An Asset Management Framework for Educational Buildings with Life-Cycle Cost Analysis

by

Ahmed A. M. Elhakeem

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Civil Engineering

Waterloo, Ontario, Canada, 2005

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-17445-6
Our file Notre référence
ISBN: 978-0-494-17445-6

ABSTRACT

The civil infrastructure including buildings, highways, bridges, and water/sewer systems, is crucial for economic growth and prosperity. Sustaining the safety and operability of infrastructure networks is a complex task due to the harsh operational environment and the limited repair budgets. Among the various infrastructure systems, educational and health care buildings represent a great challenge due to their diverse components that have different repair requirements. While many asset management systems have been introduced in the literature, few have focused on buildings or some of its individual components (e.g., roofs). Ideally, an asset management system would include important functions such as condition assessment, deterioration prediction, repair selection, and component prioritization for repair along a planning horizon. Existing systems, however, may not adequately cover all these functions and lack optimization features that are suitable for large scale networks.

This research introduces a comprehensive asset management framework to support the efficient planning of maintenance and repair programs for educational buildings. The proposed framework has unique focus on tracking the dynamics of defects in various building components and in optimally repairing these defects. The framework introduces the following novel developments: 1) a simple visual approach to support speedy and less subjective assessment of the current severities of defects associated with various building components during field inspection; 2) a modified Markov chain approach using optimization for component-dependent prediction of future severities along the planning horizon; 3) a procedure for determining the least-cost strategy to repair component deficiencies in each year of the planning horizon; and 4) a procedure for network-level optimization to prioritize components for repair, considering practical constraints and user preferences. The framework, as such, is designed to deal with large scale networks by applying optimization sequentially to project-level decisions (repair types) and then to network-level decisions of selecting the components to repair along the planning horizon.

The framework has been implemented in user-friendly prototype and its performance tested using data obtained from a school board in North America. Based on extensive experimentation with various optimization techniques and strategies on different problem sizes, the system proved to be practical and capable of optimizing repair funds for up to 1,200 components, simultaneously, using gradient-based mathematical optimization, and for a much larger number using the genetic algorithms technique. The system as such, will aid consultants and owner organizations administering large inventory of buildings (e.g., municipalities and governmental agencies) in making appropriate decisions that ensure the sustainable operation of the infrastructure assets with least cost.

ACKNOWLEDGMENTS

First and foremost I would like to thank God whose guidance lead me this far.

My thanks also go to several people whose support helped me to bring forth this work. I would first like to express my sincere thanks and gratitude to my professor and supervisor Dr. Tarek Hegazy for his keen interest, extensive assistance, guidance, generous support and close supervision throughout the course of my program. It has been a great honor to work with him and to learn from his experience.

I also wish to thank Dr. Emad Elbeltagi for his fruitful discussions and comments that affected the quality of this work. Also, I wish to thank all friends in the construction group at the University of Waterloo.

I appreciate the valuable and practical guidance provided by many experts at the Toronto District School Board (TDSB), especially Mr. Larry Hayter, the facility condition specialist.

Last, I would like to thank my family for their emotional support and encouragement throughout the period of my studies.

Ahmed Elhakeem

TABLE OF CONTENTS

Chapter 3: BUILDING CONDITION ASSESSMENT: A VISUAL APPROACH

Chapter 4: LIFE-CYCLE-COST MODEL FOR BUILDING COMPONENTS

Chapter 5: FORMULATION FOR NETWORK-LEVEL OPTIMIZATION & REPAIR FUND ALLOCATION

Chapter 6: INTEGRATION AND TESTING

Chapter 7: CONCLUSIONS AND FUTURE RESEARCH

LIST OF FIGURES

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

LIST OF TABLES

Chapter 2

Chapter 3

Chapter 5

Chapter 6

ACRONYMS

OBS	Organizational Breakdown Structure
TDSB	Toronto District School Board
DI_j	Deterioration Index for instance j
W_i	Weight of deficiency i (percentage)
S_{ij}	Severity of deficiency i of instance j
d	Number of deficiencies for certain component
CDI	Component deterioration Index
Z	Relative size of the instance relative to the whole component
RIF	Relative Importance Factor
DI_{parent}	Deterioration Index at any level in the asset hierarchy higher than the component level
DI_{child}	Deterioration Index for any component (child) of any parent
V-CAP	Visual Condition Assessment Program
LCCA	Life-Cycle Cost Analysis
[PS]	The possible deterioration states for Markov (vector)
[TPM]	Transition Probability Matrix for Markov (square matrix)
$[IP_o]$	Initial Probability Vector (initial condition)
$[FP_t]$	Future Probability Vector (condition after t periods)
FS_t	Future state (after t periods)
$DI_{t(M)}$	Predicted DI using Markov
$DI_{t(av)}$	The average DI from inspection results
$P_{i,i}$	The diagonal probability in the [TPM]
RS	Repair Scenario
k	Repair year
DI_{AR}	DI after repair
%RC	Percentage Repair Cost
DDI	Desirable Deterioration Index
$IRC	Instance Repair Cost
$CRC	Component Replacement Cost

SOAP	Sequential Optimization Analysis Program
EP	Expected Performance
DI_N	Network Deterioration Index
$\$B_k$	Available Budget for repair at year k
GAs	Genetic Algorithms
$\$B_n$	Budget for sub-network n
A-FAP	Asset Fund Allocation Program
D-Tracker	Deficiency Tracker (developed system)

Chapter 1

INTRODUCTION

1.1 General

As the infrastructure touches almost all aspects of life from transportation and water/sewer to health care facilities and educational buildings, its ailing signs worry a broad category of the public. While natural disasters have certainly contributed to some infrastructure failures, many others have failed because of the lack of repair and maintenance, inaccurate condition assessments, improper spending, or more precisely, inadequate management.

Looking at the infrastructure report cards published by the ASCE from 1998 to 2005, it is easily concluded that limited improvements were observed in all infrastructure sectors, despite of the current spending levels. The 2005 report card gave failing grades to many infrastructure systems, and identified the need for $1.6 trillion (US) to bring the US assets to acceptable condition (ASCE 2005a). Similarly, the environmental, social, and transportation infrastructure systems alone in Canada require huge investments that amount to approximately $10 billion (US) annually for ten years (Federation of Canadian Municipalities 1999). Despite of the large need for repairs, the Infrastructure Canada Program allocated $2 billion (US) for the year 2000 for all infrastructure sectors (Federation of Canadian Municipalities 2001). In general, all infrastructure sectors face a large shortfall due to urgent needs for rehabilitation and repair. For example, the investment shortfall in municipal and regional roads is reported to be $9 billion (The National Research Council of Canada 1999), and the shortfall in public transit is $8 billion or more (The Canadian Urban Transit Association 1999). With the non-residential buildings in Canada and the United States being the largest among all other infrastructure sectors, 40% and 63%, respectively (Figure 1.1, Statistics Canada 1995 and U.S. Census Bureau 1999), such sector is expected to suffer the largest shortfall in expenditures on rehabilitation and repair.

The major component of non-residential buildings is schools and educational facilities. In the ASCE report card of 2003, the school sector was given the worst grade (D^{-}) with no improvement from 2001 (ASCE 2005 b). The report card also showed that, 59,400 schools (approximately three-quarters of the schools in the USA) require repairs, renovations, or modernization in order to attain an acceptable condition. Accordingly, about $127 billion (US) are required to restore school facilities to a good condition (U.S. Department of Education 1999). The National Education Association announced that the need is even greater, as they estimated it by more than $268 billion (US) (NEA 2000).

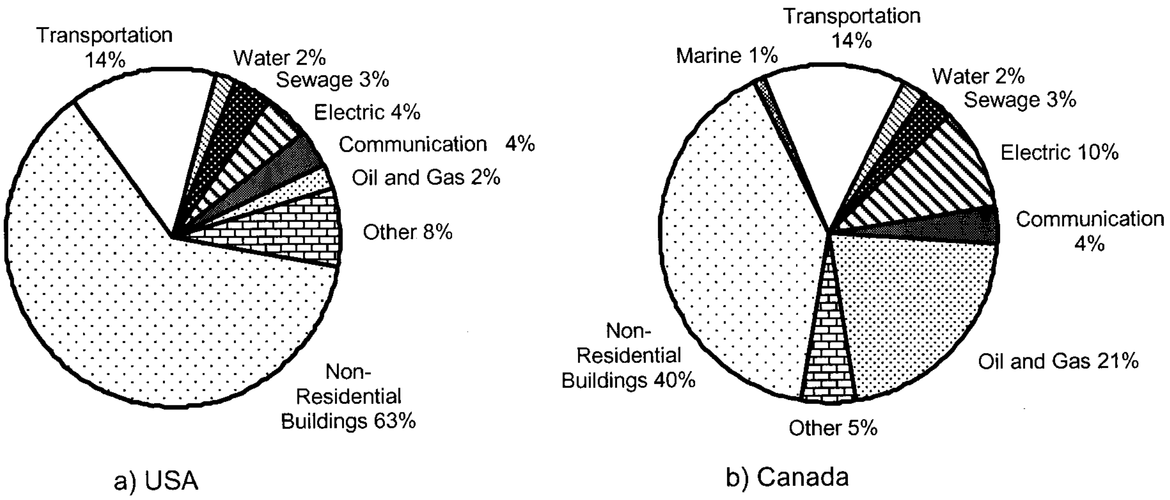


Figure 1.1: Average yearly expenditures by type of infrastructure

As a result of these huge shortfalls, there are hundreds of billions of dollars in maintenance backlog in North America (Vanier 2001). As an example, Figure 1.2 presents an age profile of the educational buildings of the Toronto District School Board (TDSB) with their expected renewal needs, compared to the funding level (RECAPP® 1.0 Training Workshop Manual 2003).

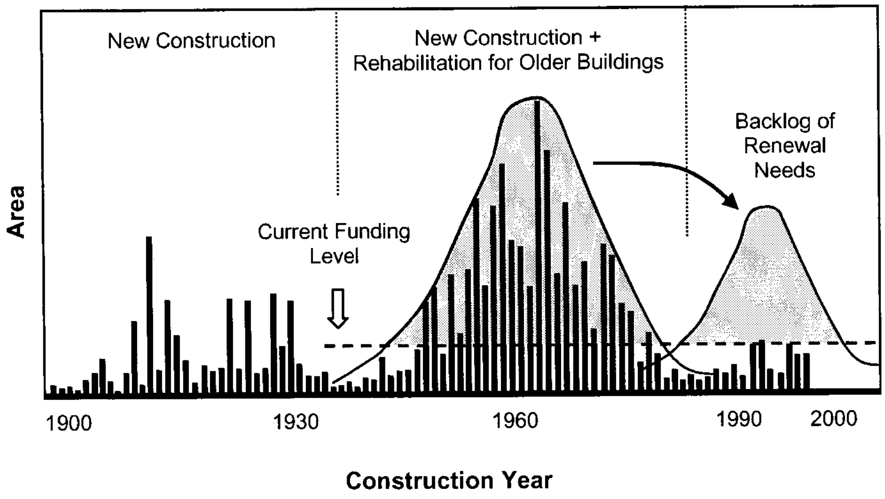


Figure 1.2: Construction-Age profile with expected backlog (RECAPP 1.0)

With the deteriorating condition of the infrastructure and the huge backlog of expenditures, municipalities and public organizations have come under increasing pressure to develop new strategies to manage public assets with limited budgets in a way to ensure long-term sustainability. Hence, the management of infrastructure assets, in general, and educational buildings, in particular, has become a complex and challenging task.

1.2 Research Motivation

The research has been motivated by the following:

Lack of Decision Support Tools: In general, there is a serious lack of standards, guidelines, and computer software to assist managers in their decisions related to repair- fund allocation and selection of cost-effective repair strategies (Melvin 1992; Coullahan and Siegried 1996; Earl 1997). Other, issues related to the inadequacy of existing decision support tools include:

1. Costly and subjective condition assessment process, particularly for old assets that lack historical information (approximately 30% of the 642 school buildings in Toronto are 50 years or older, Figure 1.3);
2. Inaccurate models for the prediction of asset deterioration behaviour and the determination of suitable repair strategies; and
3. Inappropriate mathematical approaches to consider both network-level and asset-level needs, and to optimally prioritize assets for repair purposes, under budget constraints.

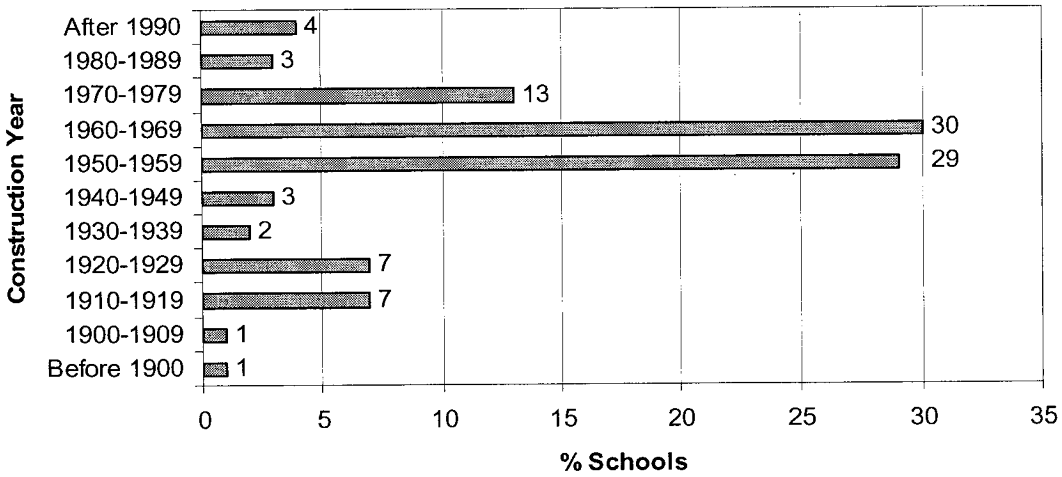


Figure 1.3: Age distribution for Toronto schools

Limited Scope of Existing Systems: Currently, a large number of Computerized Maintenance Management Systems (CMMSs) are available. Detailed information on more than 300 CMMS packages was reviewed by Vanier (2001). Many of these systems, however, are relational database applications that prove to be only isolated solutions to specific market niches. Based on an extensive study of these systems, Vanier's conclusion was that the CMMS domain is mature and is useful to manage work orders, trouble calls, equipment cribs, invoicing, time recording, and storing inventories and preventive maintenance schedules. However, the systems' capabilities with respect to life-cycle-costing and service life prediction, which are vital for asset management are limited (Vanier 2001).

Lack of Integration: Although the importance of proper inspection, deterioration prediction, life-cycle-cost analysis, and asset prioritization has been recognized by many professionals and researchers, existing tools and procedures do not consider all aspects together. Existing systems often focus on one aspect only (e.g., inspection) and do not provide an integrated, comprehensive solution to address the needs for maintaining building assets throughout their service life (IRC 1994; Kaiser 1996; Vanier 1999).

1.3 Research Objectives and Scope

The primary objective of this research is to investigate the re-engineering of the infrastructure asset management process and to structure a comprehensive asset management framework for educational buildings. Detailed objectives are as follows (Figure 1.4):

1. Investigate traditional asset management systems and identify potential improvements that are related to condition assessment, deterioration prediction, repair strategies, after-repair condition improvement, and life-cycle cost optimization;

2. Establish a methodology for accurate condition assessment of educational buildings and develop a simplified system for visual inspection, recording, and analysis of field inspection data. This system will provide visual guidance during the assessment process to make it faster, cheaper, less subjective, and suitable for less experienced users;

3. Use the condition assessment data to develop time-dependent deterioration models of various building components by integrating optimization into the Markov chain process;

4. Analyze the types and costs of different repair scenarios that apply to building components, and establish a procedure for determining the optimum repair strategy for each year in the planning horizon;

5. Develop a life-cycle-cost model for building network, considering the deterioration model, least-cost repair strategies, budget limits, and other user-constraints. Then, utilize a non-

traditional optimization technique (genetic algorithms) which suits large-scale problems to optimally prioritize assets for repair purposes; and

6. Develop a prototype, based on the proposed developments, and demonstrate its use on a real-life application.

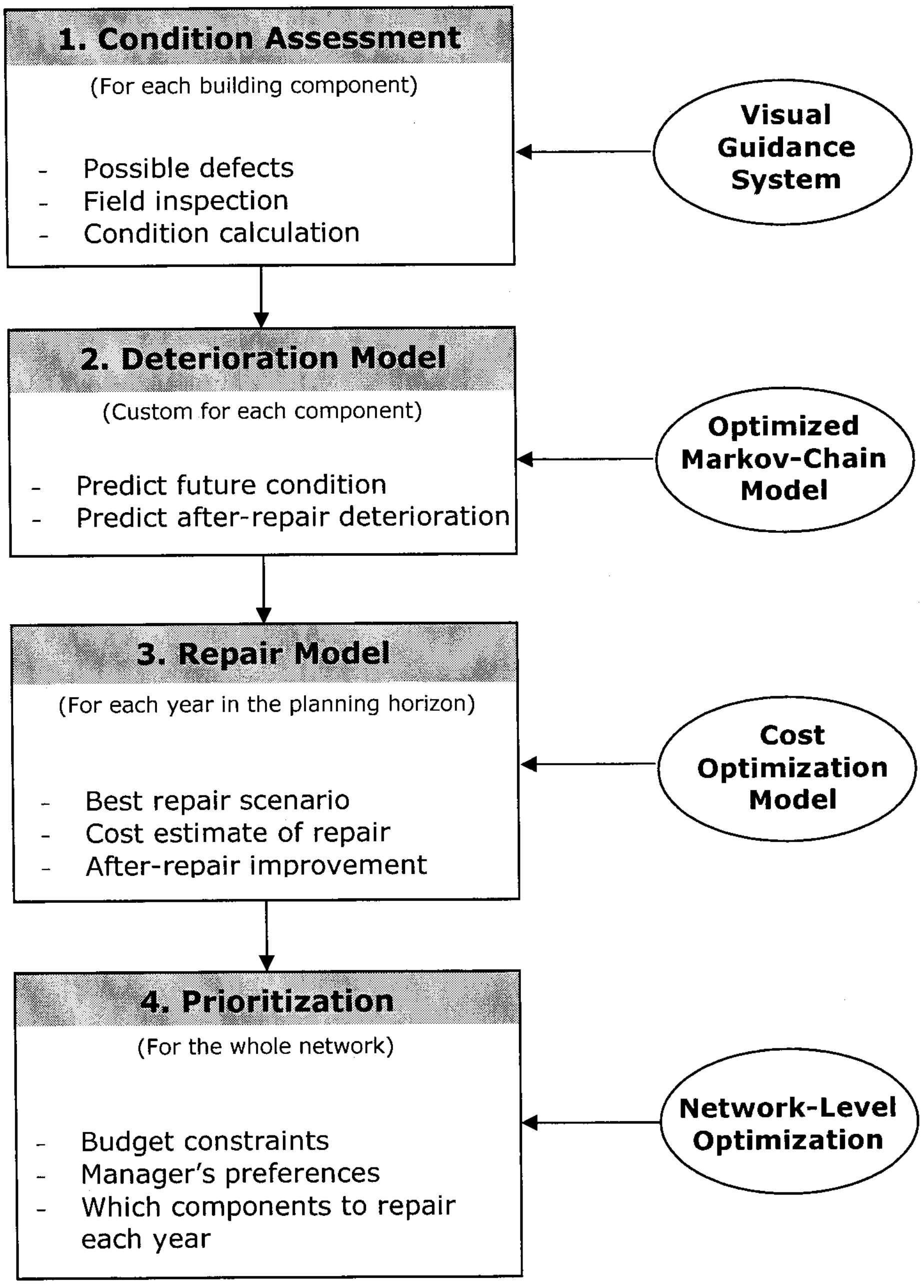

1. Condition Assessment
(For each building component)
- Possible defects
- Field inspection
- Condition calculation
Visual Guidance System
2. Deterioration Model
(Custom for each component)
- Predict future condition
- Predict after-repair deterioration
Optimized Markov-Chain Model
3. Repair Model
(For each year in the planning horizon)
- Best repair scenario
- Cost estimate of repair
- After-repair improvement
Cost Optimization Model
4. Prioritization
(For the whole network)
- Budget constraints
- Manager's preferences
- Which components to repair each year
Network-Level Optimization

Figure 1.4: Proposed framework for asset management

In essence, this research aims at structuring a comprehensive asset management system that can optimally allocate funds and prioritize assets for repair purposes. While optimization is an essential part of this research, the thesis does not claim to introduce any new mathematical optimization formulations. Rather, it introduces new approaches to represent the asset management problem into suitable decision support system in which practical variables, constraints, and objectives are represented so that optimization can be successfully applied.

Although the focus will be on asset management for educational buildings, the proposed system can be adapted to other types of assets such as bridges, highways, and airports. The planning horizon for this model will be a five-year program.

The proposed research will aid consultants and owner organizations, such as municipalities and governmental agencies, to make appropriate decisions that ensure the sustainable operation of the infrastructure assets with the least cost and optimum operational condition.

1.4 Research Methodology

The approach to be employed in this research in order to achieve the aforementioned objectives consists of the following:

1. Conduct an extensive literature review of asset management research, including techniques, software, and models;

2. Evaluate the relative importance of various building components through a questionnaire survey among building maintenance professionals. The survey determines the importance of each component to: (1) educational process; (2) safety; and (3) impact on other components. The relative importance of components are used as weights to roll up the condition at upper levels in the asset hierarchy;

3. Collaborate with the TDSB to analyze their database of buildings and repair history. This facilitates the establishment of an interactive visual database (pictures) of building components and their condition ratings. In addition, the TDSB database will be used in the analysis of defects and cost estimates of repair options;

4. Investigate the use of new advances in data recording and data transfer technology such as Tablet PCs to support the field inspection process through an easy-to-use program for recording component defects in a fast and accurate manner;

5. Use assessment data to update Markov chain through an optimization process that reduces the error between actual measurements and Markov predictions;

6. Develop a prototype for an integrated framework that incorporates condition assessment, deterioration prediction, repair selection, fund allocation, and life-cycle optimization. This prototype has two levels of optimization. One level at the component-level to determine optimum repair strategy for each year in the plan and the other at the network-level to select the components for repair during the planning horizon; and

7. Validate the developed system and demonstrate its capabilities on a real-life project and finally, present the conclusions and recommendations.

1.5 Thesis Organization

Chapter 2 presents a literature review of the traditional and most recent efforts related to condition assessment, deterioration models, repair cost estimate / impact, and asset prioritization;

Chapter 3 introduces the proposed condition assessment model and its visual guidance system to make accurate and fast assessments. Various visual guidance mechanisms are discussed then an implementation program V-CAP is presented. The concept and formulation to calculate a deterioration index to express the physical condition of any component is also explained;

Chapter 4 introduces two new approaches for deterioration and repair modeling which are applied at the single component level to present its life-cycle cost. This chapter covers also the integration between condition assessment and deterioration modeling and between deterioration and repair selection;

Chapter 5 introduces the network-level optimization to determine the optimum year for repair for each component in the network, considering the budget constraints and manager's preferences;

Chapter 6 provides the integration program (D-TRACKER), then followed by the testing process using a real-life example; and

Chapter 7 provides the conclusions and future research.

Chapter 2

LITERATURE REVIEW

This chapter presents a comprehensive review of the state-of-the-art described in several areas related to the present study, including: asset management systems, condition assessment, asset deterioration, repair strategies, asset prioritization, and life-cycle cost analysis.

2.1 Infrastructure Assets: The Ongoing Crises

The term "assets" represents all valuable items that can be owned. Assets include money, workers, materials, scientists, projects, and any type of resources that relate to an organization. One broad category of assets is the infrastructure, which is usually owned by the government or by large firms, and is used to provide the basic facilities, services, and installations needed for the functioning of a community or society. The infrastructure touches almost all aspects of life, including transportation, communications systems, water/sewer, and buildings for schools, hospitals, post offices, and prisons, etc. In general, the infrastructure is characterized by being huge in size, complex in nature, costly to build, operate, and maintain, and highly challenging to manage.

