), "Literature Review and Commentary on the Verification, Validation, and Accreditation of Models and Simulations,", SCSC, pp. 10-17.

[121] Zeigler, B. P. (1976), Theory of Modelling and Simulation, John Wiley & Sons, New York.

[122] Zeigler, B. P., J. W. Rozenblit and E. R. Christensen (1991), "Reducing the Validation Bottleneck with a Knowledge-based, Distributed Simulation Environment," Expert Systems with Application, Vol. 3, pp. 329-342.

[123] Zhang, D. and D. Nguyen (1989), "A Technique for Knowledge Base Verification," TAI-89, IEEE Int. Workshop on Tools for Artificial Intelligence, pp. 399-406.