Since about the late 1970s, various ailing signs of the infrastructure caught the attention of the media and the public. A famous book "America in Ruins: The Decaying Infrastructure" by Choate and Walter (1981) has brought the attention to infrastructure deterioration consequences (loss in lives and properties). Users, investors, and public officials became more concerned after hearing some critical incidents of sudden collapses and failures of various infrastructure components. The public awareness of these incidents and the identification of potential failure areas have led to a perception of an infrastructure crisis (Hudson et al. 1997). In April 1971, the standards to develop a bridge-inspection program were issued in the United State (Infrastructure 1992). Since then, bridge management systems and inspection programs have continuously improved. This act unfortunately was a response to a real crisis, when the 39-year-old Silver Bridge collapsed in West Virginia in 1967, the collapse resulted in 46 lost lives and a lot of property damage (Hudson et al. 1997). Although the collapse history is always gloomy, it is a good motivator for researches and governments to spend money, time, and efforts. Table 2.1 provides some examples of failures, none of them were due to natural disasters such as earthquakes or tornadoes. Hence, other reasons were the cause, most probably lack of maintenance and repair, inadequate inspection and condition evaluation, improper funding / spending, or more precisely, inadequate management.

Table 2.1: Examples of Infrastructure Problems / Failures

Year	Infrastructure Crisis	Repayments	Reference
1982	An 80-year-old aqueduct failed in New Jersey, USA	Three days with no drinking water for 300,000 residents	Kwiatkowski 1986
1983	Bridge collapsed in Connecticut, USA	3 killed and 3 seriously injured	Wagner 1984
2000	High school gym roof collapsed in Cleveland, USA	Injuring 3 students and 2 adults	I Civil Engineer 2005 a
2001	A bridge collapsed in northern Portugal, LISBON, Portugal	Up to 70 people were feared dead	I Civil Engineer 2005 b
2002	School staircase collapsed in North China, China	21 teenage student died and 47 more were injured	People's Daily 2005
2002	A nine-story apartment building collapsed in St. Petersburg, Russia	3 killed, and about 430 people were left homeless.	I Civil Engineer 2005 c

2.2 Asset Management Systems

The asset management systems are those tools which can support owner organizations to better manage their assets. Asset management is defined as "a comprehensive business strategy employing people, information and technology to effectively allocate available funds amongst valid and competing asset needs" (TAC 1999).

Also, according to the Federal Highway Administration (FHWA) "Asset Management is a systematic process of maintaining, upgrading and operating physical assets cost effectively. It combines engineering and mathematical analysis with sound business practice and economic theory. Asset management systems are goal driven and like the traditional planning process, include components for data collection, strategy evaluation, program selection, and feedback. The asset management model explicitly addresses integration of decisions made across all program areas" (FHWA 1999).

In 1997, Hudson et al., described an asset management system as an operation package that consists of the (methods, procedures, data, software, policies, decisions, etc.) that enable the carrying out of all the activities involved in asset management.

Asset management, in general, involves several activities which can be grouped in different forms. Based on various sources in the literature, the main functions of an ideal asset management system cover the following aspects (Figure 2.1):

a. Condition Assessment;

b. Deterioration Modeling;

c. Repair Alternatives and Strategies;

d. Improvement after Repair; and

e. Asset Prioritization and Repair Fund Allocation.

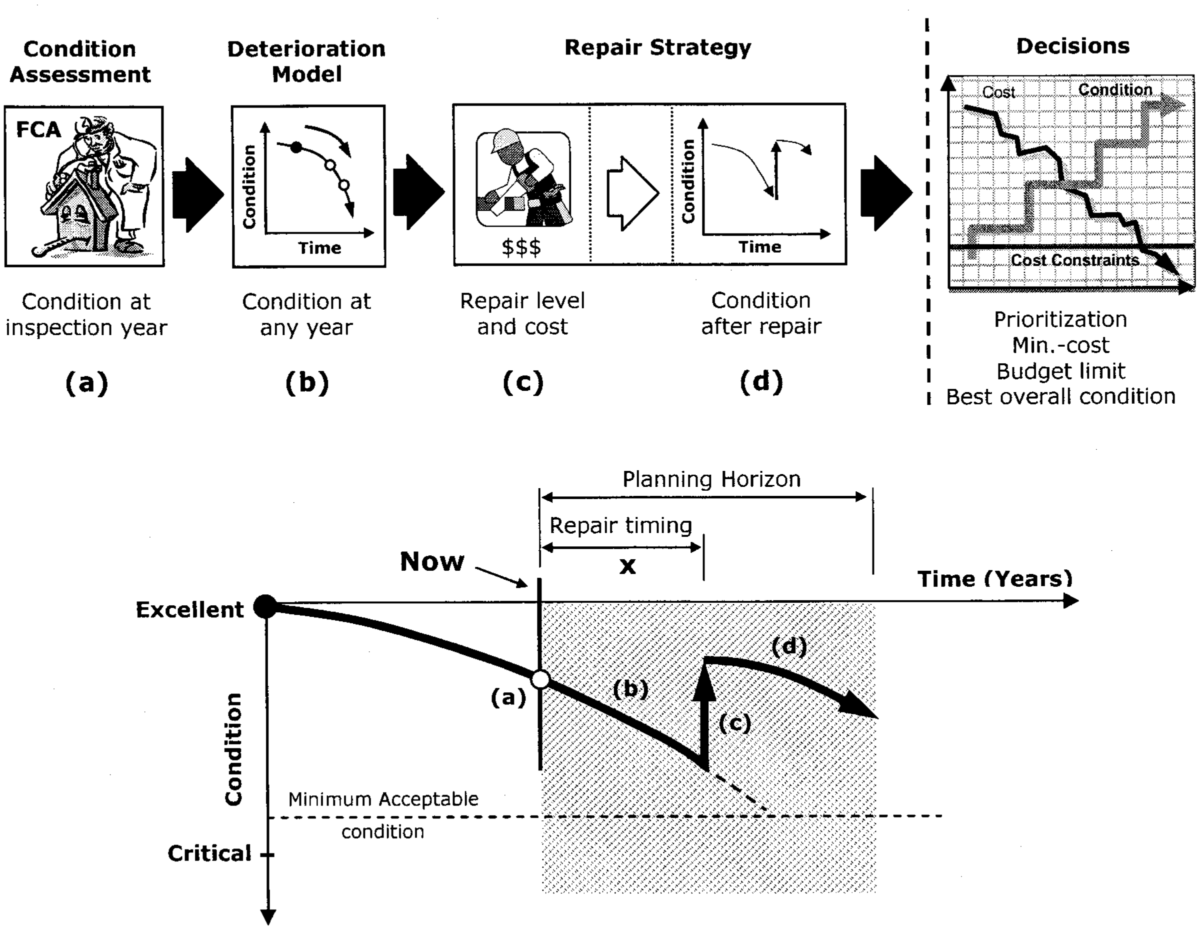


Figure 2.1: Main functions of asset management

Literature search related to each of these five functions will be covered in sub-sections 2.3.1 to 2.3.5. Figure 2.1 illustrates the importance of all these functions to the asset management process. As shown in the bottom part of Figure 2.1, the condition assessment process determines point (a) on the curve, which indicates the current condition at the time of inspection. Future deterioration (part (b) on the curve) can then be predicted using deterioration models and generally takes the shape shown in the figure. If a decision is made to repair the component after (x) years from now using a certain repair strategy and associated cost, the repair is expected to improve the condition, as shown in part (c) of the curve. The improvement in condition due to repair (length of part (c)) is dependent upon the type of repair strategy used (i.e., full replacement versus minor repair). After completion of repair in year (x), the component will still deteriorate in the future years, as shown in part (d) on the curve.

For a network of multiple assets, an asset management system can answer three main questions:

1. Which components have high priorities to be included in a repair plan (the curve in Figure 2.1 represents an example of a selected component);
2. When to repair each selected component (the value of year (x) for the component in Figure 2.1); and
3. What kind of repair should be applied for each selected component (affects the height of arrow (c) in Figure 2.1).

2.2.1 Asset Management Challenges

Developing an asset management system for buildings is not a simple task and is more challenging than other systems that have limited number of components. The level of complexity of the system depends on the type of assets being considered. To demonstrate the complexity of managing building assets, a typical school building is considered. The building can possess about 170 components (Interior Door, Roof, Boiler, Transformer, etc). Furthermore, each of these components can have several instances at the same building, i.e., a roof component for example; can have several roof sections depending upon the building size. Schools also have multiple windows, boilers, and doors. Assuming each component has only three instances, the resultant is about 500 unique components or instances. Therefore, in order to evaluate the condition of a school building, 500 discrete components (grouped into 170 categories) need to be inspected, rated, and further analyzed to finally determine the overall condition. Since these 500 components cover only one school, the degree of complexity multiplies many folds in the case of a school board that manages hundreds of schools. One example is the Toronto District School Board (TDSB) which is responsible for 642 schools, where the inspections, analyses, and ratings involve more than 300,000 components.

The complexity of managing building assets extends beyond the exhaustive condition assessment, to cover every other step including the selection of repair options and the prioritization of assets for repair purposes. This is because each component is dealt with independently throughout the whole process. Making decisions at the system level or the whole building level, rather than the detailed instance level, is neither feasible nor accurate for building assets.

2.2.2 Scope of Existing Asset Management Systems

Existing asset management systems vary in their scope and the type of functions supported by the system. Some features which are considered differently by different systems include: (1) the time horizons considered in the decision support process; (2) the ability to consider single or multiple assets and multiple components; and (3) the extent of the functions being included in the system (out of the possible five functions of Figure 2.1). These three aspects are briefly discussed as follows:

1. **Planning Time Horizons:** Existing systems generally consider one or more of three planning horizons: the strategic horizon (long term planning); the tactical horizon (five years on average); and the capital horizon (yearly program). Figure 2.2 shows the three main planning horizons as illustrated by one of the software for asset management (the RECAPP system, RECAPP user guide 2002). Ideally, the outputs of the strategic plan determine the funding level that is used as a constraint on the tactical plan. This is the situation dealt with in this study, where a certain funding constraint is preset and is used to optimize the tactical and capital plans.

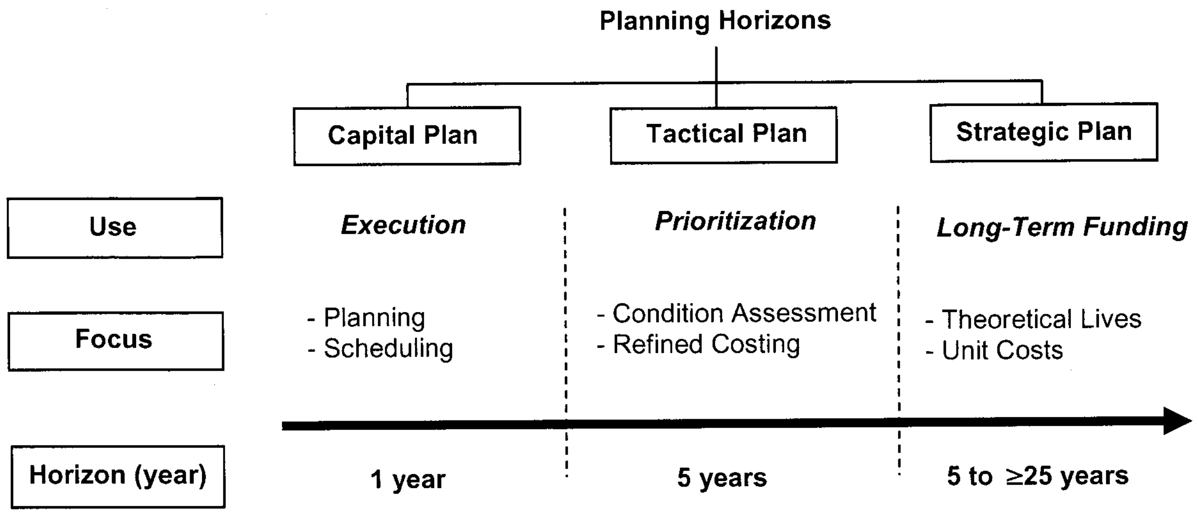


Figure 2.2: Planning time horizons

2. **Multiple Assets or Components:** Few software systems are currently available to manage building assets considering all components (e.g., BUILDER and RECAPP). Many software systems, however, exist to manage only one type of component within the building (e.g., ROOFER, which deals only with roofs).

3. **Asset Management Functions:** Although detailed asset management requires the five main functions described earlier, available systems may not cover them all and/or may not adequately cover them.

Since BUILDER and RECAPP are the two most common systems being used for buildings, they will be reviewed thoroughly in this literature. Their features, however, are highlighted as follows:

BUILDER is the newest product in the family of Engineered Management Systems (EMS) development by the U.S. Army Corps of Engineers at its Engineering Research and Development Centre - Construction Engineering Research Laboratory (ERDC-CERL) in Champaign, IL, USA. BUILDER provides engineers and facility managers with an automated tool to support decisions regarding when, where, and how best to maintain buildings and their key components. BUILDER is a Windows®-based software with many features such as: an inventory of major building components; photo imaging; checklist-style; pen-based inspections; condition indexes; condition prediction capabilities; service live adjustments; seismic and other building compliance ratings; budget planning procedures; prioritized long-range work-planning procedures; presentation graphics; and linkage to CAD software; (BUILDER user guide 2002).

ReCAPP® (Renewal Capital Asset Planning Process) was initially developed to support data gathering and reporting for audit clients. RECAPP has similar capabilities as that of BUILDER. (RECAPP user guide 2002).

2.3 Asset Management Functions

As discussed in section 2.2, the five common asset management functions are: (1) condition assessment; (2) deterioration modeling; (3) repair strategies; (4) improvement (condition after repair); and (5) asset prioritization. These functions are discussed in detail in the following sub-sections.

2.3.1 Condition Assessment

In 1993, Rugless defined condition assessment as "a process of systematically evaluating an organization's capital assets in order to project repair, renewal, or replacement needs that will preserve their ability to support the mission or activities they were assigned to serve". Condition assessment is the most important function in asset management process. Its results represent the start point for other functions such as deterioration or repair selection. It is worth to mention that, some systems (e.g., RECAPP) can use only the assessment results directly to generate a repair priority list, without using functions for deterioration or detailed life-cycle analysis.

Four main aspects are needed for a detailed condition assessment, as shown in Figure 2.3, and discussed as follows:

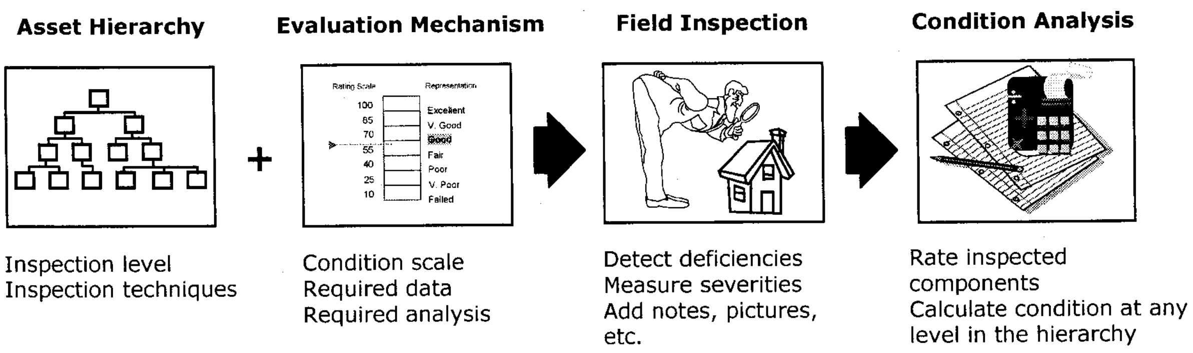


Figure 2.3: Main aspects of the condition assessment process

1. Asset Hierarchy: As an essential step for condition assessment, a building is hierarchically decomposed into its main components. The hierarchy tries to classify and cluster these components under different categories. For example, a building can be divided into different disciplines or systems (electrical, mechanical, structural …etc.), that can further be divided into a more detailed level till the component level (interior doors, exterior doors, windows, ceiling …etc.). The grouping of components into a branch in the hierarchy may be done to reflect similar characteristics (e.g., materials), similar inspection needs, or similar deterioration behavior, etc (Uzarski and Burley 1997).

An example of one of the standard hierarchies used by many researchers (e.g., Langevine et al. 2005) is the UNIFORMAT II (UNIFORMAT II 2005). Generally, however, all hierarchies start with the whole building at the upper level and end up with almost the same components included in the building. Table 2.2 illustrates the asset hierarchy used in BUILDER, where a building is divided into 12 systems, and then subdivided to a total of about 150 components. From the hierarchy in Table 2.2, it is noted that the hierarchy ends at the subcomponent level (level 4, e.g., "frame", "surface", and "hardware" subcomponents of the "interior wooden door" component). Each subcomponent is assigned an importance factor (called value factor) from 0 to 1 to facilitate the calculation of the condition from the subcomponent to the component levels. One of the advantages of this hierarchy is the use of a separate level (level 3, section) to classify components based on material / age / etc. Added to the hierarchy, BUILDER has a list of a 20-generic distress types to be used for evaluating the condition of any subcomponent.

Table 2.2: Asset Hierarchy of BUILDER 2.1 (2002)

Level 1: System	Level 2: Component	Level 3: Section	Level 4: Subcomponent		Generic criteria for condition assessment
Site Structural Roofing Exterior Circulation Exterior Closures Interior Construction Plumbing HVAC Electrical Fire Suppression Conveying Specialties	Floor Surface Interior Doors ... etc.	- Material Glass Metal Wood - Age - Area - Floor - ...etc.	Frame Hardware Surface	(0.52) (0.47) (0.71) ↑ Value factor	- Animal/Insect Damages - Broken - Clogged - Corrosion - Crack - Damage - Deterioration - Displaced - Efflorescence - Excessive Noise/Vibration - Holes - Loose - Missing - etc.
Total = 12	**Total ≈ 150**				**Total = 20**

As opposed to the hierarchy of BUILDER, another example hierarchy is that of the RECAPP system, shown in Table 2.3. In this hierarchy, four main levels are specified to decompose a building to its components then the instance level (level 5). As opposed to generic deficiencies, RECAPP has component-specific deficiencies that can be used to evaluate the condition of any instance of a component. The hierarchy, however, does not allow the grouping of instances based on material or age. Also, not all the assembly level is decomposed to components (the 133 assemblies form only 169 components, as shown in Table 2.3).

2. Condition Evaluation Mechanism: Evaluating the condition of a single instance of a component can be performed by two approaches: a distress survey and/or a direct condition rating (Uzarski 2002). Uzarski further reported that the distress survey procedure is the most accurate and reproducible approach. It also provides a record of what's wrong with the inspected instance. The direct condition rating approach, on the other hand, is less accurate but faster method for performing a condition survey. It involves visually inspecting each component and evaluating that item against a set of criteria.

Table 2.3: Asset Hierarchy of RECAPP 1.0 (2002)

Level1: Discipline	Level2: System	Level3: Assembly	Level4: Component	Level5: Instance	Component-specific Deficiency list
-Property **-Arch./ Structural** -Conveying System. -Mechanical -Electrical -Environmental -Functional	-Foundations -Superstructures -Exterior Closures -Roofing **-Interior Construction**	- Partitions - Moveable Partitions - Inter. Doors - Inter. Door Hardware **- Interior Wall Finishes** - Floor Finishes - Ceiling Finishes ...etc.	-Paint Wall covering -Vinyl Wall covering **-Stucco Wall Finish** -Ceramic Wall Tile -Glazed Wall Cover. ... etc.	**i.e.,** wall 1 wall 2 ... Wall n	-Excessive cracking / spalling (50%) -Uneven surface (15%) -Wall penetrations not properly sealed (10%) -Water damage (25%) Deficiency weights, reflecting effect on the component
Total = 7	**Total = 32**	**Total = 133**	**Total = 169 components**		**Total = 885 deficiencies**

To decide between the use of a direct rating approach or a distress survey approach, it is important to know the purpose of the assessment. To merely identify the condition of the component, then the direct condition rating approach is sufficient. But, if the purpose is to identify current problems / failures, then the distress survey approach should be used (Uzarski 2002).

The evaluation criteria, as discussed in the two hierarchies of BUILDER and RECAPP represent possible deficiencies that suit the distress survey method. In the RECAPP system, each component has its own list of specific deficiencies and their weights reflect their relative impact on condition. In the field, inspectors judge the severity of each possible deficiency and RECAPP then calculates a condition index, as will be discussed later. BUILDER, on the other hand, uses its 20 generic distress types in the evaluation process. In the field, the inspector evaluates each subcomponent against these 20 distress types, providing his judgment on two measurements (density and extent) for each distress. This process, however, is time consuming. For example, to evaluate a component with only three subcomponents, the inspector is required to provide (20 x 2 x 3 = 120) subjective measurements. Based on these measurements, condition index is calculated.

Using any system, the values of the condition indices provide means for comparing the condition of various components. The condition index scale for building components is usually from 0 to 100, where 0 represents a critical (failure) condition and 100 represents a new condition. No matter what numeric scale is used, but a linguistic representation can be made from the numeric values, as shown in the example of Figure 2.4 (used by BUILDER, Uzarski and Burley 1997). Some other condition scales and corresponding linguistic representations are shown in Table 2.4.

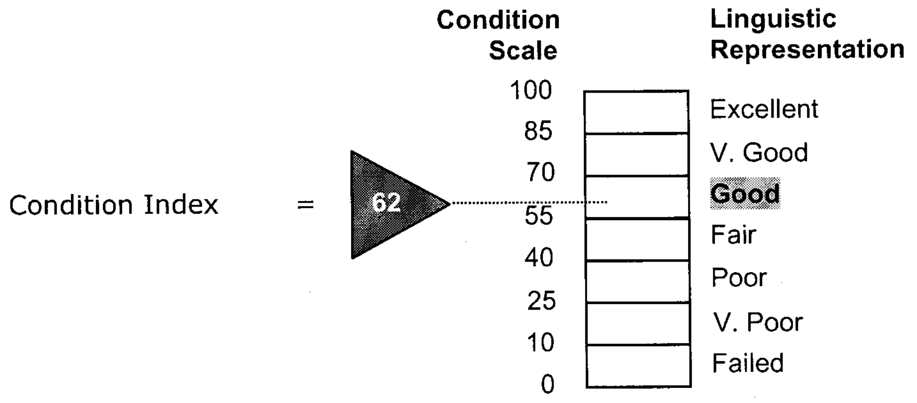


Figure 2.4: Condition scale and linguistic representation

Table 2.4: Rating Scales and Linguistic Representations

Reference	Asset Type	Condition Scale	Linguistic Representation
Lee and Aktan 1997	Buildings	1 - 4	Deterioration: (1= no; 2 = slight; 3 = moderate; and 4 = severe).
Elhakeem and Hegazy 2005a	Buildings	0 - 100	Deterioration: (0-20) = no; (20-40) = slight; (40-60) = moderate; (60-80) = sever; and (80-100) = critical.
Greimann et al. 1997	Locks and Dams	0 - 100	Maintenance need: (0-39) = only after further investigation; (40-69) = only if economically feasible; and (70-100) = no action is required.
Pontis 1995	Bridges	1 - 5	Deterioration Process: (1 = protected; 2 = exposed; 3 = vulnerable; 4 = attacked; and 5 = damaged).
Lounis et al. 1998	Any Asset	1-7	Condition category: (1 = Failed; 2 = V. Poor; 3 = Poor; 4 = Fair; 5 = Good; 6 = V. Good; and 7= Excellent).

3. Inspection and Data Collection: Evaluating the condition for building components using a distress survey requires full knowledge about the deficiencies that the component can suffer from. To detect these distresses and measure its severities, a systematic approach for field inspection becomes crucial. The mission of the inspection process is to obtain or get the required data to calculate and/or measure the performance or to evaluate the condition (calculating a numeric value that reflects certain condition).

Inspection should be performed in a fashion that is consistent, accurate, and less subjective. To standardize the process, researchers developed check lists and deficiency lists for inspection (e.g., RECAPP, BUILDER). These lists can be either in a paper form or in an electronic form. Some researchers, on the other hand, try to automate the inspection process using robots, images, satellite technology, automated devices, and/or smart sensors (e.g., Maser et al. 1997). Many programs and techniques developed in the literature can be categorized into four main groups, as follows:

- Visual inspection;
- Photographic and optical methods;
- Non-destructive evaluation methods; and
- Smart sensors.

Table 2.5 provides a summary of these efforts. Also, Figures 2.6 and 2.7 show two screen shots for the inspection survey systems used by the BUILDER, and the RECAPP systems, respectively. Both systems allow the user to add pictures, notes, general information, and detailed deficiency descriptions.

4. Analysis of Inspection Data: As the inspection process provides measurements of the severity and density of component deficiencies, some analyses are required to translate these measurements into a condition value. Once the condition of a component is calculated, it is used to calculate the condition at any level in the asset hierarch (condition aggregation).

In the literature, empirical or regression-based formulas are often used to express the component condition based on some critical measurements. One example is the following formula, determined by Greimann et al. in 1997 for Meter Lock Gate:

$$CI = 100\,(0.4)^{x/x_{\max}} \qquad \text{(2.1, Greimann et al. 1997)}$$

where (x) is the measurement that represents the anchorage movement of the gate. Such formula is determined by experts in the field and can use measurements obtained from visual inspection or smart sensor data.

Table 2.5: Inspection Techniques Used in the Literature

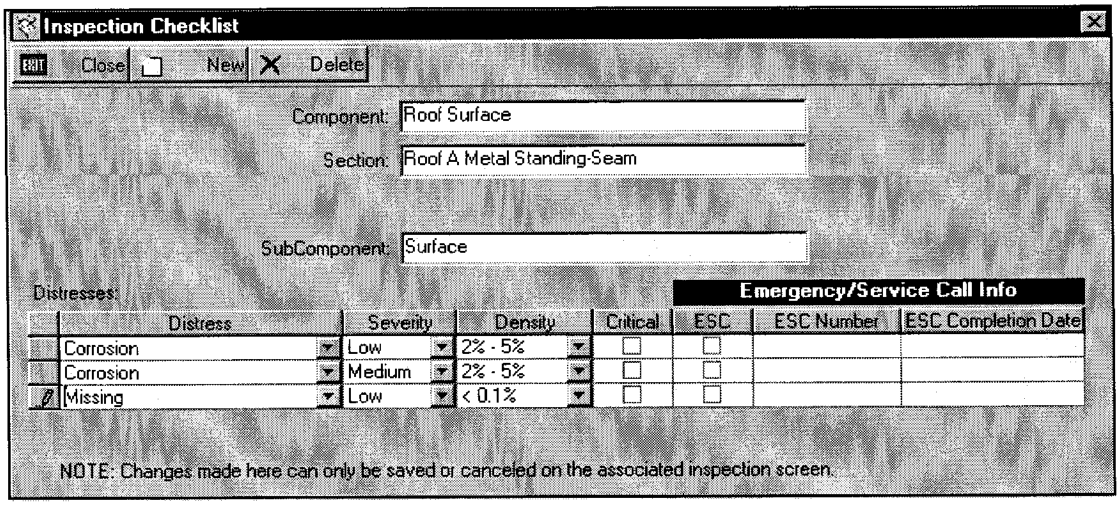
	Reference	Application Areas	Technique	Equipments	Measurements	Comments

Visual Inspection	Greimann et al. 1997;
	Uzarski 2002;
	RECAPP 2002;
	Elhakeem and Hegazy 2005a.
		Buildings, Highway, and other structures.	Data is recorded on paper or hand held devices.	Simple tools, Cameras and Subjective observation.	Anchorage movements, elevation changes, deflections, misalignments, cracks, dents, and corrosion.	**Most useful in buildings, however, time consuming, costly, subjective, labor intensive, and prone to errors.**

Photographic and Optical	Abraham et al. 1997;
	Fukuhara et al. 1990;
	Fundakowski et al. 1991.
		Mostly for bridges, Highway, and Underground Utilities.	Evaluate the condition by analyzing the images.	Video/ Digital/ Scan cameras, Closed-Circuit TV , and /or mechanical gyroscope.	Roughness, cracks and damaged areas.	Minimum disturbance to public, safe for inspectors, fast, and accurate.
						Need standardization in the area of image resolution.

Non-Destructive Evaluation	Maser and Zarghamee 1997;
	Grivas et al. 1997;
	Mason et al. 1993;
	Heiler et al 1993;
	Lee and Chou 1993;
	Maser 1995;
	Warhus et al. 1995.
		Aqueduct; Transportation infrastructure; Bridges; some Building components.	Collect images from various sources to be analyzed.	Infrared thermograph, laser, ultrasonic sensors, and ground penetration radar Equipments.	Hot or wet areas.
					Bridge deck delamination, rebar corrosion, rail wear, and pavement roughness and rutting.
						Minimum disturbance to public, safe for inspectors, fast, and accurate.

Smart Sensors	Maser et al. 1997	Bridges.	Measure the deformation and transmit the results continually using sensors.	Small self-contained battery powered transducers.	Displacements, strains, rotations, and accelerations of key bridge elements.	Real time data collection and processing.



Component: Roof Surface

Section: Roof A Metal Standing-Seam

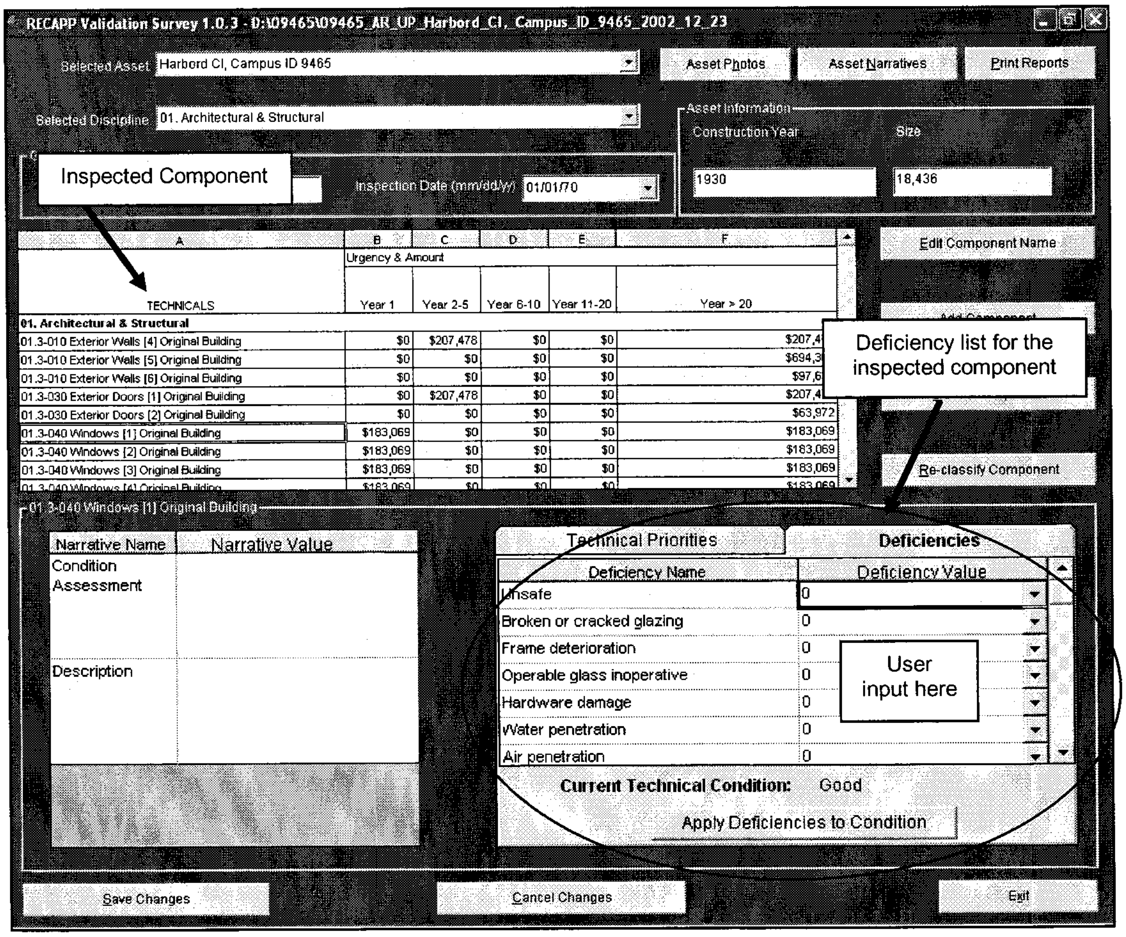
SubComponent: Surface

Distresses:

					Emergency/Service Call Info		
Distress	Severity	Density	Critical	ESC	ESC Number	ESC Completion Date	
Corrosion	Low	2% - 5%	□	□			
Corrosion	Medium	2% - 5%	□	□			
Missing	Low	< 0.1%	□	□			

NOTE: Changes made here can only be saved or canceled on the associated inspection screen.

Figure 2.5: BUILDER inspection checklist



Selected Asset: Harbord Cl, Campus ID 9465

Selected Discipline: 01. Architectural & Structural

Asset Information

Construction Year: 1930

Size: 18,436

Inspection Date (mm/dd/yy): 01/01/70

Inspected Component

A	B	C	D	E	F
	Urgency & Amount				
TECHNICALS	Year 1	Year 2-5	Year 6-10	Year 11-20	Year > 20
01. Architectural & Structural					
01.3-010 Exterior Walls [4] Original Building	$0	$207,478	$0	$0	$207,4
01.3-010 Exterior Walls [5] Original Building	$0	$0	$0	$0	$694,3
01.3-010 Exterior Walls [6] Original Building	$0	$0	$0	$0	$97,6
01.3-030 Exterior Doors [1] Original Building	$0	$207,478	$0	$0	$207,4
01.3-030 Exterior Doors [2] Original Building	$0	$0	$0	$0	$63,972
01.3-040 Windows [1] Original Building	$183,069	$0	$0	$0	$183,069
01.3-040 Windows [2] Original Building	$183,069	$0	$0	$0	$183,069
01.3-040 Windows [3] Original Building	$183,069	$0	$0	$0	$183,069
01.3-040 Windows [4] Original Building	$183,069	$0	$0	$0	$183,069

Deficiency list for the inspected component

01.3-040 Windows [1] Original Building

Narrative Name	Narrative Value
Condition Assessment	
Description	

Deficiency Name	Deficiency Value
Unsafe	0
Broken or cracked glazing	0
Frame deterioration	0
Operable glass inoperative	0
Hardware damage	0
Water penetration	0
Air penetration	0

User input here

Current Technical Condition: Good

Figure 2.6: RECAPP validation survey form, version 1.0.3

Using a deficiency list as used in the BUILDER and the RECAPP system, field measurements can be easily used to calculate condition index as follows:

Data Analysis in BUILDER: As shown in the BUILDER's hierarchy of Table 2.2, the section level identifies components by age, material, etc. For each section, samples are selected to be inspected (sampling guidelines are included in the documentation of BUILDER). The calculations consider all the subcomponents of the selected sample. The subcomponent condition index (CIS) is calculated using the weighted-deduct density model of Uzarski and Burley (1997). The model relates the observed degree and severity of deterioration for all types of distress (20 generic types, Table 2.2). Equation (2.2) is then used to calculate the condition for the uth subcomponent.

$$CIS_u = 100 - \sum_{i=1}^{p} \sum_{j=1}^{m_i} a(T_{j1} S_{j1} D_{ij}) F(t, d) \qquad (2.2)$$

where CIS_u = Condition Index for the uth subcomponent (S_u);

a = Deduct weighting value depending on distress type T_j, severity level S_j, and distress density D_{ij};

i = Counter for distress types;

j = Counter for severity levels.

p = Total number of distress types for subcomponent group under consideration;

m_i = Number of severity levels for the ith distress type; and

F(t,d) = Adjustment factor for multiple distress.

Once the CIS values for all subcomponents in the sample are calculated, BUILDER calculates the component condition index using relative value factors among subcomponents as weights (Table 2.2). The condition at any level in the hierarchy including the system level and the overall building level are also calculated using weights. The analysis is illustrated in Figure 2.7. the rolling-up process goes from bottom to top of the hierarchy, where the Parent Condition Index (PCI) is computed by taking the weighted average of its Children Condition Index, weighted by size/quantity or replacement cost.

Data Analysis in RECAPP The process to extract the condition in the RECAPP system is much easier than BUILDER because the hierarchy used in RECAPP utilizes a specific list of deficiencies for each component. Hence, only the severities for these deficiencies have to be checked and evaluated then weighted by the pre-specified weight for each defect. The weights are normalized so that the summation of weights = 100%. Equation (2.3) is used in the RECAPP system to calculate the condition.

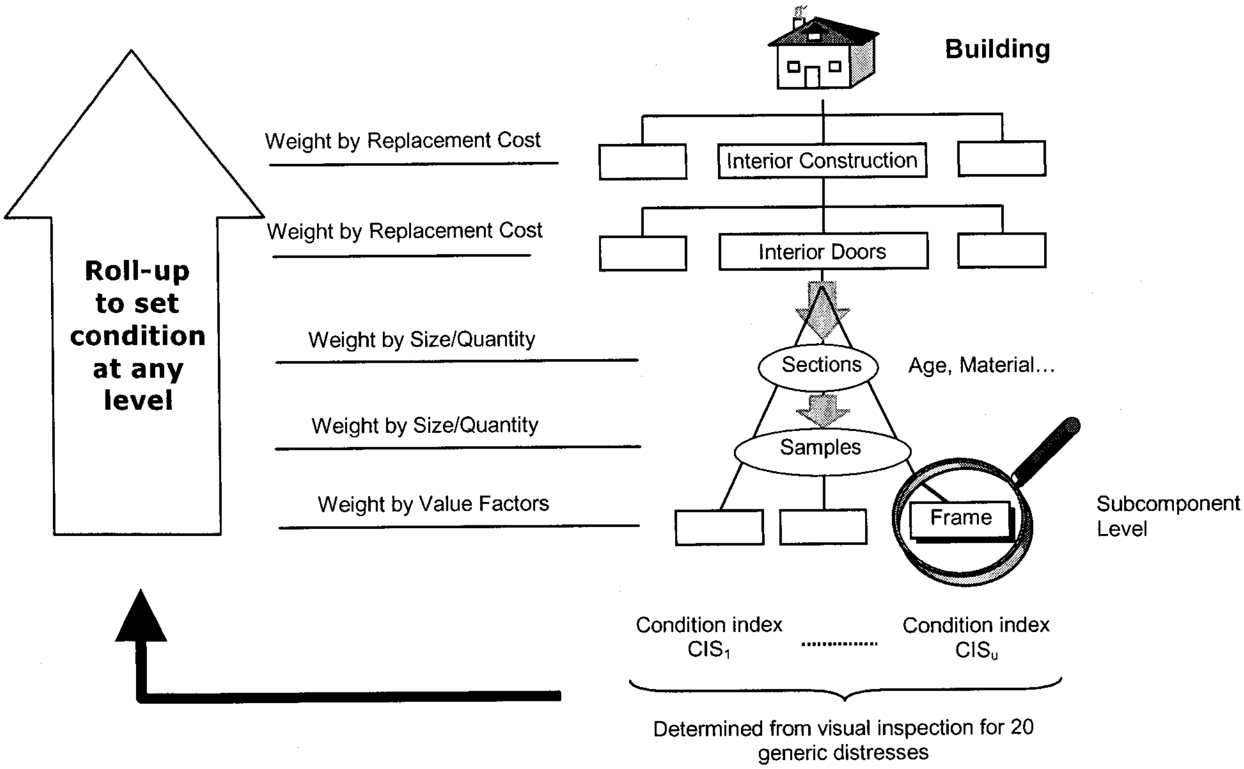


Figure 2.7: BUILDER condition assessment processes

$$CI_j = 100 - \frac{\sum_{i=1}^{d} W_i \cdot S_i}{100} \tag{2.3}$$

where CI_j = Condition Index for the j^{th} (component or section);

W_i = Weight for deficiency (i);

S_i = Severity extent for deficiency (i);

i = Counter for possible deficiencies of component (j).

The condition index is a value that ranges from 0 to 100 where:

- From (0 to 10) represents Critical Condition;

- From (11 to 24) represents Poor Condition;
- From (25 to 49) represents Fair Condition; and
- From (50 to 100) represents Good Condition.

For example, if a component has four deficiencies D_1, D_2, D_3, & D_4 with weights 10, 30, 45, & 15, respectively, and if the inspector determined corresponding severities of 25, 30, 10, & 15, then the CI = 100 – (10x25+30x30+45x10+15x15)/100) = 81.75, which implies that the component is in good condition.

Although the approach depends on deficiencies, RECAPP as shown in Figure 2.6 gives the user the option to bypass the deficiency list and give an overall evaluation for the component (i.e., good, fair, poor, or critical) based on experience. The default in the RECAPP system is to give a direct assessment.

As explained, both the BUILDER and the RECAPP systems use weights to calculate the condition of the component and to roll up the condition to higher levels in the hierarchy. These weights values however, are explained in these systems without references to the way they are deducted from. Literature also shows limited efforts to determine these weights for building components. Shohet and Perelstein (2004), for example, used the life-cycle-costs of various building systems to determine their weights. Langevine et al. (2005) used the analytical hierarchy process (AHP) to determine the weights through a process of comparison for the relative importance of the elements within each individual level in the hierarchy.

2.3.2 Deterioration Modeling

As a result of aging, severe environmental conditions, and deferred maintenance decisions, assets continuously deteriorate. Decisions related to infrastructure maintenance and rehabilitation (M&R) depend not only on assets' current (measured) conditions, but also on their predicted deterioration behaviour with time. Deterioration models, as such, are essential for any asset management system because they can predict the future condition of an asset or its components (Madanat 1993; Madanat et al. 1997; Morcous 2002a; Elhakeem and Hegazy 2005b).

Techniques for Predicting the Deterioration Behaviour: Deterioration models have been developed using a variety of techniques that can be categorized into three main categories: (1) deterministic; (2) stochastic; and (3) artificial intelligence models. Figure 2.8 summarizes these categories and describes the techniques and methods under each category. The most widely-used techniques within these categories are explained as follows:

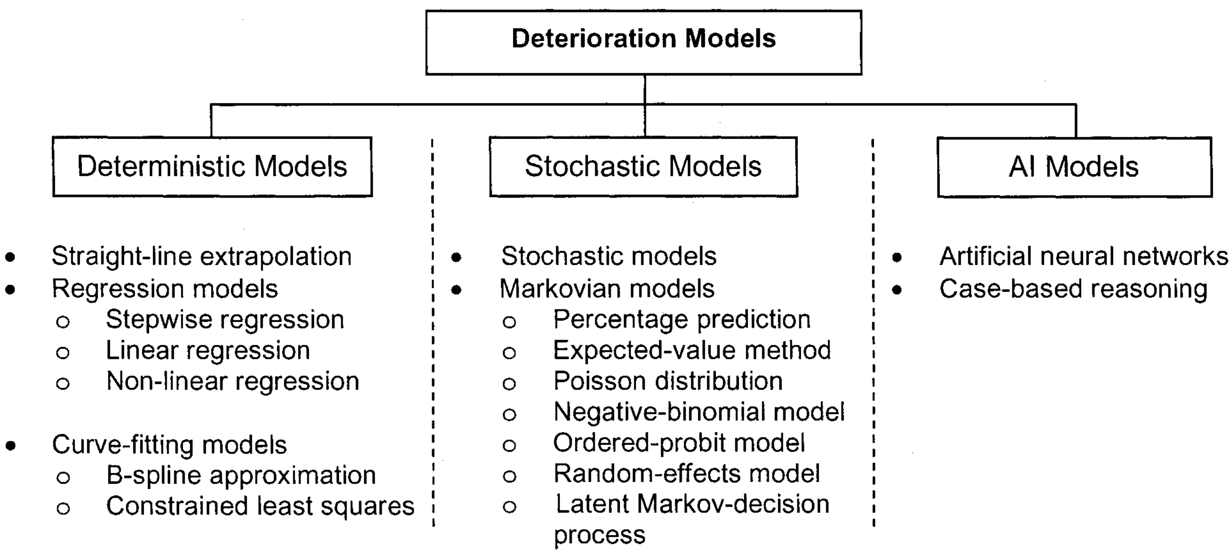
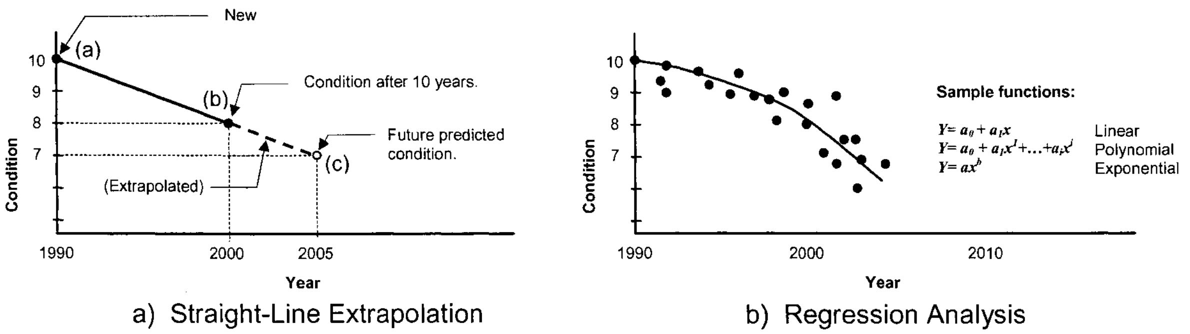


Figure 2.8: Deterioration models (based on Morcous et al. 2002b)

1. Deterministic Models: These models vary from the very simple straight-line extrapolation model (Figure 2.9 a) to the more involved regression analysis models (Figure 2.9 b). In the simple case, the model is established by stretching a line between two points (a) and (b) with known conditions, it is possible to extrapolate the future condition at any time for a third point (c), as shown in Figure 2.9 (a). This approach is being used by the BUILDER system.

Regression (Figure 2.9 b) provides a more accurate representation of future deterioration than simple straight-line extrapolation. Regression is a statistical tool that can be used to investigate relationships between variables. There are various types of regression such as linear, non-linear, stepwise, and multiple regression. The technique starts with assuming a suitable function that fits the available data. This can be done by using scatter diagram, if it shows, for example, the data seems to be possibly fitted by a line ($y = a.x + b$), then regression analysis tries to optimally determine the coefficients that represent that line (i.e., the slope (a) and the distance till intersecting with the y axis (b)). The process determines these coefficients based on minimizing the error between the predicted values and the actual ones as illustrated in Figure 2.9 (b). In non-linear regression the function can be represented as a polynomial ($y = a_0 + a_1.x^1 + a_2.x^2 + \ldots$) or as an exponential function. Various functions can be tried and the closest to data one can be selected to represent the relationship among the variables. This can be done by calculating the correlation factor (R^2) which ranges from 1 to 0, where close to one represents the best representation for the data.



a) Straight-Line Extrapolation b) Regression Analysis

Figure 2.9: Deterministic models

2. Stochastic Markovian Models: Markovian models are the most common stochastic techniques that have been used extensively in modeling the deterioration of infrastructure facilities (Butt et al. 1987; Jiang et al. 1988). These models use the Markov Decision Process (MDP) that predicts the deterioration of a component by defining discrete condition states and accumulating the probability of transition from one condition state to another over multiple discrete time intervals (Lounis et al. 1998). Transition probabilities are represented by a matrix of order (n X n) called the transition probability matrix [TPM], Equation (2.4), where (n) is the number of possible condition states.

$$\boldsymbol{TPM} = \begin{array}{c|cccc} \text{State} & 1 & 2 & \ldots & n \\ \hline 1 & p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ 2 & p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \bullet & \bullet & \bullet & p_{i,j} & \bullet \\ \bullet & \bullet & \bullet & & \bullet \\ n & p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{array} \tag{2.4}$$

As shown on the top of [TPM], condition states vary from 1 to (n), where state1 represents new condition, state2 represents a deteriorated condition, and so on, to state (n) which represents the critical condition. Each element ($p_{i,j}$) in the matrix represents the probability to move from state (i) to state (j) during a certain time interval called the transition period (t). The sum of the probabilities in each row is 1.0. Element ($p_{1,2}$) in the [TPM], for example, is the probability of deteriorating from state1 to state2. Element ($p_{2,1}$), on the other hand, represents the probability to improve from state2 to state1. Since building components only deteriorate with time, then the bottom triangle of the [TPM] matrix can be made of zero probabilities. Also reasonably assuming that, deterioration happens gradually, hence the component deteriorates only to the next state, and as such the [TPM] can be reduced to Equation (2.5).

$$TPM = \begin{bmatrix} p_{1,1} & p_{1,2} & 0 & 0 & . & 0 \\ 0 & p_{2,2} & p_{2,3} & 0 & . & 0 \\ 0 & 0 & p_{i,i} & p_{i,i+1} & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & . & p_{n,n} \end{bmatrix} = \begin{bmatrix} p_{1,1} & 1-p_{1,1} & 0 & 0 & . & 0 \\ 0 & p_{2,2} & 1-p_{2,2} & 0 & . & 0 \\ 0 & 0 & p_{i,i} & 1-p_{i,i} & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & . & 1 \end{bmatrix} \qquad (2.5)$$

Using the same matrix representation, condition of a component at any time (t) can be represented in a row vector as follow: condition = [0.5 0.3 0.1 … 0], where the values in the vector represent the probabilities of belonging to the various condition states. As a special case, the condition at a new component, at (t = 0), is defined as shown in Equation (2.6), this is called the initial probability row vector [IP$_o$].

$$\begin{array}{lccccc} & \text{State} \quad 1 & 2 & 3 & \ldots & n \\ IP_o = & [1.0 & 0 & 0 & \ldots & 0] \end{array} \qquad (2.6)$$

Given the [IP$_o$], then the future condition vector [FP$_t$], after (t) transition periods, can be calculated as follows (Collins 1972):

$$\boldsymbol{FP_t = IP_o \cdot TPM^t} \qquad (2.7)$$

Once the [FP$_t$] vector is calculated, then a single condition state value can be obtained as follows:

$$\boldsymbol{ST_t = FP_t . PS} \qquad (2.8)$$

where, [PS] is the vector of possible states,

$$\boldsymbol{PS} = \begin{bmatrix} S_1 \\ S_2 \\ \bullet \\ S_n \end{bmatrix} \qquad ,e.g., \quad = \begin{bmatrix} 1 \\ 2 \\ \bullet \\ n \end{bmatrix} \tag{2.9}$$

As such, the predicted state (ST) at any time (t) can be generally presented as follows:

$$[ST_t]_{1\times 1} = [IP_o]_{1\times n} \cdot [TPM]^t_{n\times n} \cdot [PS]_{n\times 1} \tag{2.10}$$

Equation (2.10) predicts the condition state after (t) periods from the known condition at time (0); it is possible as such, to use this equation with different (ts), (i.e., t = 1, 2, 3..., T) to determine the relationship between condition and time (the performance curve).

While the Markov chain is the most widely used approach for deterioration modeling, it has some drawbacks, as listed in Table 2.6. In view of these drawbacks, improvements are needed to increase the accuracy of future condition prediction, as discussed in Chapter4.

Table 2.6: Pros and Cons of Markov Chain Approach

Pros	Cons
Captures the uncertainty of the deterioration process;	Assumes fixed transition probabilities (Collins 1972); Assumes state independence for simplicity (DeStefano and Grivas 1998; Madanat et al. 1997);
Future condition is a function of current condition;	Do not predict the condition improvement after repair (Madanat and Ibrahim 1995);
Can be improved to suit the present study.	Difficult to consider the interaction among different components (Sianipar and Adams 1997); and
	Require updates when new data are obtained from inspection.

3. Artificial Intelligent (AI) Techniques: Among available AI techniques, Artificial Neural Networks are the most common approach that suites the area of deterioration prediction. Huang and Moor (1997), Lou et al. (2001), and Yang et al. (2003) presented models for pavement deterioration prediction. Limited efforts using other AI techniques also exist, including: fuzzy logic systems (Bandara and Gunaratne 2001), genetic algorithms (Shekharan 2000; Hedfi and Stephanos 2001), and hybrid systems (Abdelrahim and George 2000). A detailed summary of AI applications for deterioration modeling can be found in Flintsch and Chen (2004).

2.3.3 Repair Modeling

Repair modeling means determining suitable repair options and estimating their costs and impacts on condition. In current management systems, the possible types of repair are set subjectively as minor, moderate, or major repair, etc., that are assigned fixed cost as percentages from replacement cost (Seo, J. 1994). This however, may not be accurate in the case of buildings, as these classifications do not define clear strategies to repair specific deficiencies.

The replacement cost for a component is often easy to calculate either from published data references or by soliciting quotes to contractors / suppliers. Published data such as the R.S. Means (2000) includes cost data relevant to almost all types of building components. R.S. Means publishes two notable reference books: Square Foot Costs, and Assemblies Costs, which can be used for elemental estimating. Both references were developed based on an average of over 11,500 actual projects reported to R.S. Means from contractors, designer, and owners. R.S. Means simply provide tables for the cost per square foot of various projects, in addition to adjustment factors for project size and city indexes (Hegazy 2002).

When that replacement cost is defined, the cost to repair a component can be difficult to calculate. One common simplification is to consider the repair cost as a percentage of the full replacement cost. For example, the repair options (light-repair, medium-repair, and extensive-repair) for bridge decks were estimated by Seo, J. (1994) to cost 28.5%, 65%, and 100%, from the replacement cost, respectively.

In the **BUILDER** system, as another example, the replacement cost is calculated in a detailed manner based on the quantity of work, multiplied by the unit cost and other multipliers such as the area cost factor (ACF) which accounts for the following geographic differences in costs (BUILDER user guide 2002):

- Local costs of construction labor, equipment, and materials;
- Weather, climate, and seismic requirements;

- Labor availability; and
- Labor productivity.

Using the replacement cost, the cost to repair a component in BUILDER is then calculated based on current condition index using a proprietary algorithm. Not much information, however, is provided on the algorithm used.

The **RECAPP** system, on the other hand, uses a replacement strategy as a default and estimates replacement cost using fixed cost tables. The costs in these tables are stored per square foot of the educational area.

Many other researchers also discuss the importance of defining the cost of alternative rehabilitation options in any asset management system, however, without much detail about the cost calculations.

In addition to cost calculations, a model to predict the condition improvement due to repair is needed. In the literature, improvement due to repair is usually considered through empirical judgments or by performing another round of after-repair inspection (Langevine et al. 2005). One example of a pre-defined table for identifying the future condition after repair, proposed by Seo, J. (1994), for bridge decks is shown in Table 2.7. In the building domain, Langevine et al. (2005) used the data in Table 2.8 to express the expected improvement for various rehabilitation options.

This approach to improvement prediction as such, represents a simplification and improvements are needed, as discussed in Chapter 4.

Table 2.7: Predicted Improvement after Rehabilitation

Condition Rating		Condition after rehabilitation	
		5, 6	7, 8
Condition before rehabilitation	3, 4	Medium	Extensive
	5, 6	Light	Medium
	7, 8	-	Light

Table 2.8: Predicted Improvement after various Rehabilitation Options

Condition		Replacement	Major	Minor	Preventative
Best	A	-	-	-	-
	B	-	-	-	-
	C	-	-	B	-
	D	A	B	C	D
Worse	F				

2.3.4 After-Repair Deterioration

Once the improved condition due to repair is determined, it is important to determine the deterioration behavior in the periods after the repair time. Commonly, researchers assume that the deterioration trend is parallel to the deterioration trend before repair, as shown in Figure 2.10 (Seo, J. 1994; Hegazy et al. 2004; Langevine et al. 2005).

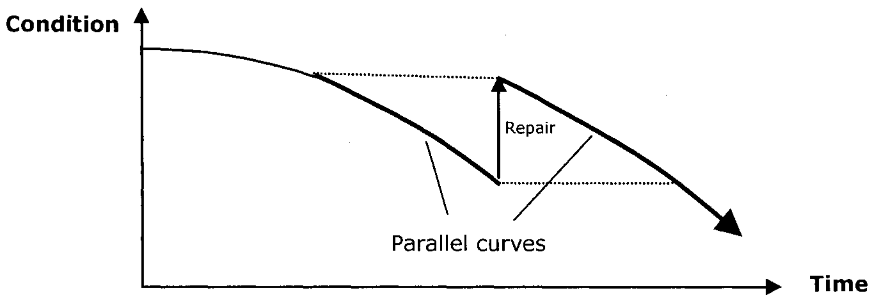


Figure 2.10: After repair deterioration

2.3.5 Asset Prioritization for Repair Purposes

One of the main challenges facing asset managers is the distribution of the available budget in order to maintain asset conditions within satisfactory levels or to maximize the benefits of the spent money (Al-Battaineh et al. 2005). In the ideal situation of unconstrained budgets, all repair needs of all components can be addressed (Hudson et al. 1997). However, repair funds for most public infrastructure agencies are limited or constrained. In this case, asset prioritization for repair purposes becomes crucial, and decisions regarding which components to repair, the repair strategy, and the repair timing need to be practically and efficiently decided.

Asset priorities can be set by various methods, ranging from simple subjective ranking to a rigorous optimization in which all possible combinations of "which, what, and when" are evaluated, with respect to an objective function (Hudson et al. 1997). Table 2.9 summarizes some of existing methods. Some explanation is provided as follows:

1. Asset Prioritization Based on Ranking: In practice, ranking is the prevailing approach for asset prioritization. As shown in Table 2.9, ranking can be performed in three ways (Hudson et al. 1997):

- Simple subjective ranking which is easy to apply in most situations, but the results may not be the best;

- Ranking with weights to account for the amount of usage and/or the relative importance depending on the infrastructure involved, such type provides much more rational and valid results; and

- Ranking with weights and economic analysis, in such a type a closer-to-optimal result is more likely to be ensured.

Table 2.9: Existing Prioritization Methods (based on Hudson et al. 1997)

Method			Simplicity	Optimality	Comments
			complex simple	low high	
Ranking		Simple subjective ranking	□ □ ☑	☑ □ □	Inconsistent
		Parameter based ranking	□ □ ☑	☑ □ □	Consider for example the condition or level of service.
		Parameter/economic based ranking	□ ☑ □	□ ☑ □	Close to optimum
Optimization	Near- Optimization methods	Marginal cost-effectiveness approach	□ ☑ □	□ ☑ □	-
	Optimization methods	Year-by-year based optimization	□ ☑ □	□ ☑ □	The timing effect is not considered
		Comprehensive optimization	☑ □ □	□ □ ☑	-

The procedures for the priority-setting of many public works agencies have been reviewed by Hatry and Steinthal in 1984. They provided the rating criteria that are used by others which are summarized as shown in Table 2.10.

In the building domain, different systems use different prioritization criteria. Both BUILDER and RECAPP depend on scoring each work item (instance or section) and giving the highest score, the priority to receive the available funds. The expected repair year is determined directly by the inspector during the inspection process. As such, the calculated score does not reflect when to repair. More details are illustrated as follows:

Table 2.10: Ranking Criteria

Need	Environmental Factors	Resources
– Condition/structural integrity – Frequency/existing problems – Expected lifetime of existing infrastructure – Likelihood of future problems/danger – Risk/magnitude of future problems – Consequences of delay – Customer complaints – Availability of emergency backup	– Geographic disbursement/equity – Probable support/opposition – User costs and preferences – Community disruption – Area wide needs – Number of users benefited – Importance to transit or commerce	– Funding availability/capital costs – Operating and maintenance costs – Staff workloads – Probability of completion

BUILDER: the system allows the user to specify some criteria (termed objectives) for the prioritization process. These objectives (e.g., Building Use, System, Current Condition Index, etc.) are arranged in levels, forming a tree format called the objective tree. The user is allowed to specify the objectives to be used in the prioritization process and their relative weights. For each objective, its sub objectives are given relative importance values (called point values, out of 100). Based on this user-defined objective tree, BUILDER calculates the score fro each component, depending on its use, system, and condition (for example).

RECAPP: Asset prioritization is addressed in the system based on calculating a score (called Priority Factor) for each possible repair for each component. The score is determined from six parameters (criteria), A, B, C, D, E, and F. Parameters "A" and "B" represent the relative importance of the related building and its capacity, respectively. Also, parameters "C" and "D" represent the relative importance of the component and the value of its condition index, respectively. Parameters "E" and "F" relate to the repair type and the level of difficulty in doing this repair, respectively. To calculate the prioritization score, the six parameters are weighted subjectively.

Based on this discussion, while ranking is simple, a lot of subjectivity is involved. Also, the impact of the measured condition, which reflects repair need, on the prioritization process is not noticeable, despite of the great effort and cost spent on the condition assessment process.

2. Asset Prioritization Based on Optimization: As discussed earlier, prioritization can be represented as a problem of distributing limited funds among a group of components in a certain network. This problem is a multi-attribute decision making problem requiring simultaneous evaluation at both the network level (i.e., which component to repair), as well as the project level (i.e., which repair strategy for each component) (Hegazy et al. 2004). Accordingly, the decisions regarding which components to repair, the repair strategy, and the repair timing represent key decisions that need structured formulation to provide best solutions. Due to the large size of building networks, this problem becomes a combinatorial optimization problem in which the solution space is extremely large. The quality of the solution, therefore, will depend on how the optimization problem is formulated and on the optimization tool used.

In the literature, various models have been developed to support either network-level or project-level decisions individually, and to a less extent to support both of them. With optimization requiring a suitable objective function, the majority of researchers have proposed the use of total life-cycle cost to be minimized (e.g., Hegazy et al. 2004). The Life-Cycle-Cost (LCC) is the total discounted cost of owning, operating, maintaining, repairing, and disposing of the asset or the system over a period of time. The main benefit of considering LCCs is the fact that decisions are made considering the benefit gained along the whole planning horizon.

The models that consider life-cycle cost analysis can be classified into two categories: near optimization; and optimization, as follows:

Near-Optimization Methods: Near optimization, based on a marginal cost-effectiveness method, has found considerable application in the roads and pavements area (Hass 1994), and could be applied in many other infrastructure areas. The approach calculates the effectiveness (E) of various repair decisions by adding the areas under the performance curve, weighted by the usage or length, etc. (Figure 2.11), Hudson et al. (1997).

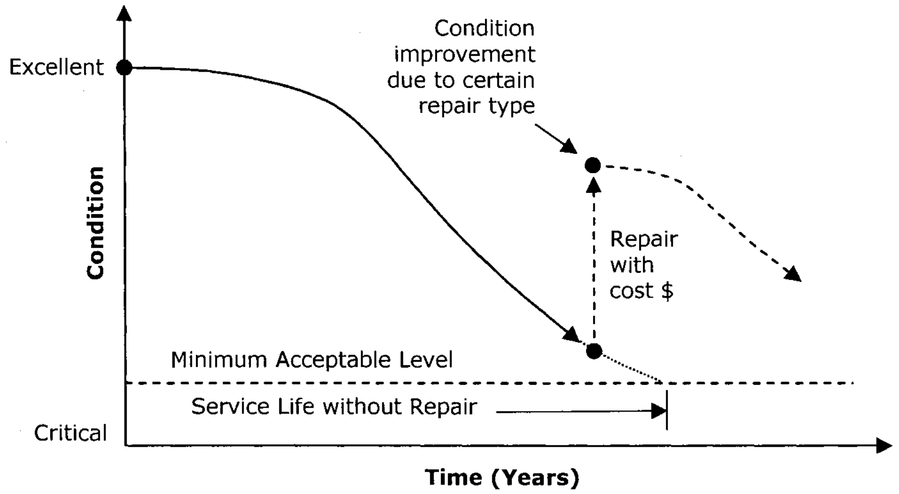


Figure 2.11: Effect of the repair strategy on the performance

Afterward, it selects the best repair decision for each section as the one that has the best cost-effectiveness ratio. Although this approach considers the most-effective repair option, it works basically at the project level, which might not be optimal at the network level, if limited budgets are considered.

Optimization Methods: Optimization can be used for either single-year or multi-year prioritization with LCC, or other, being the objective function. The process searches through the available alternatives to satisfy a specific objective function (i.e., a value is maximized, such as effectiveness, or minimized, such as costs). In the literature, few efforts are available, mostly in the area of bridges. Hegazy et al. (2004) for example, introduced a model that prioritizes bridge deck repairs considering the life-cycle costs over a five-year planning horizon. The model uses deterioration data for various types of bridges found in the literature then utilizes genetic algorithms to find the optimum repair decisions (type and timing of repair). In the building domain, Langevine et al. (2005) formulated an optimization problem over a short-term (one year) using integer programming. The objective function was set to maximize the performance of the building within the available budget constraint. Being limited to a single year, however, does not benefit from LCC analysis.

Genetic Algorithms: Considering both network-level and project-level decisions complicates the life-cycle cost analysis and makes traditional optimization tools insufficient to deal with the large problem size. Accordingly, new evolutionary optimization techniques (e.g., GAs) have evolved and can be suitable for large problems. Genetic Algorithms (GAs) have been in use to optimize complex problems in different areas. Such problems were characterized by being huge in size and including many interrelated relationships that make from traditional techniques such as, linear programming an inappropriate or inadequate tool. GAs is an outcome of the recent developments in artificial intelligence and computer technology, first introduced by Holland in 1975. Such non-traditional optimization technique was emerged as an efficient method for searching complex solution spaces for the global optimum. Genetic algorithms work by emulating the natural evolution in living organisms through a process of crossover and mutation among a group of random parent solutions and cycles of generating and testing offspring solutions until the optimum or close-to-optimum solution is found, Holland (1975).

GAs has been successfully applied in numerous areas in civil engineering and construction management. Close to current study, GAs applications including maintenance planning (Dogaki et al. 2000), life cycle management (Itoh et al. 1997), multi objective deck rehabilitation (Liu et al. 1997), bridge management decisions (Maiyamoto 2001, Hegazy et al. 2004), and infrastructure budget allocation (Al-Battaineh et al. 2005). The use of the GAs technique, therefore, has good potential for use in asset management optimization.

2.4 Summary and Concluding Remarks

A summary of the literature with respect to the various aspects of this research is as follows:

Condition assessment: Although there is a variety of techniques and technologies that can be applied to perform the assessment, visual inspection is the most suitable approach that can suit the nature of the buildings assets, because the diversity of components involved in such type of assets. Other techniques may suit few types of components in the building, but not all of them. As it was obvious from the literature review, visual inspection was described by many researchers as a costly time consuming technique. Moreover, it provides subjective and less accurate results. Current improvement in this area is in providing electronic checklists or deficiency lists to overcome the subjectivity and making the process faster. Although these tools claim to have deficiency lists, it is often bypassed to use quick subjective assessment to save time. Also, no mechanism available to carry the deficiency measures into condition predictions. These systems can be described as good databases that provide enough spaces to add pictures and notes during the condition assessment process, to be analyzed later. However, they do not provide simple-to-use tools, need professional inspectors to do the assessment, and lack of providing the required guidance to do correct assessments during the inspection process.

Models for Deterioration Prediction: In the area of prediction, the main directions on the research level were to examine the use of new approaches such as AI techniques. In the practical field, however, predictions based on regression analysis or Markov models are in extensive use. It was noted that, comprehensive systems prefer to use less sophisticated approaches such as regression analysis or even a less accurate models such as linear extrapolation as of the BUILDER program. The founders of BUILDER claim to change the prediction module to a more sophisticated one in the near future. RECAPP, on the other hand, pass over the predictions and prioritize the assets by using the inspection data directly. In general, Markov model provides suitable and accurate predictions for such type of projects that are uncertain in nature. The main challenge of Markovian models however, were mainly in, how to obtain the transition probability matrix for each component in the building, and the difficulty to update the transition matrix in case of new data becomes available.

Models for Repair Strategies: The current approach classifies the repair strategies to types such as minor repair, major repair, and replacement. For each type of repair, the approach assumes a fixed percentage from replacement to express the cost and a fixed impact (after-repair condition) based on the current condition and the type of repair. Predefining repair types however, is neither practical nor well defined in case of building components. A repair type as such, can have various meanings. As a conclusion, although the replacement cost can be determined accurately based on systematic calculations, the repair cost and improvement are still subjective assumptions.

Asset Prioritization: This area is the least developed area of asset management. Current approaches used in existing asset management systems to prioritize building components have the following drawbacks:

- Asset prioritization based on simple ranking does not consider the year of year or the life-cycle costs;
- Life-cycle cost analysis made at the project-level, separate from the network-level can lead to less optimum decisions; and
- Gradient-based mathematical optimization does not suit large-scale problems, particularly when project-level and network-level decisions are considered together.

Chapter 3

BUILDING CONDITION ASSESSMENT: A VISUAL APPROACH

3.1 Introduction

This chapter proposes a simplified approach for accurate assessment of the current condition of assets. The proposed approach aims at supporting the condition assessment process by introducing a visual guidance system that makes the visual inspection faster, less subjective, and suitable for less experienced personnel. Details of the proposed assessment system are presented in this chapter along with an example application to illustrate its usefulness in assessing the condition at any level in the asset hierarchy. The main features of the implementation program (V-CAP) and how to manage the assessment process are also covered.

3.2 Proposed Condition Assessment System

As concluded from Chapter 2, the drawbacks of existing condition assessment systems are:

1. Lack of logical decomposition of the building into a suitable hierarchy;
2. Cumbersome method for calculating condition indices for building components; and
3. Slow, costly, and subjective field inspection.

To overcome these drawbacks, a condition assessment system has been proposed with the following three main functions (Figure 3.1):

1. A logical asset hierarchy;
2. A mechanism for condition evaluation and condition calculation; and
3. A visual guidance system to support field inspection.

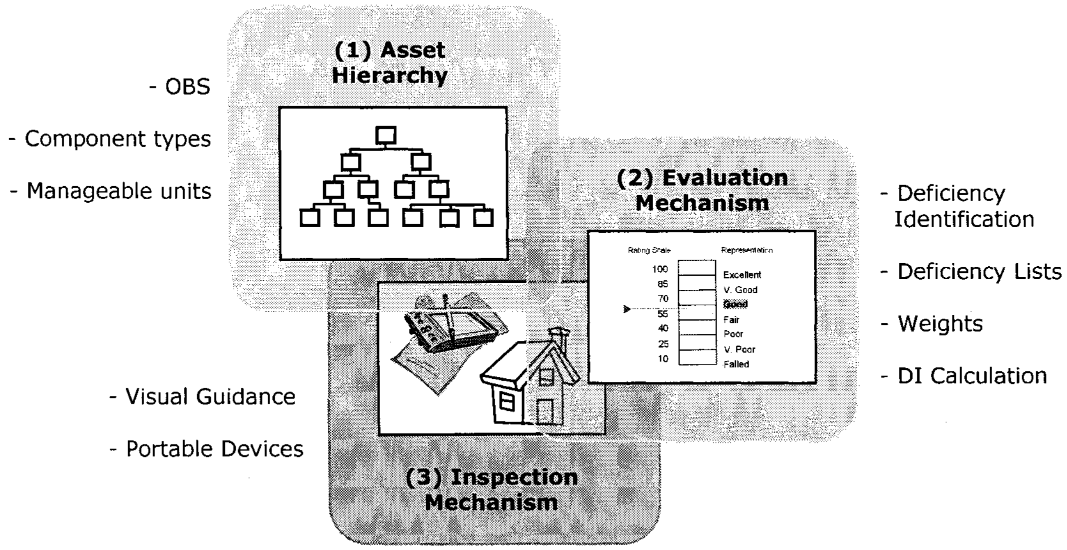


Figure 3.1: Main functions of the proposed condition assessment system

The three functions in Figure 3.1 interact with each other to serve the objectives of evaluating the condition of each instance of all building components. For example, the asset hierarchy expands to the detailed instance level, at which the inspection can be carried out. The evaluation mechanism, on the other hand, defines measurable criteria that can be easily assessed during the physical inspection. Also, the calculated conditions are suitable for aggregation at any level of the proposed asset hierarchy. The developments made within these three functions are explained in the next subsections.

3.2.1 Proposed Asset Hierarchy

The proposed condition assessment system uses an asset hierarchy that reaches to a manageable level that suits the decision making / planning and the physical execution of repair. Based on a study of the pros and cons related to various hierarchies in the literature (section 2.3.1), a new asset hierarchy that combines the benefits of existing hierarchies has been structured in five levels (Figure 3.2):

1. **System level:** includes aggregated systems such as Site Work, Structural, Architectural, and Mechanical;
2. **Subsystem level:** includes groups of components under each system (e.g., the Exterior Closure under the Architectural System);
3. **Component level:** includes all various items in the building such as, interior doors, exterior

doors, boilers, windows, transformers, and ceiling;

4. **Type/Element level**: to account for various types or elements of the same component (e.g., wooden versus steel windows or the main elements of a roof system such as the roof membrane, surface, and flashing); and

5. **Instance level:** to specify the various instances of the same component at different locations within the facility (e.g., east, west, etc.) or other user specified criteria such as age, deterioration, or historical value.

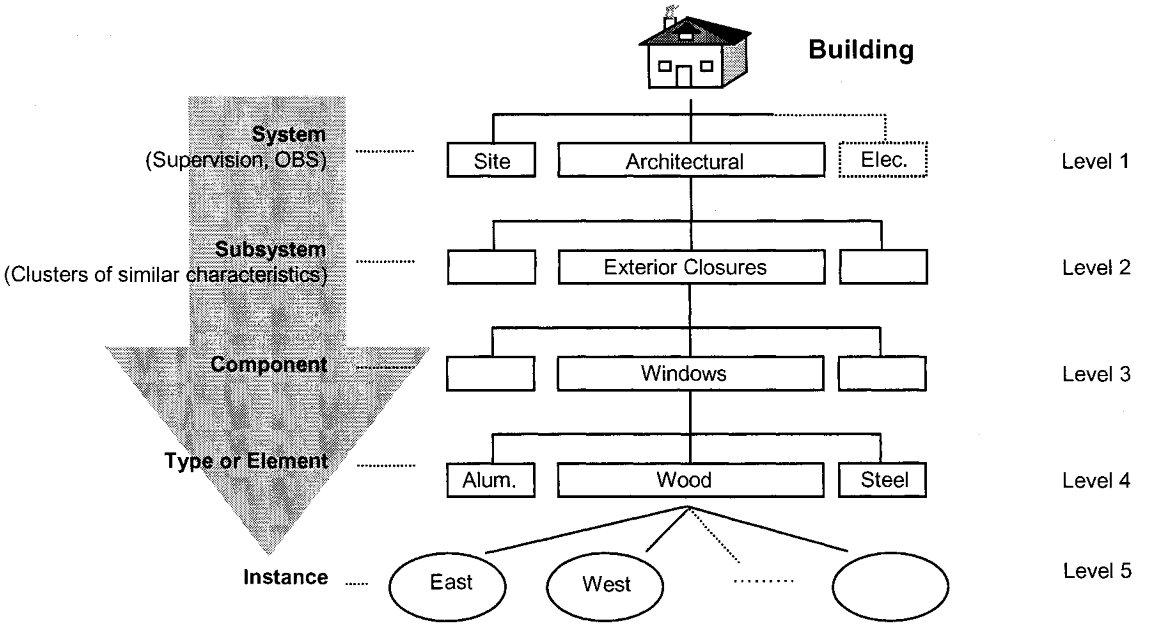


Figure 3.2: Proposed asset hierarchy

The system level in the proposed hierarchy is designed to correspond to the Organizational Breakdown Structure (OBS) of educational organizations (i.e., school boards). The hierarchy, however, can be easily adapted to any other organizational structure. The main benefits of this level is to facilitate the revision process of assessed components, to evaluate the performance of each department in keeping its components in safe and satisfactory condition, and to possibly organize the spending and fund allocation among various systems as an organizational preference.

Under each system, the subsystem level defines clusters of components with same common characteristics. The clustering can be based on material or deterioration mode, etc. For example, under the system "Site Work", all the components that require paving are grouped under one subsystem called "Paved Surfaces", which involves, sidewalks, playgrounds, roads, etc. This level is important because it facilitates the comparison among groups with similar characteristics which can help finding better prioritization preferences.

Underneath the subsystem level, the component level covers all items such as, boiler, roof, window, gym, library, etc. Such level can be easily adapted to suite any asset-specific components.

The fourth level is the type or element level which specifies various sub categories of a component such as wooden windows, steel windows, etc. This level helps making more accurate evaluation criteria for the assessment process. Specific lists of deficiencies will be provided at this level with their reasonable weights to reflect the impact of each deficiency to condition.

The last level in the proposed hierarchy is the instance level which involves the individual physical items that need to be inspected, such as, the "east-side wooden windows in building 1", boiler1, boiler2, etc. This level divides the component into manageable sections (instances) that suit practical fund allocation, particularly when some instances (e.g., some doors in a building) have different conditions. As an example, a school building may have three boilers and it is not practical to replace them all at once. As opposed to that, the boilers are divided into three separated instances (i.e., boiler1, boiler2, and boiler3), which are assessed separately to decide when to repair each one and the type of repair to be applied.

3.2.2 Evaluation Mechanism

As presented in the literature of Chapter 2, two common approaches are possible for evaluating the condition of any instance of a building component: (1) direct condition rating; and (2) rating based on a distress survey. Since the distress survey is more accurate in providing a record of what's wrong with the inspected instances, it is used in this study. This approach, however, requires more details regarding the possible deficiencies that can affect the component and how to evaluate them.

The process of condition assessment using distress survey first defines the common defects (often referred to as deficiencies) that affect each type of a component. The process then calculates a condition index (on a reasonable numeric scale) using a mathematical relationship between the weights of various defects and the inspected severities of these defects.

1. Type-Specific Deficiency Lists: Accurate deficiency lists are key to accurate condition assessment. These lists were determined in two steps as follows:

Step 1: The past condition assessment reports were studied and analyzed. In 2003, the Toronto District School Board (TDSB) in Canada conducted a large condition assessment survey for its schools by experienced investigators. The survey covered about 650 schools and resulted in individual school reports that describe the conditions and expected needs. These reports have been used as the basis for the present study. These reports include similar components at various ages (conditions), in different schools. As such, the reports cover all problems along the life of a component. In the analysis, reported text descriptions regarding the condition of each component were aggregated. Three pieces of information were then extracted: (1) generic deficiencies; (2) possible types of the component; and (3) symptoms of the observed deficiencies. Figure 3.3 illustrates, schematically, this analysis for the "Window" component.

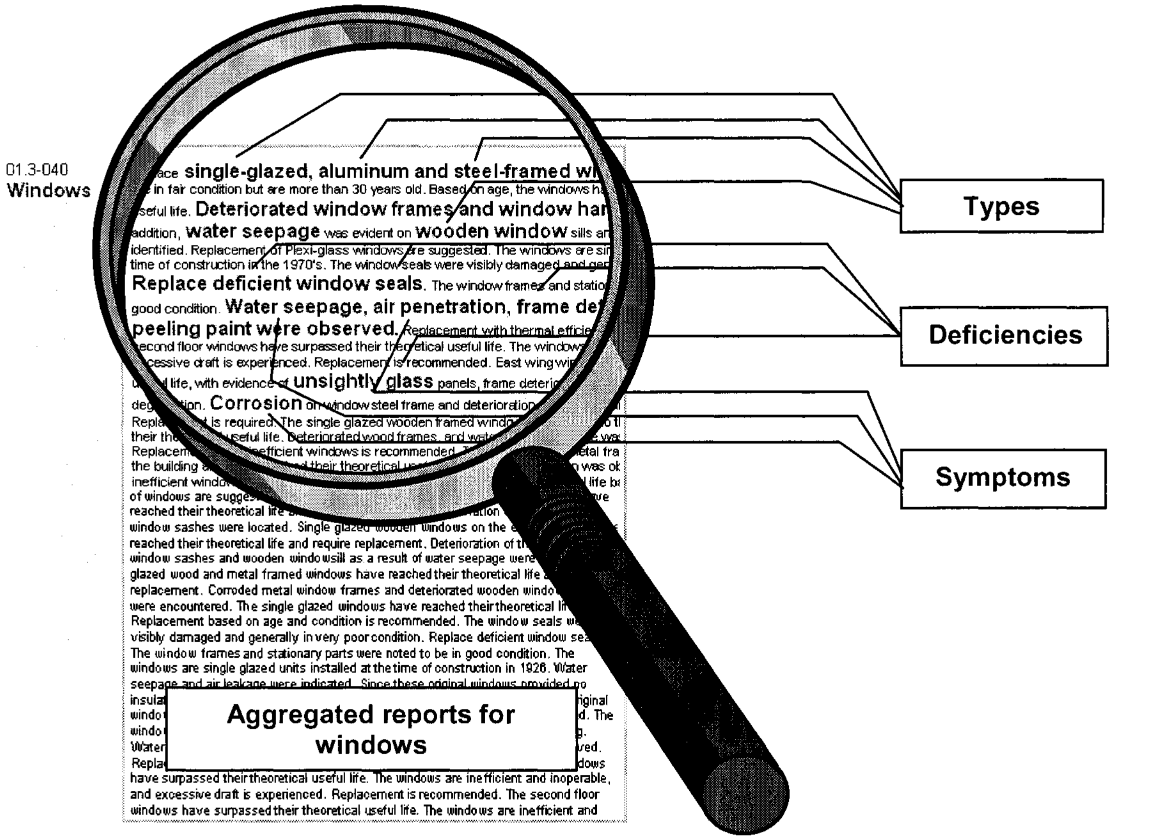

01.3-040
Windows
single-glazed, aluminum and steel-framed
Deteriorated window frames and window
water seepage
wooden window
Replace deficient window seals.
Water seepage, air penetration, frame de
peeling paint were observed.
unsightly glass
Corrosion
Types
Deficiencies
Symptoms
Aggregated reports for windows

Figure 3.3: Process for extracting main information from old reports

As an example of the analysis results, the window component is considered. From the text description of various condition assessment reports, five generic deficiencies for windows along with their possible symptoms were identified as shown in Table 3.1. Also, six types of windows could be formed as the combination of various frame types and number of glazing layers, i.e., single or double glazed with three frame types, steel, wood, and aluminum.

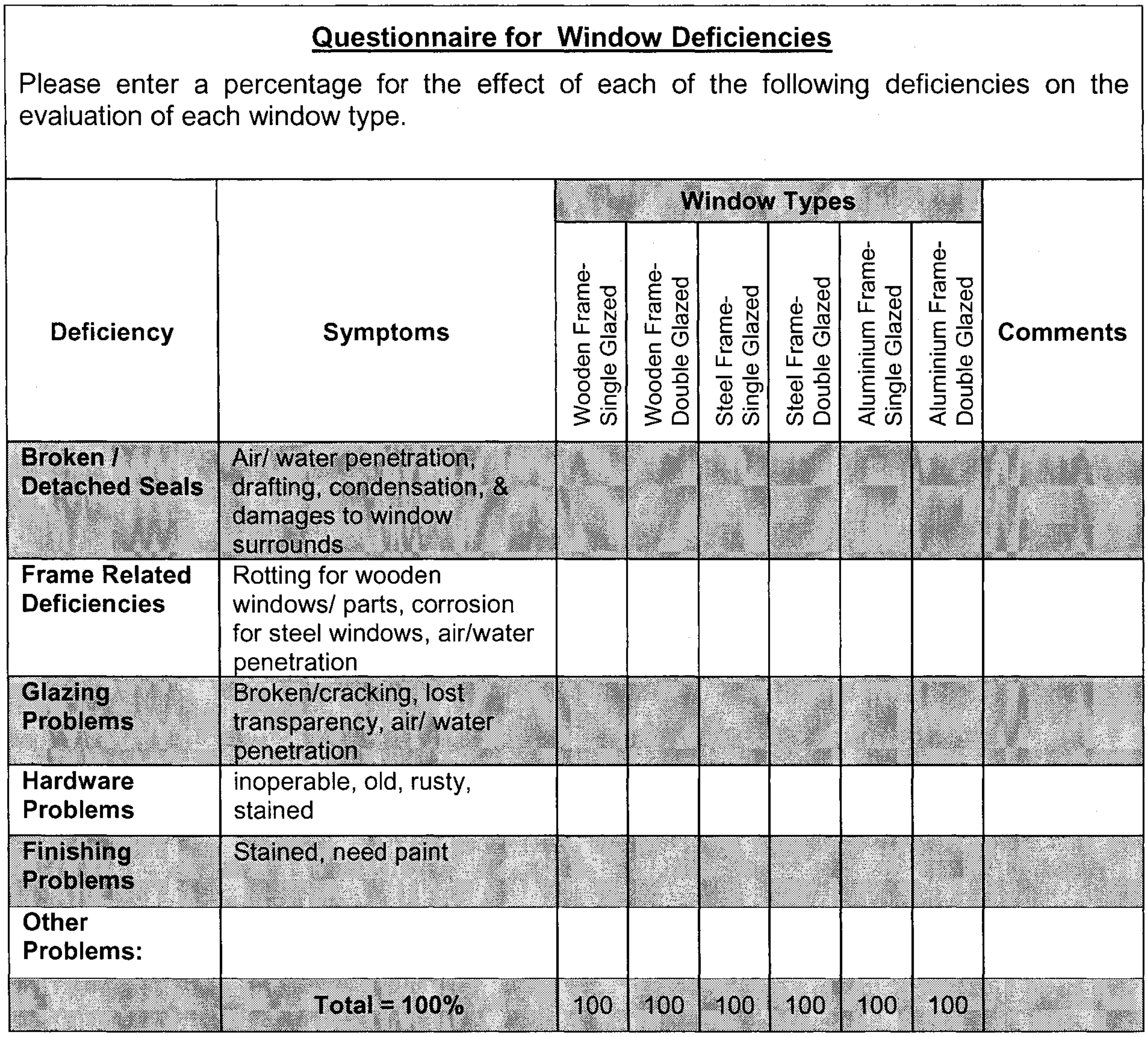
Table 3.1: Window Deficiencies and Symptoms.

Possible Deficiencies	Symptoms
Broken / Detached Seals	Air and water penetration, condensation
Frame Related Deficiencies	Broken, rot or rust, air and water penetration
Glazing Problems	Cracks, broken, lost transparency
Hardware Problems	Rusty, obsolete, inoperable
Finishing Problems	Need paint, wet / dirt surroundings

Step 2: The second step is to refine the general deficiencies through a questionnaire survey. For the window example, the list of deficiencies along with the six possible types were presented to building inspectors and window maintenance personnel at the TDSB through a questionnaire survey. The survey was prepared to determine the weights of each deficiency with respect to the various window types. Figure 3.4 shows the questionnaire survey for the window component. Based on the responses received, the number of glazing layers did not mandate change in the generic deficiency list. As such, only the type of frame (wood, steel, or aluminum) forms the three types of windows that require specific weights or additional deficiencies to be added to the generic list. Accordingly, the average weights of the deficiencies in the three window types are considered, as summarized in Table 3.2.

Table 3.2: Windows' Type-Specific Deficiency Lists.

Possible Deficiencies	Weights for various window types (%)		
	Wood	Steel	Aluminum
Broken / Detached Seals	25	28	38
Frame Related Deficiencies	22	20	25
Glazing Problems	31	27	10
Hardware Problems	7	10	20
Finishing Problems	15	15	7
	□ = 100%	**□ = 100%**	**□ = 100%**



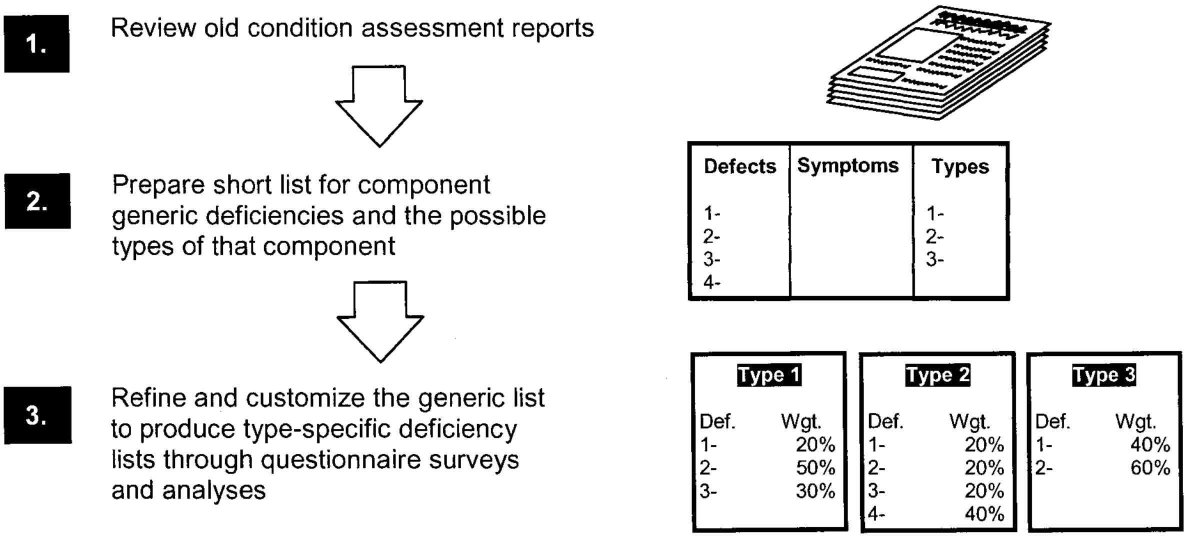
Questionnaire for Window Deficiencies

Please enter a percentage for the effect of each of the following deficiencies on the evaluation of each window type.

Deficiency	Symptoms	Window Types						Comments
		Wooden Frame-Single Glazed	Wooden Frame-Double Glazed	Steel Frame-Single Glazed	Steel Frame-Double Glazed	Aluminium Frame-Single Glazed	Aluminium Frame-Double Glazed	
Broken / Detached Seals	Air/ water penetration, drafting, condensation, & damages to window surrounds							
Frame Related Deficiencies	Rotting for wooden windows/ parts, corrosion for steel windows, air/water penetration							
Glazing Problems	Broken/cracking, lost transparency, air/ water penetration							
Hardware Problems	inoperable, old, rusty, stained							
Finishing Problems	Stained, need paint							
Other Problems:								
	Total = 100%	100	100	100	100	100	100	

Figure 3.4: Questionnaire survey for type-specific deficiencies of windows

The whole procedure to determine the type-specific deficiency lists is summarized in Figure 3.5. It is worth to mention that, more information has also been extracted and analyzed from the old assessment reports as will be discussed later. Once the weights associated with the deficiency list of each type/element are defined, it is used as basis for conducting accurate condition assessment for all instances.



1. Review old condition assessment reports

2. Prepare short list for component generic deficiencies and the possible types of that component

3. Refine and customize the generic list to produce type-specific deficiency lists through questionnaire surveys and analyses

Defects	Symptoms	Types
1- 2- 3- 4-		1- 2- 3-

Type 1	
Def.	Wgt.
1-	20%
2-	50%
3-	30%

Type 2	
Def.	Wgt.
1-	20%
2-	20%
3-	20%
4-	40%

Type 3	
Def.	Wgt.
1-	40%
2-	60%

Figure 3.5: Procedure to determine type-specific deficiencies

2. Condition Representation: To allow consistency among the condition values of various building components, a Deterioration Index (DI) of 100 points is used in this study. The (DI) values indicate the level of component deterioration. The (DI) ranges from 0 to 100 where (DI) of 0 implies minimum deterioration (excellent condition), while a (DI) of 100 implies maximum deterioration or extremely critical condition. The (DI) range is divided into 11 divisions with five linguistic expressions, as shown in Figure 3.6.

Condition	Excellent			Good		Fair		Poor		Critical	
Deterioration Index (DI)	0	10	20	30	40	50	60	70	80	90	100

Figure 3.6: Scale for Deterioration Index

DI Calculation: In the proposed approach, Deterioration Index (DI) is calculated from a given list of defects for a component. Having the list of deficiencies along their weights, a single (DI) value can be calculated for any instance (j) using the following general equation:

$$DI_j = \frac{\sum_{i=1}^{d} W_i \times S_{ij}}{100} \tag{3.1}$$

where, DI$_j$ = Deterioration Index for the (j th) instance;

W$_i$ = Weight for deficiency (i); and

S$_{ij}$ = Inspector judgment of the extent of deficiency (i) (severity) for instance (j).

In a matrix form, Equation (3.1) can be presented as follows:

$$DI_j = \sum^{i} \begin{array}{c} D_1 \\ D_2 \\ \\ D_i \end{array} \begin{bmatrix} w_1 \\ w_2 \\ \bullet \\ w_i \end{bmatrix} \times \begin{bmatrix} S_{1j} \\ S_{2j} \\ \bullet \\ S_{ij} \end{bmatrix} / 100$$

For example, assume a specific instance with three deficiencies D1, D2, and D3 that have weights of 25, 50, and 25, respectively. If the inspector estimates the associated severities as 70, 20, and 40, then the (DI) equals to (25x70 + 50x20 + 25x40) / 100 = 37.5.

3.2.3 Inspection and Data Collection Mechanism

The whole purpose of a condition assessment exercise is to determine accurate deterioration indices (DIs) for all instances of all components in the building being assessed. For a single instance of a component (e.g., boiler B1 in the building), field inspection is essential to visually inspect if any of the related deficiencies exist and to which extent is its severity. Accordingly, the deterioration index (DI) can be calculated using equation (3.1). However, regardless of the management system used, the quality of the assessment largely depends on the process by which the severities are determined, which is a highly subjective and experience-based process. Moreover, the process is costly and time consuming. To support accurate and less subjective assessment, an interactive visual guidance system that suits hand-held devices is presented in the next section. Various guidance systems will be discussed, and then the visual condition assessment program V-CAP will be introduced.

3.3 Visual Guidance Systems

Visual Guidance is presented in this thesis as an approach to minimize the subjectivity associated with the condition assessment process. To design and implement the proposed system, an initial study was carried out by collecting a database of pictures of various building components at different condition. Afterwards, possible system designs were investigated and the most appropriate one was used to implement the visual guidance system. Details are provided in the following subsections.

3.3.1 Preliminary Investigation

The same reports used to determine the type-specific deficiency lists contain along with the text description lots of pictures to validate the assessed conditions. These reports, as such, can be used to collect and extract sample pictures for various components at various condition states. Figure 3.7 shows sample pictures related to various deteriorated windows.

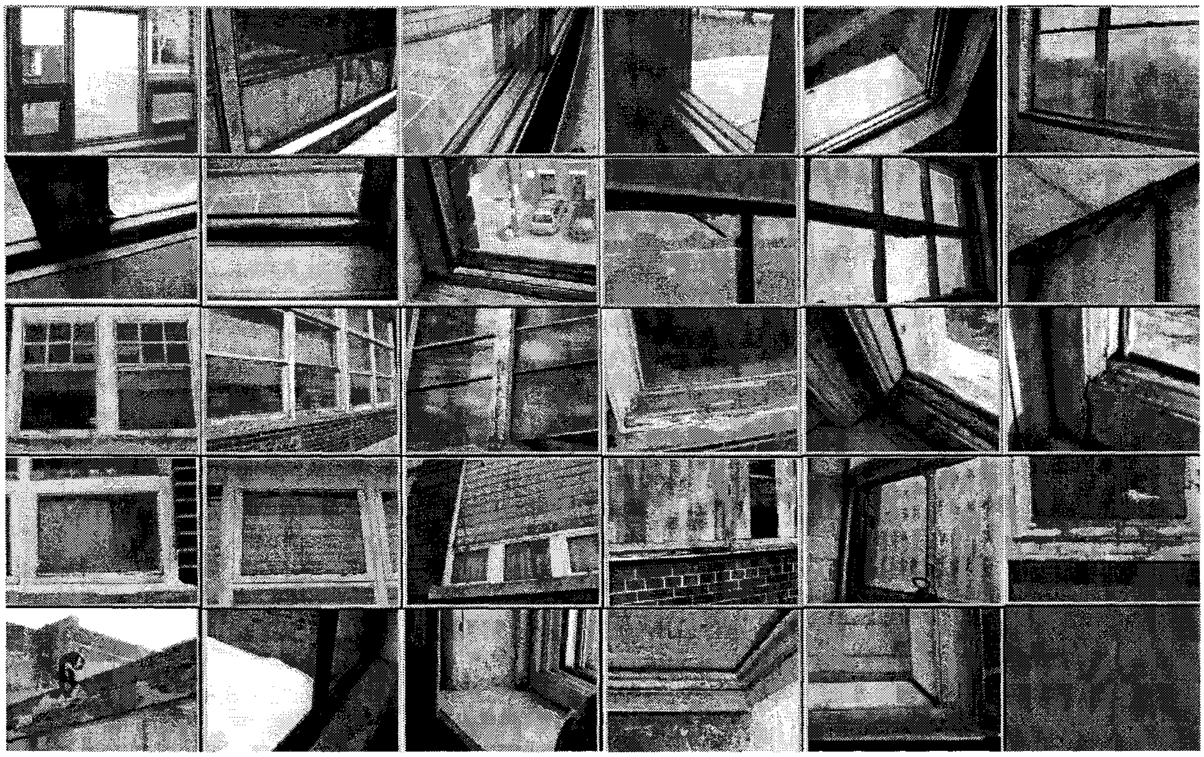


Figure 3.7: Sample pictures for the window component

The pictures in Figure 3.7 show the ability to design guidance systems at various supporting levels. Some general pictures of the component that describes the overall condition can be used to support direct rating. On the other hand, other pictures can be used to describe the deterioration of the component's elements (i.e., assessing the deficiencies' severities).

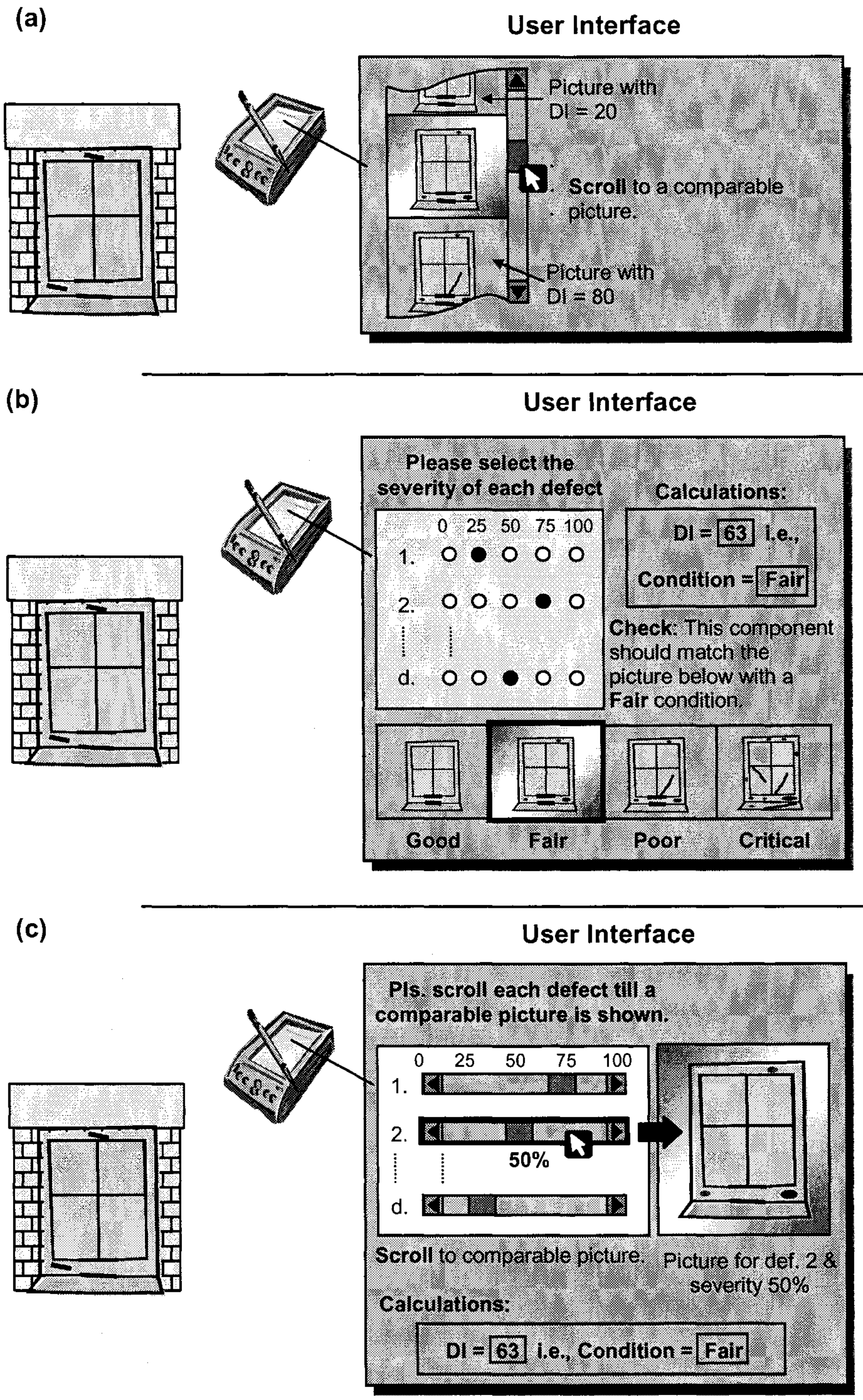
3.3.2 Various Designs for a Visual Guidance System

Figure 3.8 shows three possible designs that use visual guidance with their expected pros and cons. The first approach (Figure 3.8 a) is the easiest and fastest approach, but can not provide details regarding what's wrong with the assessed component. As such, it suits the direct rating condition assessment approach. In this design, the available pictures of a component are pre-ranked and given proper (DI) values (e.g., 10, 20, 30... 100). The proposed design stores the ranked pictures and facilitates seeing them one after the other. During the assessment process, the user is asked to select the picture which closely matches the actual component. Accordingly, the system rates the component similar to the rate of the selected picture. The number of pictures required to build the system is expected to range from 5 to 10 pictures per component.

The second and the third design for visual guidance are appropriate for distress surveys, as they can provide detailed information about the severity extent of the observed deficiencies.

In the second design, four static pictures for each component will be available during the assessment of a similar component. These four pictures show the component in various condition states, good, fair, poor, and critical. The user will be asked to start with a personal judgment to select the severity of each deficiency, accordingly, the condition is automatically calculated and a comparable picture is highlighted (Figure 3.8 b). The user then compares the presented picture with the real component under assessment. Accordingly, the user may confirm or making slight changes to adjust the initial severities.

The third design (Figure 3.8 c) is the most accurate, hence requires more specific pictures to describe the severity of each deficiency. With the scroll of each deficiency, pictures will be presented with various severity levels. The user should stop scrolling the severity level when a comparable picture is presented. The number of pictures to develop such a system varies based on the number of deficiencies; about 25 pictures per component are expected.


(a)
User Interface
Picture with DI = 20
Scroll to a comparable picture.
Picture with DI = 80
(b)
User Interface
Please select the severity of each defect
0 25 50 75 100
1. ○ ● ○ ○ ○
2. ○ ○ ○ ● ○
d. ○ ○ ● ○ ○
Calculations:
DI = 63 i.e.,
Condition = Fair
Check: This component should match the picture below with a Fair condition.
Good
Fair
Poor
Critical
(c)
User Interface
Pls. scroll each defect till a comparable picture is shown.
0 25 50 75 100
1.
2.
50%
d.
Scroll to comparable picture.
Picture for def. 2 & severity 50%
Calculations:
DI = 63 i.e., Condition = Fair

Comments (a)

- Pictures with DI=10, 20... 100 are stored. The component is matched to the closest picture; e.g., picture for DI = 60.
- Fast, simple, & accurate.
- Requires 5 to 10 pictures per component (based on scale).
- Direct condition assessment without details about deficiencies.

Comments (b)

- Four pictures are stored for each component at different condition states. The pictures confirm the appropriateness of the user specified severities.
- Requires 4 pictures per component.
- Provides detailed deficiency evaluation.
- More Accurate.
- Requires personal judgment to select the severities.

Comments (c)

- Picture changes with user selection of each deficiency and each severity level.
- Requires many pictures for each component (about 25).
- Provides accurate and detailed deficiency evaluation.
- Fast and most accurate.
- Personal judgment is almost eliminated.

Figure 3.8: Three possible designs for on-site visual guidance

3.4 Condition at Higher Levels in the Asset Hierarchy

Rolling up the condition means determining the condition at higher levels (e.g., component, subsystem, system, building, or network level) in the asset hierarchy from the calculated condition at the lowest level (instance level). Therefore, a mechanism for aggregating the (DIs) at a parent level from all its children becomes necessary. To do that, it is necessary to define proper weights among the various children to reflect their relative contribution to the overall condition of the parent. Aggregating the condition from the instance to the component level is simple and straightforward. Assume for example that a building has a window component with four instances (north, south, west, and east), with the number of "identically sized" windows for each instance being 10, 15, 10, and 20, respectively, and the (DIs) of these instances being 35, 40, 45, and 30, respectively. Then the relative weight of the four instances is 18.2%, 27.3%, 18.2%, and 36.3%, respectively as determined from the percentage of windows within the instance. As such, the window component condition (i.e., deterioration index) of that specific building can be calculated as: 0.182x35 + 0.273x40 + 0.182x45 + 0.363x30 = 36.36, which is represented mathematically as follows:

$$CDI_c = \frac{\sum_j (Z_{jc} \times DI_{jc})}{\sum_j Z_{jc}} \qquad \forall\ j \in component\ (c) \qquad (3.2)$$

where, (CDI_c) represents the aggregated (DI) of parent component (c); (DI_{jc}) is the deterioration index of instance (j); and (Z_{jc}) is the percentage of instance (j).

Aggregating the condition at levels higher than the component level, however, can be challenging and requires accurate weights (importance factors) to be determined across various components. To illustrate the challenging calculations, assume, for simplicity, that a system level includes only two components: a roof component; and a wall covering component. Considering the consequences of a bad condition, the roof component can logically be classified as more important than wall covering, for the following reasons: (1) in case of roof failure, more components may fail (e.g., ceiling, paint, etc.) than in the case of wall covering failure; and (2) the building may be fully / partially shutdown for repair / safety reasons, which is not the case in the failure of wall covering. As such, determining proper values for the relative importance is essential to condition aggregation, as discussed in the next paragraph. However, knowing the relative importance factors (RIF), the deterioration index at any level higher than the component level (i.e., subsystem, system, building, and network) can be calculated from the components' (DIs), using the following general equation:

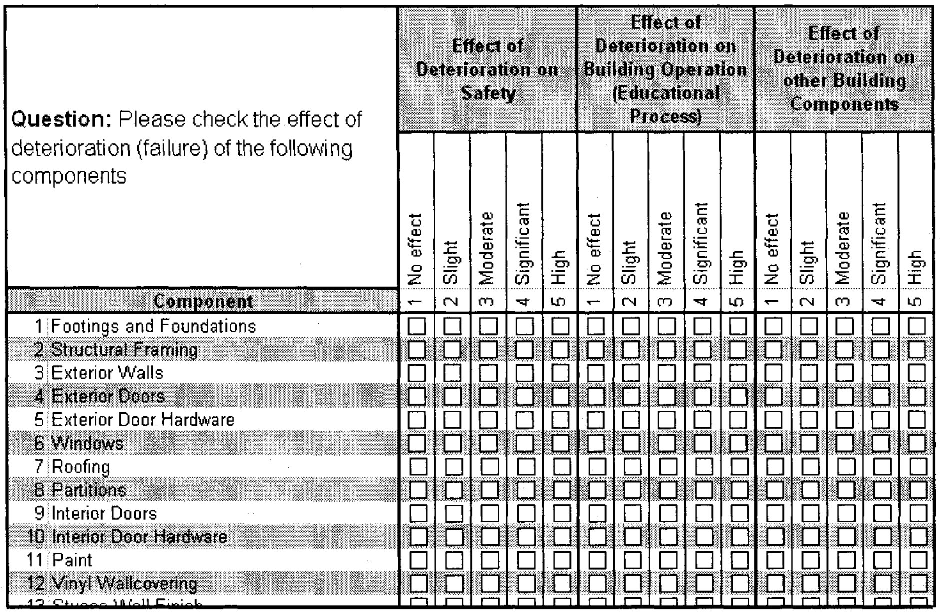
$$DI_{Parent} = \frac{\sum_{child}\left(RIF_{child} \times CDI_{child}\right)}{\sum_{child} RIF_{child}} \qquad \forall \quad child \in Parent \qquad (3.3)$$

where, (CDI_{child}) and (RIF_{child}) are the (DI) and relative importance of each component (only component) that is a child to the desired parent. The (CDI_{child}) values are determined from Equation (3.2).

It is important to note that, the discussed approach can roll up the condition (DI) at any level in the hierarchy as a function of the instances' conditions, relative sizes, and relative importance. As such, the instances with small size will have limited impact on the overall condition at the network level. While this is acceptable for representing the condition, it might prevent those small size instances from receiving repair funds. Therefore, for fund allocation, a minor modification will be made to condition calculation so that the condition and importance not the size of an instance are used for fund allocation, as will be discussed in Chapter 5.

Components' Relative Importance Factors: The relative importance factor for a component is determined as a value ranging from 0 to 100, where the value of 100 implies high importance. To determine this value, the impacts of the component's bad condition (failure) on three main parameters (decided after discussions with the administrators at the TDSB) are evaluated. These three parameters are safety, building performance, and effect on other components. Accordingly, an electronic questionnaire survey was prepared to solicit the opinion of repair and maintenance personnel at the TDSB on the relative impact of component's failure on safety, building operation, and other components. Part of the survey is shown in Figure 3.9 and sample results are shown in Table 3.3 for Architecture/Structural components. The results were calculated based on 14 responses (on average) for 143 individual building components.

It is noted that the relative importance is calculated based on equal weights for the three parameters. Adding more parameters or changing the weights among the three parameters can be made custom based on the organizational need.



Question: Please check the effect of deterioration (failure) of the following components	Effect of Deterioration on Safety					Effect of Deterioration on Building Operation (Educational Process)					Effect of Deterioration on other Building Components				
	1 No effect	2 Slight	3 Moderate	4 Significant	5 High	1 No effect	2 Slight	3 Moderate	4 Significant	5 High	1 No effect	2 Slight	3 Moderate	4 Significant	5 High
Component															
1 Footings and Foundations	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
2 Structural Framing	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
3 Exterior Walls	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
4 Exterior Doors	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
5 Exterior Door Hardware	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
6 Windows	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
7 Roofing	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
8 Partitions	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
9 Interior Doors	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
10 Interior Door Hardware	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
11 Paint	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
12 Vinyl Wallcovering	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
13 Stucco Wall Finish	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

Figure 3.9: Questionnaire survey for relative importance

Table 3.3: Relative importance (sample results)

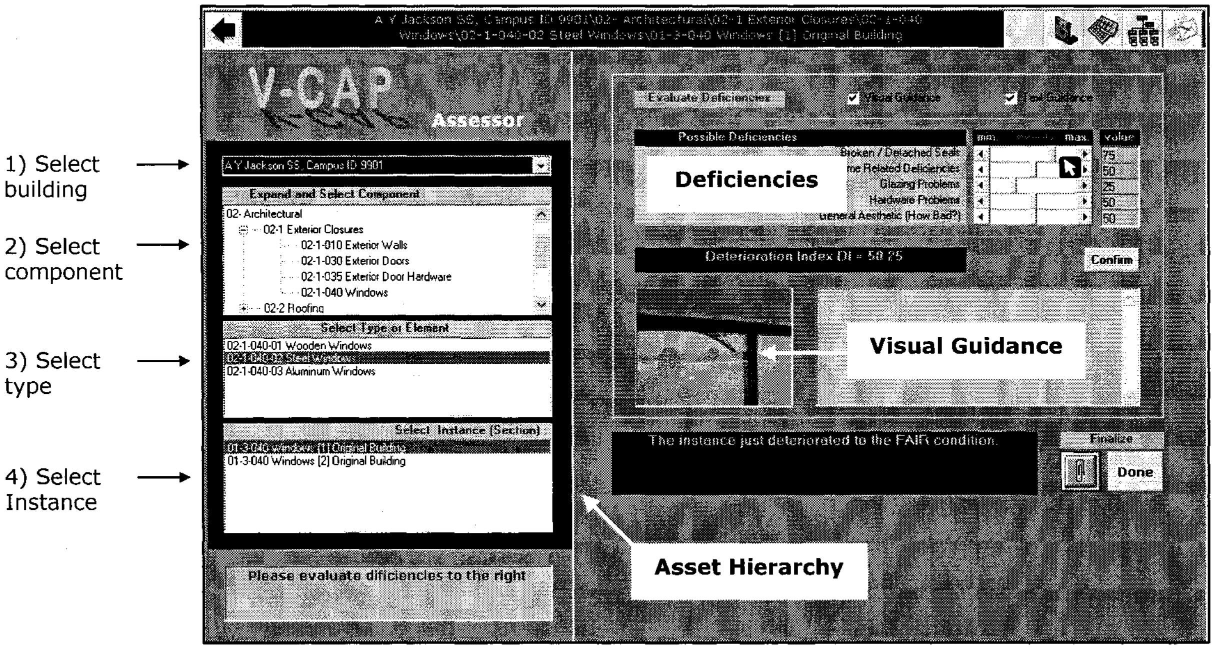
Component	Relative Importance
Footings and Foundations	69
Structural Framing	79
Exterior Walls	68
Exterior Doors	38
Roofing	70
Partitions	40

3.5 Implementation: Visual Condition Assessment Program (V-CAP)

Visual Condition Assessment Program or V-CAP is a computerized system which facilitates and manages the process of condition assessment. V-CAP is developed based on the design of Figure 3.8

(c), which suits the distress survey rating approach. In order to generalize the use of the program, V-CAP is developed in a way to load external data files, covering any 5-level asset hierarchy, list of assets, deficiencies, and any guidance pictures. The system as such can be adapted for any organization or asset type. The use and detailed features are provided in Chapter 6. However, this section outlines the structure for the developed system and the outputs, in addition to, how to better manage the whole process and make it faster and less expensive. The inspection tool of the developed system is shown in Figure 3.10.

As shown in Figure 3.10, V-CAP is designed with an easy-to-use interface which facilitates the selection of instances for inspection and the evaluation of deficiencies. The visual guidance system gives the user the opportunity to visually compare the pictures provided and the real condition of the component under assessment in order to make accurate assessments, in a fast and simple manner.



A Y Jackson SS, Campus ID 9901\02- Architectural\02-1 Exterior Closures\02-1-040 Windows\02-1-040-02 Steel Windows\01-3-040 Windows [1] Original Building

A Y Jackson SS, Campus ID 9901

02- Architectural
02-1 Exterior Closures
02-1-010 Exterior Walls
02-1-030 Exterior Doors
02-1-035 Exterior Door Hardware
02-1-040 Windows
02-2 Roofing

02-1-040-01 Wooden Windows
02-1-040-02 Steel Windows
02-1-040-03 Aluminum Windows

01-3-040 Windows [1] Original Building
01-3-040 Windows [2] Original Building

Possible Deficiencies	Value
Broken / Detached Seals	75
me Related Deficiencies	50
Glazing Problems	25
Hardware Problems	50
General Aesthetic (How Bad?)	50

Deterioration Index DI = 50.25

The instance just deteriorated to the FAIR condition.

Please evaluate dificiencies to the right

Figure 3.10: V-CAP inspection tool

V-CAP is developed on a spreadsheet platform in which various sheets were used to tabulate various pieces of information. The various worksheets are implemented in a VB executable file. The external data files (as mentioned earlier for the hierarchy, list of assets, etc) are loaded and organized in these sheets then linked together through VB codes. The various sheets are as follows:

Asset Sheet: One sheet is used to list the assets and their relevant information. Sample information is, the asset ID, asset name, address, type of building, educational area, construction year, historical value, etc.

Hierarchy Sheets: The five-level hierarchy is divided into three parts and allocated in three different sheets. Part 1 includes the first three levels (system, subsystem, and component) which are general levels regardless the involved assets. Part 2 is also a general level and includes various Types / Elements of various components. The two parts are linked by the component level. Part 3 is the instance level which is an asset-dependent level, as each asset has its own custom number of instances. This part is stored externally (each asset has one data file). The data file related to certain asset is loaded once that asset is selected for inspection during the inspection process and allocated a separate sheet. The instance sheet also shows all relevant information for the individual instances, such as, the last replacement year, location, etc. The link between the Type/Element sheet and the Instance sheet is established through the type level.

Deficiency Sheet: This sheet is used to store all possible deficiencies with their specific weights for all types/elements of all components. Once the instance is selected, this sheet is responsible for providing the user with the proper list of related deficiencies for severity evaluation and (DI) calculation.

Guidance Sheet: One sheet is used for visual guidance which holds for all components links to load the proper picture from the picture database (external source) and another sheet is used to load some text guidance that describes the symptoms of various deficiencies.

Relative Importance Sheet: This sheet stores for all components their relative importance factors.

3.5.1 Managing the Assessment Process Using V-CAP

With the simple-to-use and visual guidance of V-CAP, the system can be used by less experienced personnel. Thus making it possible to conduct the condition assessment (survey) in a parallel fashion, where local facility staff can help assess their assets. Such decentralization in condition assessment has its pros and cons. In term of advantages, it shortens the assessment time and cost by using parallel assessments at reasonably cheaper costs. On the other hand, the main disadvantage is how to ensure the relative quality of the results and that an inspector did the assessment according to specs. This, however, is controlled in V-CAP by adding some revision features to track inspectors' performances.

V-CAP is simply a computer program; it can be installed in more than one portable device (Tablet PC). Once inspectors finish the inspection, results can be transferred and analyzed centrally in the main office. The V-CAP system, as such, includes a server-side (V-CAP main) as one part of the whole asset management framework, and many local copies (V-CAP locals), as client-side versions for inspectors with hand-held devices. Exchanging the information between the client-side and the server-side is proposed to be via e-mail or by saving the assessment results in a shared folder. V-CAP main compiles the assessments received from local stations in one file then allows revisions and reporting (Figure 3.11).

V-CAP Features: The main features provided in V-CAP to support the management of the condition assessment process are summarized in the following points (more details are provided in Chapter 6):

1. Three main functions (inspection tool, revision tool, and reporting tool);
2. Usable on Tablet PCs to facilitate the process and make data entry faster with minimum errors;
3. Dynamic text and visual guidance; and
4. E-mail, camera, and editing features to link client and server, capture real pictures for the assessed component, and add comments and text descriptions for the assessed component.

It is noted that, the previous features relate to V-CAP main, however, the client version has the same features except the revision and reporting features.

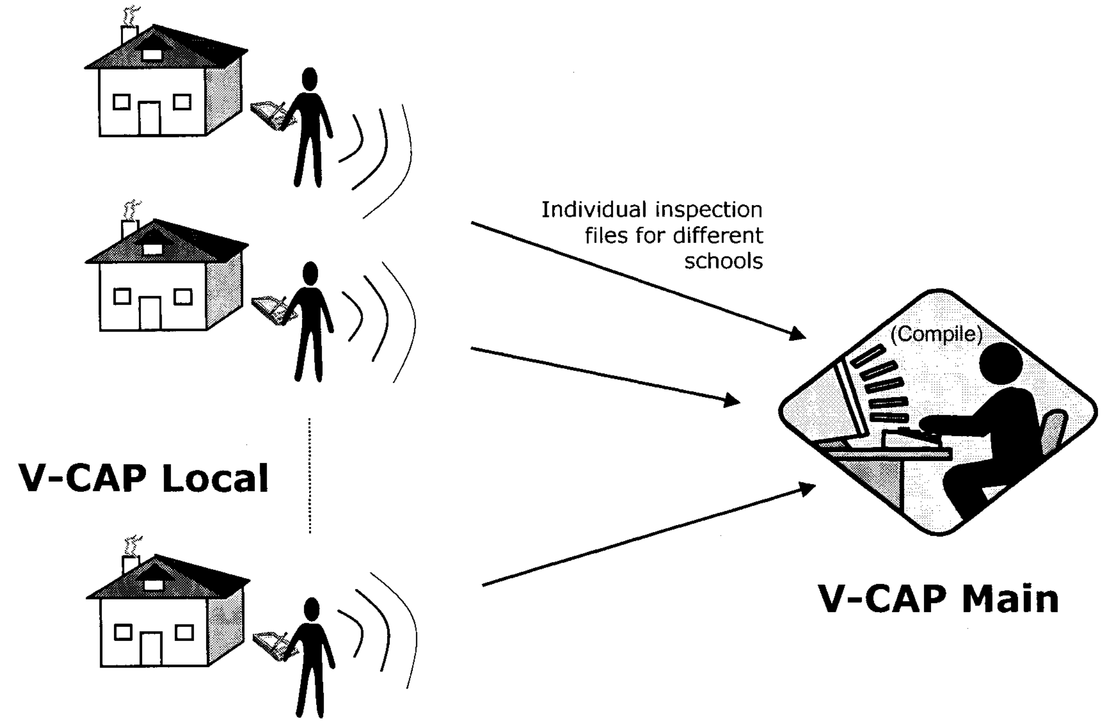
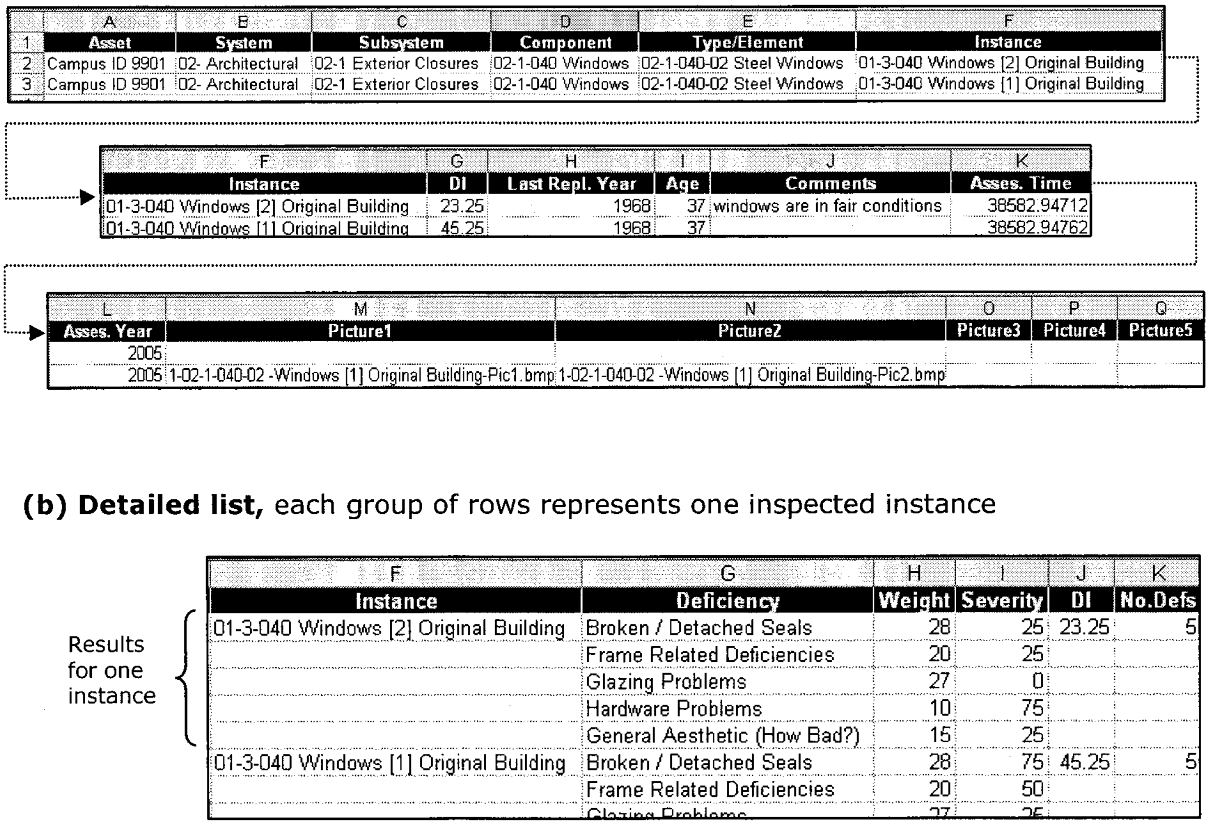


Figure 3.11: Inspection process

3.5.2 V-CAP Outputs

V-CAP produces two outputs: a summary list of the DIs of inspected instances and a detailed deficiency list including the evaluations of deficiencies' severities. The summary list includes the following fields: asset name, full hierarchy, last replacement year, age at inspection time, DI value, assessment time, comments, inspection year, and links to the captured pictures. The detailed list, on the other hand, includes in addition to the items in the summary list, the number of deficiencies, deficiencies' names, deficiencies' weights, and deficiencies' estimated severities. V-CAP output is saved as an external data file which can be edited or augmented and/or sent to the main office for revision and approval. Figure 3.12 shows the V-CAP outputs in a spreadsheet format with illustrative values, the upper part represents the summary list and the bottom part shows the detailed list.



(a) Summary list, each row represents one inspected instance

	A	B	C	D	E	F
1	Asset	System	Subsystem	Component	Type/Element	Instance
2	Campus ID 9901	02- Architectural	02-1 Exterior Closures	02-1-040 Windows	02-1-040-02 Steel Windows	01-3-040 Windows [2] Original Building
3	Campus ID 9901	02- Architectural	02-1 Exterior Closures	02-1-040 Windows	02-1-040-02 Steel Windows	01-3-040 Windows [1] Original Building

F	G	H	I	J	K
Instance	DI	Last Repl. Year	Age	Comments	Asses. Time
01-3-040 Windows [2] Original Building	23.25	1968	37	windows are in fair conditions	38582.94712
01-3-040 Windows [1] Original Building	45.25	1968	37		38582.94762

L	M	N	O	P	Q
Asses. Year	Picture1	Picture2	Picture3	Picture4	Picture5
2005					
2005	1-02-1-040-02 -Windows [1] Original Building-Pic1.bmp	1-02-1-040-02 -Windows [1] Original Building-Pic2.bmp			

(b) Detailed list, each group of rows represents one inspected instance

F	G	H	I	J	K
Instance	Deficiency	Weight	Severity	DI	No.Defs
01-3-040 Windows [2] Original Building	Broken / Detached Seals	28	25	23.25	5
	Frame Related Deficiencies	20	25		
	Glazing Problems	27	0		
	Hardware Problems	10	75		
	General Aesthetic (How Bad?)	15	25		
01-3-040 Windows [1] Original Building	Broken / Detached Seals	28	75	45.25	5
	Frame Related Deficiencies	20	50		
	Glazing Problems	27	25		

Figure 3.12: V-CAP outputs

From the outputs of V-CAP (condition assessment), life cycle cost analysis can start at two management levels: (1) the project level (instance) to determine the minimum repair costs to compensate for deterioration; and (2) the network level (all instances) to optimize yearly available budgets and to maximize the overall network condition. These two analyses will be covered in Chapters 4 and 5, respectively (Figure 3.13).

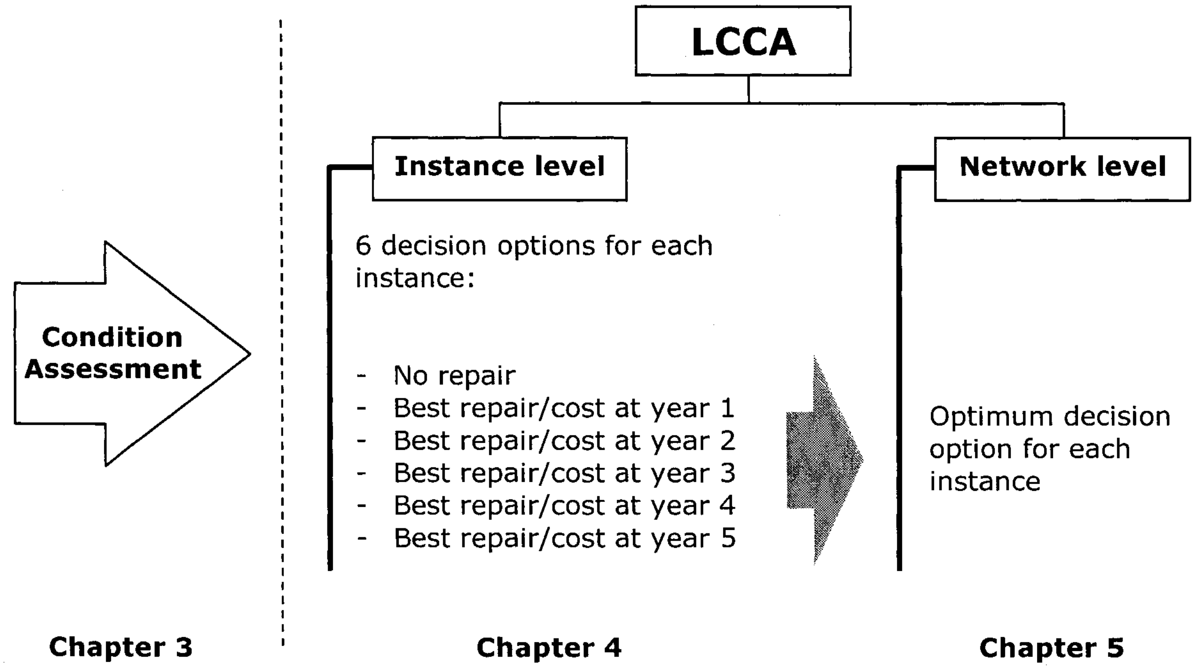

LCCA
Instance level
Network level
Condition Assessment
6 decision options for each instance:
- No repair
- Best repair/cost at year 1
- Best repair/cost at year 2
- Best repair/cost at year 3
- Best repair/cost at year 4
- Best repair/cost at year 5
Optimum decision option for each instance
Chapter 3
Chapter 4
Chapter 5

Figure 3.13: LCCA at both project-level and network-level

3.6 Conclusions

In this chapter a system is presented to enhance the process of visual inspection. The presented system is easy to use by less experienced individuals, thus making it possible for local personnel at facilities sites to conduct the condition assessment simultaneously to save time and cost. To design the system, past pictures of components at various conditions, were captured from previous assessment reports. The pictures were then used to build a visual database that interacts with the user during the assessment process and provides less subjective guidance for assessing components, in speedy manner. Various designs of the visual guidance system are investigated in terms of accuracy and easy of use. Accordingly, a Visual Condition Assessment Program (V-CAP) was implemented, as one of the main modules in the proposed asset management framework.

Chapter 4

LIFE-CYCLE-COST MODEL FOR BUILDING COMPONENTS

4.1 Introduction

Chapter 3 discussed an approach to facilitate field inspection and condition assessment of buildings. This approach determines the current condition of building components, represented by deterioration indices. To translate these deterioration indices into specific repair strategies for the deteriorated components, this chapter presents developments in the following two models that represent a detailed life cycle cost analysis for each instance (Figure 4.1):

1. A model to predict future deterioration of each component under any repair/no-repair decision; and
2. A model to investigate alternative repair actions and their effect on both cost and ability to recover the component deficiencies.

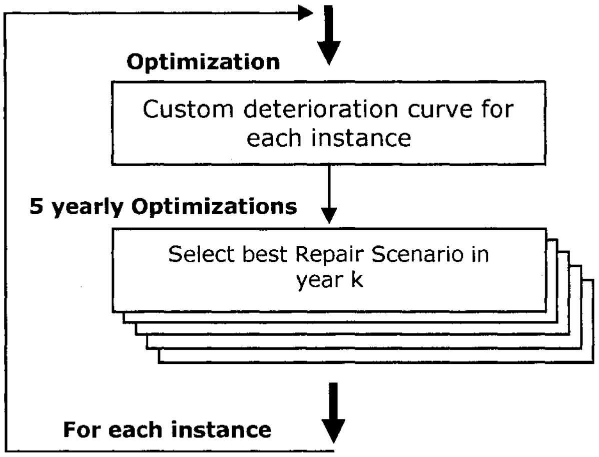

Optimization
Custom deterioration curve for each instance
5 yearly Optimizations
Select best Repair Scenario in year k
For each instance

Figure 4.1: Instance-level optimization

The two models facilitate the life-cycle cost analysis of building components to determine the least-cost repair scenario for each year of the planning horizon. Details on the developments of the two models are presented in the following sections.

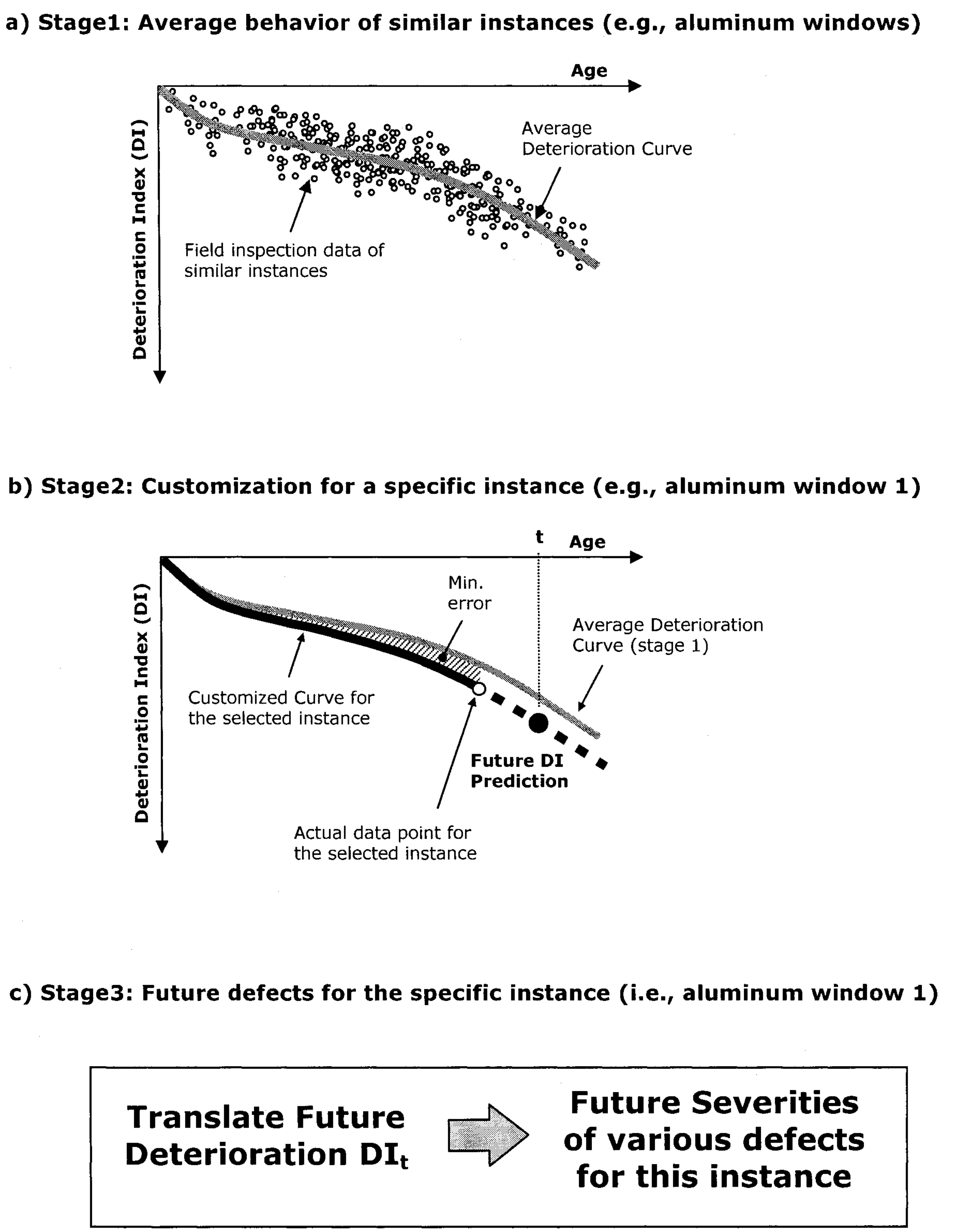
4.2 Future Deterioration of Building Components: A Three-Stage Prediction Model

Predicting the future condition of asset components is a crucial part in a good asset management system. Deterioration modeling, however, is not a simple task, particularly due to: (1) the unforeseen parameters (other than age) that may affect the deterioration of a component; (2) the serious lack in historical data even for known parameters that affect component deterioration; and (3) the variability in deterioration behavior even among similar components. To account for the previous challenges and to improve the accuracy of deterioration prediction, therefore, this study deals with each single instance and predicts its custom deterioration behavior through three stages (Figure 4.2):

1. Generate for each component type (e.g., aluminum windows) an average deterioration curve from condition assessment data;
2. Generate for each instance (e.g., aluminum window 1: east side) a deterioration curve that is the same as the average deterioration curve, but customized to specific data that relate to that instance; and
3. Utilize the custom deterioration curve for the instance to predict its future deterioration indices (DIs). From these future (DIs), expected future defects (problems that might need future repair) are then predicted.

4.2.1 Stage1: Average Deterioration Curve

In order to predict an average deterioration behavior for a certain type of component (e.g., aluminum windows), the results of a completed inspection and assessment survey is used. In case of large organizations, such as the TDSB which manages about 650 schools, an assessment survey is expected to cover a wide spectrum of assets with various components and types. Consequently, the results of such a survey will provide measurements for the deterioration indices (DIs) for thousands of instances. Similar instances at different schools have various ages at the inspection time. Accordingly, many data points can be plotted with age and associated DI, as shown in Figure 4.2 (a).



a) Stage1: Average behavior of similar instances (e.g., aluminum windows)

b) Stage2: Customization for a specific instance (e.g., aluminum window 1)

c) Stage3: Future defects for the specific instance (i.e., aluminum window 1)

Translate Future Deterioration DI_t

Future Severities of various defects for this instance

Figure 4.2: Three – stage deterioration model

For example, if 200 schools out of the 650 schools of the TDSB have "aluminum windows", and each school has four instances (i.e., east, west, north, and south), the total number of inspected aluminum windows is 800 instances. At the inspection year, where all (DIs) are measured, each instance has its own age, which represents the number of years from its last replacement year (or construction year if not replaced). Once all points are plotted, an average deterioration curve (i.e., Age versus DI) for this type of instances can be plotted using regression or simple averaging, as shown in Figure 4.2 (a). The process is then repeated for all types of components.

4.2.2 Stage2: Instance Custom Deterioration

In order to adjust the average deterioration curve to account for the deterioration behavior of a specific instance, the Markov chains approach is used. This approach depends on what is called the transition probability matrix [TPM], which shows the probabilities of moving from one condition state to another. With no [TPMs] available in the literature for building components, it is important, therefore, to generate these matrices, customized to a specific instance using optimization. The result of the optimized Markov chain process is a customized deterioration curve similar to the black curve in Figure 4.2 (b). Markov chains process and its customization are discussed as follows:

Markovian Predictions: This approach predicts the deterioration of a component by accumulating its probability of transition from one condition state to another over discrete time intervals. The approach is covered in detail in Chapter 2, however, a summary of the steps to predict the condition at any time (t) is as follows:

1. Define the number of condition states (n) that can describe the deterioration and arrange the possible states [PS] in a column vector. As an example, a five-state vector (from 1 to 5) is used, as follows:

$$[PS] = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} \begin{matrix} \textit{Best} \\ \\ \\ \\ \textit{Worst} \end{matrix}$$

2. Define the transition probability matrix [TPM] with values that represent the probabilities to move from one condition state to another. Since the [PS] has five states, the [TPM] is a 5x5 square matrix, let's assume for demonstration it is known with values as shown:

$$[TPM] = \begin{array}{c|ccccc} \text{State} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0.6 & 0.4 & 0 & 0 & 0 \\ 2 & 0 & 0.3 & 0.7 & 0 & 0 \\ 3 & 0 & 0 & 0.5 & 0.5 & 0 \\ 4 & 0 & 0 & 0 & 0.4 & 0.6 \\ 5 & 0 & 0 & 0 & 0 & 1 \end{array}$$

For example, the values in the second row show a 30% probability to stay in the same state (state2) and a 70% probability to deteriorate to the next state (state3). Also, as noticed, it is not allowed to deteriorate more than one condition state within the same time period, i.e., either to stay in the same condition state or to deteriorate to the immediate next state, other states have zero probabilities.

3. Define the initial probability row vector [IP_o], as follows:

$$\begin{array}{lccccc} \text{State} & 1 & 2 & 3 & 4 & 5 \\ [\text{IP}_o] = & [1.0 & 0 & 0 & 0 & 0] \end{array}$$

The initial probability row vector (at time 0) shows a condition of 100% being in state 1 and 0% in the rest of states from 2 to 5.

4. Calculate the future probability row vector at any time [FP_t]. The condition after (t) years from a known condition (e.g., the IP_o) can be calculated by multiplying the condition vector at the known condition by the [TPM] raised to the power (t), as follows:

$$[FP_t]_{1\times n} = [IP_o]_{1\times n} \cdot [TPM]^t_{n\times n}$$

For example, at (t = 2) the expected [FP_2] vector is,

$$[FP_2] = [1 \quad 0 \quad 0 \quad 0 \quad 0] . \begin{bmatrix} 0.6 & 0.4 & 0 & 0 & 0 \\ 0 & 0.3 & 0.7 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^2 = [0.36 \quad 0.36 \quad 0.28 \quad 0 \quad 0]$$

The results indicates that after two years from the known condition, the condition vector has the following probabilities, 36% for state 1 and 2, and 28% for state 3, then 0% for state 4 and 5.

5. Calculate a single value for the condition (i.e., the future state value FS_t). The final step is to come up with one crisp value to describe the condition. This is obtained by multiplying the future probability row vector $[FP_t]$ by the possible states column vector [PS].

$$FS_{t_{1\times 1}} = [FP_t]_{1\times n} \cdot [PS]_{n\times 1}$$

For the same example, the future state value (FS_2) is calculated to be 1.92, which indicates closeness to the second state, as follows:

$$FS_2 = \begin{bmatrix} 0.36 & 0.36 & 0.28 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} = 1.92$$

Since the above process predicts the condition state at any time (t), it can be repeated with various (t) values (1 to 50, for example) to draw a deterioration behavior for 50 years. As explained, all calculations depend on matrix multiplication, which suits spreadsheet platforms. Hence, a spreadsheet model for Markov has been developed.

Spreadsheet Modeling for Markov Chains: As described in the Markov chain process, the [TPM] is the core of prediction. A spreadsheet model for Markov chain calculations has been structured, incorporating all the formulation on spreadsheet cells. The general layout of the model is shown in Figure 4.3. On the upper left part, a [TPM] is located which is multiplied in a sequential manner for the sake of raising it to different powers, from 1 to 50, as shown to the left. The left part, as such, contains $[TPM]^1$, $[TPM]^2$, $[TPM]^3$ till $[TPM]^{50}$. The future probability vector $[FP_t]$ at any age (t) is calculated by multiplying the $[IP_o]$ by the corresponding $[TPM]^t$. All $[FP_ts]$ are calculated in the middle part of the sheet. Finally, a single value for the condition is calculated for each (t) by multiplying the calculated $[FP_t]$ by the column vector [ST]; (FS_t) results are located in the right part of the sheet.

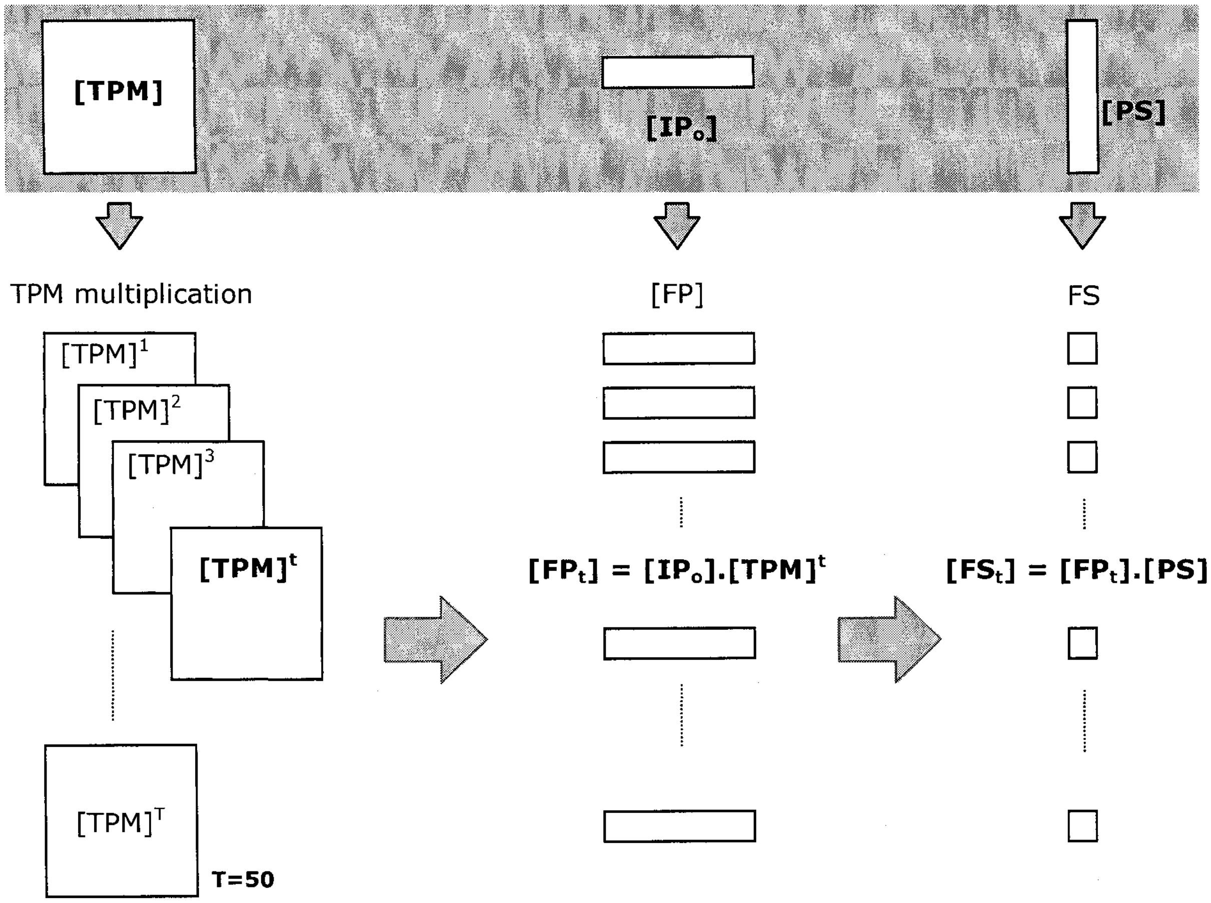


Figure 4.3: General layout for the Markov Chains spreadsheet

In order to implement the spreadsheet format for building components, the following steps were followed:

1. **Possible state vector [PS]**

An 11-state vector ranging from 0 to 10 is used to provide reasonable states for deterioration. To convert between the 11-state scale and the original (DI) scale, the results from Markov is multiplied by ten (i.e., $DI_t = FS_t \times 10$). The possible states vector [PS] is as follows:

$$[PS] = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{bmatrix} \tag{4.1}$$

2. Initial transition probability matrix [TPM]

The [TPM] is an 11x11 matrix that holds values between 0 and 1. The values occupy only the diagonal and next-to-diagonal cells (i.e., the $p_{i,\,i}$ and $p_{i,\,i+1}$), and the rest are 0s. As the summation of probabilities for each row equals to 1, then the value next to the diagonal ($p_{i,\,i+1}$) is equal to ($1 - p_{i,\,i}$), as illustrated.

$$[TPM] = \begin{bmatrix} \mathbf{0.9} & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{0.85} & 0.15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{0.7} & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{0.75} & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{0.6} & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{0.5} & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0.4} & 0.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0.3} & 0.7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0.25} & 0.75 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0.15} & 0.85 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

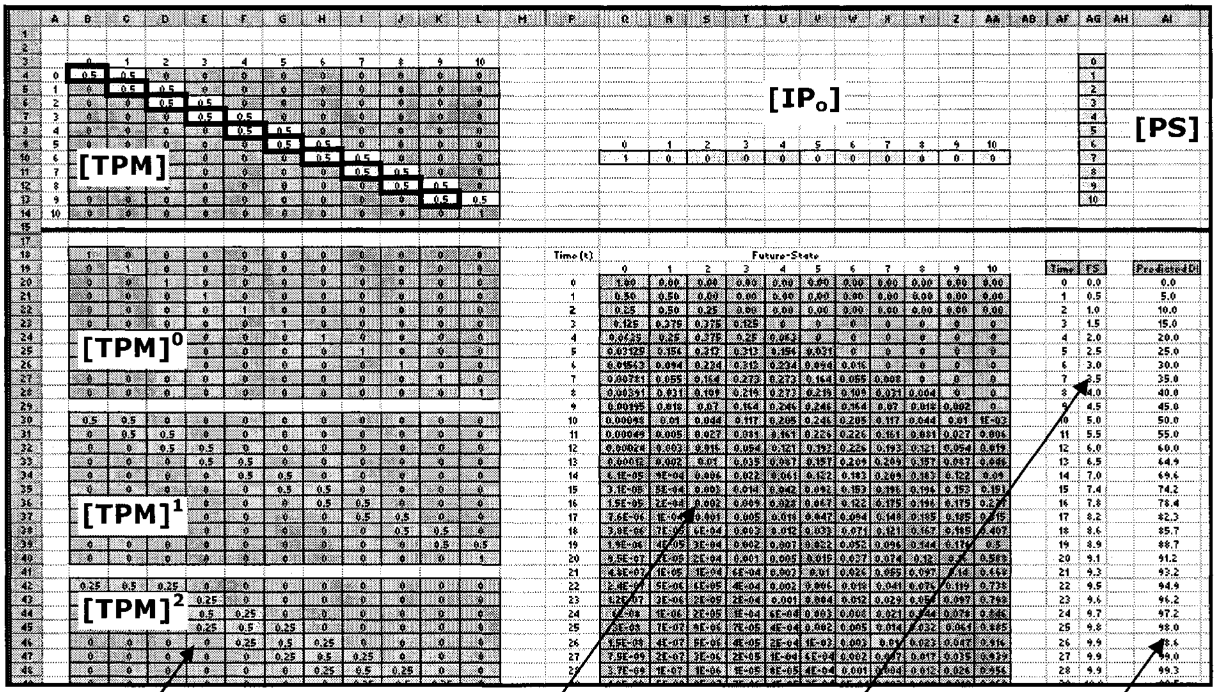
The matrix as such, can be defined by its diagonal values, which are given arbitrary values as the ones shown in the illustrative matrix. It is noted that, with any change to the [TPM] diagonal values, the spreadsheet will calculate the rest of values in the matrix (next to diagonal) and proceed to raise the whole matrix to possible (ts) powers, as shown in Figure 4.3.

3. Initial probability row vector [IP$_o$]

Since the [IP$_o$] is selected at (t = 0), then the [IP$_o$] is a row vector with 11 elements starts with one and the rest of values equal to 0s, as follows:

$$[IP_o] = [1.0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \tag{4.2}$$

The spreadsheet implementation is shown in Figure 4.4. The spreadsheet as such, links all future predictions to the [TPM], or more precisely to the diagonal values of the transition probability matrix. The model is transparent and is greatly facilitated by the powerful matrix manipulation of spreadsheet platforms and their easy-to-use interfaces.



[TPM]^t: TPM multiplications till power 50.

FP$_t$s: Each row represents the Future probability vector at a certain age t (0 to 50)

FS$_t$s: Values represent the predicted state at any age t (0 to 50)

DI: Values represent predicted DIs (i.e., scaled FSs) at any age t (0 to 50)

Figure 4.4: Spreadsheet model for Markov chains

Optimizing the [TPM} Probabilities: With the arbitrary nature of the initial [TPM] probabilities, it is likely to produce incorrect (DI) predictions. However, as the model is set up with all the internal calculations that will automatically provide correct predictions given accurate [TPM] probabilities, the model has the ability to function as an optimization model, in which the variables are the diagonal values in the [TPM] matrix. The objective of the optimization model is to determine the best [TPM] values (variables) so that the Markov predicted deterioration curve coincides with the actual deterioration curve (determined in stage1). As such, the set up of the optimization model is as follows (Figure 4.5):

Objective Function: minimize the total error between Markov-predicted deterioration indices ($DI_{t(M)}$) and the average measured ($DI_{t\ (av)}$, from stage1), summed along the age (A) of the instance being considered, as follows;

$$\text{Min} \sum_{t=1}^{t=A} \left| DI_{t_{(M)}} - DI_{t_{(av)}} \right| \tag{4.3}$$

Subject to:

$$DI_{t_{(M)}} = 10 \times [IP_o] . \begin{bmatrix} P_{1,1} & 1 - P_{1,1} & 0 & . & . & 0 \\ 0 & P_{2,2} & 1 - P_{2,2} & 0 & . & 0 \\ . & 0 & . & . & . & 0 \\ . & . & . & P_{i,i} & 1 - P_{i,i} & 0 \\ . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & . & 1 \end{bmatrix}^t . [PS] \quad \forall\ t; t=1,2,3...A \tag{4.4}$$

$$0 \leq P_{i,i} \leq 1 \tag{4.5}$$

where, [PS] and [IP$_o$] are constant matrices of Equation (4.1) and Equation (4.2), respectively.

In addition, an additional constraint can be used to customize the [TPM] for a specific instance with a known (measured DI) at any time. In this case, the error between predicted (DI) and that measured (DI) should equal to 0 (i.e., the model should correctly estimate the actual measured values for the instance), as follows:

$$\left| DI_{t_{(M)}} - DI_{t_{(\text{measured or real})}} \right| = 0 \tag{4.6}$$

Variables: The variables are the diagonal probability values ($P_{i,i}$) in the [TPM] matrix, which take real values between 0 and 1 (Equation 4.5).

$$\begin{array}{c|ccccccccccc}
\text{Condition State} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\hline
0 & P_{1,1} & p_{1,2} = 1 - p_{1,1} & & & & & & & & & \\
1 & & p_{2,2} & 1 - p_{2,2} & & & & & & & & \\
2 & & & p_{3,3} & 1 - p_{3,3} & & & & & \mathbf{0} & & \\
3 & & & & p_{4,4} & 1 - p_{4,4} & & & & & & \\
4 & & & & & p_{5,5} & 1 - p_{5,5} & & & & & \\
5 & & & & & & p_{6,6} & 1 - p_{6,6} & & & & \\
6 & & & & & & & p_{7,7} & 1 - p_{7,7} & & & \\
7 & & & & \mathbf{0} & & & & p_{8,8} & 1 - p_{8,8} & & \\
8 & & & & & & & & & p_{9,9} & 1 - p_{9,9} & \\
9 & & & & & & & & & & P_{10,10} & 1 - p_{10,10} \\
10 & & & & & & & & & & & 1
\end{array}$$

Probability Transition Matrix [TPM] shaded diagonal values represent the variables

Following Markov Formulation:

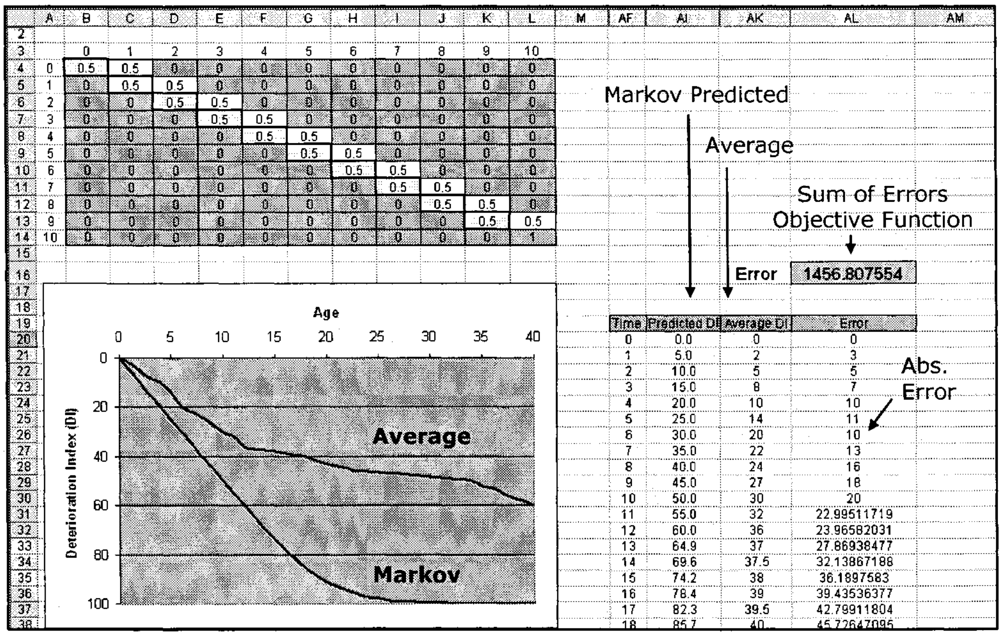
Predicted DI after t years = $[IP_o] \cdot [TPM]^t \cdot [PS] * 10$

year 1 2 . t	Average DI	Predicted DI	Absolute Error (e) = Abs (Predicted DI – Average DI)
Need to be minimized →			Σ e

Figure 4.5: Optimization settings for the spreadsheet Markov model

The implementation of the [TPM] optimization model was carried out on a spreadsheet, as shown in Figure 4.6. Same as Figure 4.4, column (AI) shows the Markov predicted (DIs) at each age. Column (AK) and (AL) were then added in Figure 4.6 to show the desired deterioration curve and the calculated absolute error at each age between Markov prediction and the desired curve. The total error is then summed in one cell (the Objective Function cell), as shown in the figure. The relationship between age and (DIs) for both Markov and Average deterioration is also plotted in Figure 4.6. The Markov predictions were based on initial probabilities of 0.5s as shown in the diagonal [TPM] values.

After the optimization was completed, optimum [TPM] values were obtained as shown in Figure 4.7 with Markov predictions becoming very close to the average (DIs). The optimization did not consider the additional constraint for customizing Markov prediction for a specific instance. Considering this case in a new optimization run, a more accurate prediction curve that passes through the actual (DI) records of that instance was obtained (Figure 4.8).



	0	1	2	3	4	5	6	7	8	9	10
0	0.5	0.5	0	0	0	0	0	0	0	0	0
1	0	0.5	0.5	0	0	0	0	0	0	0	0
2	0	0	0.5	0.5	0	0	0	0	0	0	0
3	0	0	0	0.5	0.5	0	0	0	0	0	0
4	0	0	0	0	0.5	0.5	0	0	0	0	0
5	0	0	0	0	0	0.5	0.5	0	0	0	0
6	0	0	0	0	0	0	0.5	0.5	0	0	0
7	0	0	0	0	0	0	0	0.5	0.5	0	0
8	0	0	0	0	0	0	0	0	0.5	0.5	0
9	0	0	0	0	0	0	0	0	0	0.5	0.5
10	0	0	0	0	0	0	0	0	0	0	1

Markov Predicted

Average

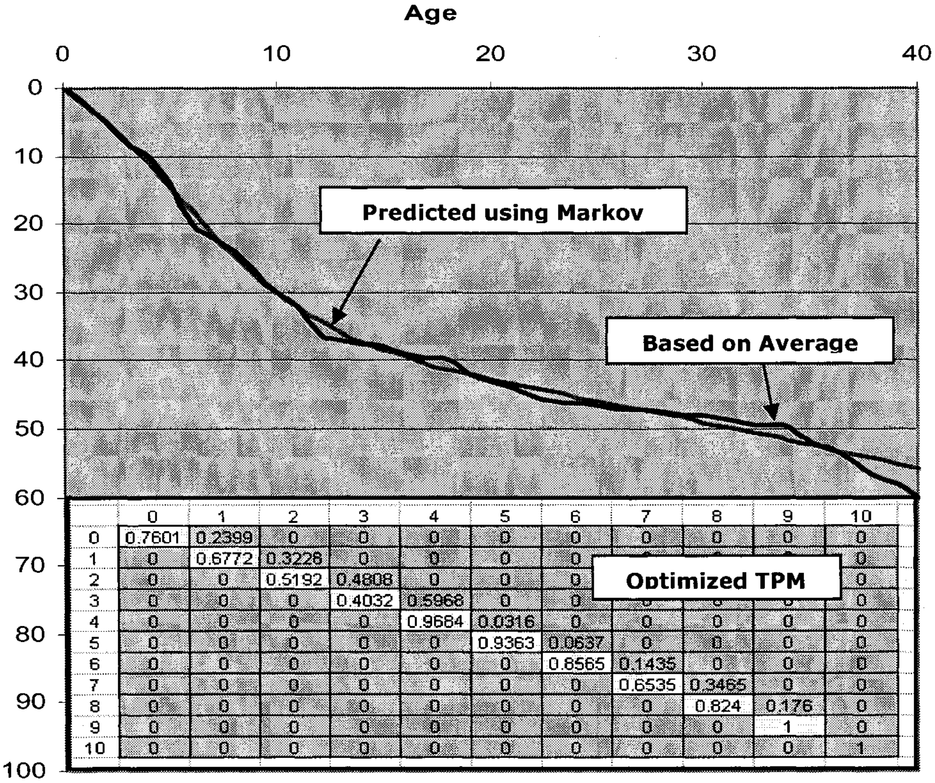
Sum of Errors
Objective Function

Error 1456.807554

Time	Predicted DI	Average DI	Error
0	0.0	0	0
1	5.0	2	3
2	10.0	5	5
3	15.0	8	7
4	20.0	10	10
5	25.0	14	11
6	30.0	20	10
7	35.0	22	13
8	40.0	24	16
9	45.0	27	18
10	50.0	30	20
11	55.0	32	22.99511719
12	60.0	36	23.96582031
13	64.9	37	27.86938477
14	69.6	37.5	32.13867188
15	74.2	38	36.1897583
16	78.4	39	39.43536377
17	82.3	39.5	42.79911804
18	85.7	40	45.72647095

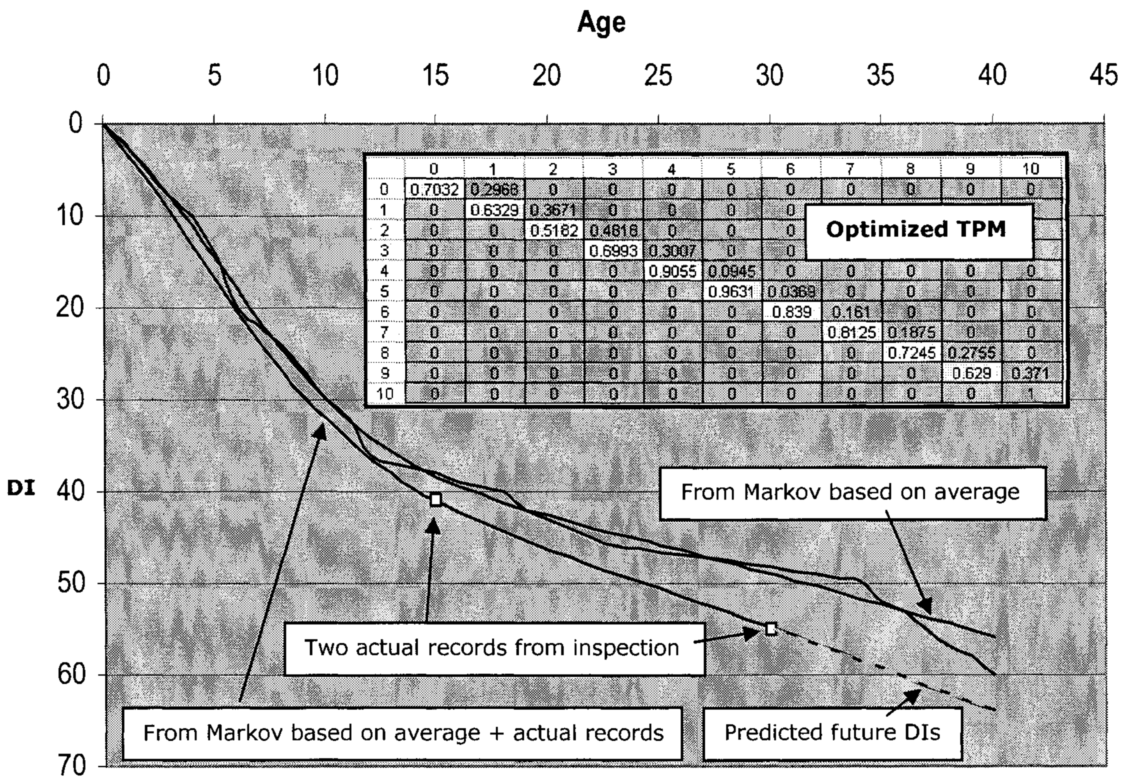
Abs. Error

Figure 4.6: Spreadsheet Markov model



	0	1	2	3	4	5	6	7	8	9	10
0	0.7601	0.2399	0	0	0	0	0	0	0	0	0
1	0	0.6772	0.3228	0	0	0	0	0	0	0	0
2	0	0	0.5192	0.4808	0	0	0	[illegible]	[illegible]	[illegible]	0
3	0	0	0	0.4032	0.5968	0	0	0	0	0	0
4	0	0	0	0	0.9684	0.0316	0	0	0	0	0
5	0	0	0	0	0	0.9363	0.0637	0	0	0	0
6	0	0	0	0	0	0	0.8565	0.1435	0	0	0
7	0	0	0	0	0	0	0	0.6535	0.3465	0	0
8	0	0	0	0	0	0	0	0	0.824	0.176	0
9	0	0	0	0	0	0	0	0	0	1	0
10	0	0	0	0	0	0	0	0	0	0	1

Figure 4.7: Optimized probabilities for the average deterioration



	0	1	2	3	4	5	6	7	8	9	10
0	0.7032	0.2968	0	0	0	0	0	0	0	0	0
1	0	0.6329	0.3671	0	0	0	0				
2	0	0	0.5182	0.4818	0	0	0				
3	0	0	0	0.6993	0.3007	0	0				
4	0	0	0	0	0.9055	0.0945	0	0	0	0	0
5	0	0	0	0	0	0.9631	0.0369	0	0	0	0
6	0	0	0	0	0	0	0.839	0.161	0	0	0
7	0	0	0	0	0	0	0	0.8125	0.1875	0	0
8	0	0	0	0	0	0	0	0	0.7245	0.2755	0
9	0	0	0	0	0	0	0	0	0	0.629	0.371
10	0	0	0	0	0	0	0	0	0	0	1

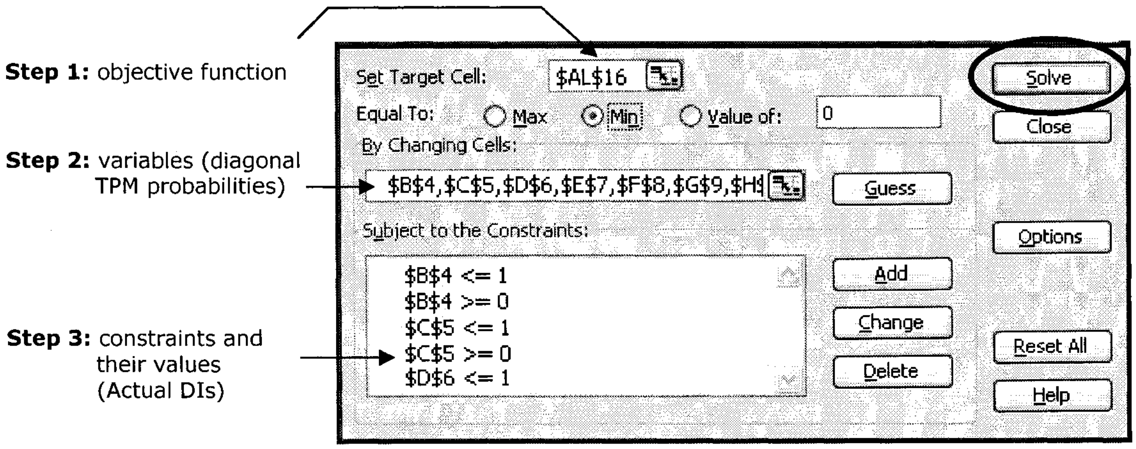
Figure 4.8: Optimized probabilities for a specific instance

With the spreadsheet model describing the objective function, constraints, and variables of the optimization problem, a standard optimization procedure was used to perform the optimization. Due to its non-linear math formulation, this problem is a Non-Linear Programming (NLP) problem and was solved using the SOLVER procedures that come with Excel software. For NLP problems, Excel SOLVER uses the Generalized Reduced Gradient (GRG) method (Lasdon et al. 1978). More details on the mathematical background on solving NLP problems can be found in Luenberger, D. G. 1984.

As an alternative to Excel SOLVER, other spreadsheet optimization tools can be used such as the Genetic Algorithms, EVOLVER (Evolver 1998) add-in program. With the variety of optimization tools available (e.g., SOLVER and EVOLVER), the developed model becomes readily usable for determining the optimum [TPM] probabilities. The general settings for optimization in all these programs can be shown in Figure 4.9.

Although the variable probabilities in the [TPM] were assigned at the diagonal only, it is still possible in the proposed model to consider severe deterioration (moving down more than one state) by considering more variables in the [TPM] matrix. In this case, the upper triangle values in the [TPM] matrix can be considered as variables.

Once the matrix is optimized (i.e., customized for the instance under consideration), it can be used to predict the future (DIs) during the planning horizon (the dotted line in Figure 4.8).



Step 1: objective function

Step 2: variables (diagonal TPM probabilities)

Step 3: constraints and their values (Actual DIs)

Set Target Cell: AL16

Equal To: ○ Max ● Min ○ Value of: 0

By Changing Cells: B4,C5,D6,E7,F8,G9,H

Subject to the Constraints:

```
$B$4 <= 1
$B$4 >= 0
$C$5 <= 1
$C$5 >= 0
$D$6 <= 1
```

Figure 4.9: Main settings for spreadsheet optimization tools

4.2.3 Stage3: Instance Future Severities of Defects

In this last stage of the proposed deterioration model, the future severities corresponding to a predicted (DI) from stage 2 will be calculated. As the instance deterioration index (DI) increases with time (i.e., the condition worsens), the deficiencies are expected to reflect increase in their severities. The prediction of the future severities, however, is not a straightforward task. For example, let's consider an instance with only two deficiencies (D_1) and (D_2) with weights 40% and 60%, respectively. If the measured severities during the inspection were 30 and 50, respectively, then using Equation (3.1), the deterioration index (DI) at inspection year is calculated as (DI = (40x30 + 60x50) / 100 = 42), as shown in Figure 4.10 (a). If it is predicted that the deterioration index after 2 years will be 54, then the severities of the two defects in the third year need to be determined by solving an equation in two unknowns, as shown in Figure 4.10 (b). Solving this equation for two unknowns only is not a simple task and is even more complex if the components have more deficiencies.

To determine the future severities, an incremental approach is used to increase the inspected severities proportionally to their relative values, until reaching the predicted (DI). To facilitate the calculation, a spreadsheet model for the calculations is developed, and a macro program is coded using Visual

Basic to follow the previous process. The model estimates future severities based on two inputs: (1) a set of measured severities (from inspection) and their weights; and (3) a future (DI), from the deterioration model. The macro code keeps increasing the severities relatively until the estimate error becomes a small value of 0.01 or less. The spreadsheet is shown in Figure 4.11.

a) At Inspection Year:

Inspected severities are 30 and 50

Defect	Weight	Inspected severities
D_1 D_2	40% 60%	30 50
		DI = 42

$$\frac{30\times 40+50\times 60}{100} = 42$$

b) At Year 3:

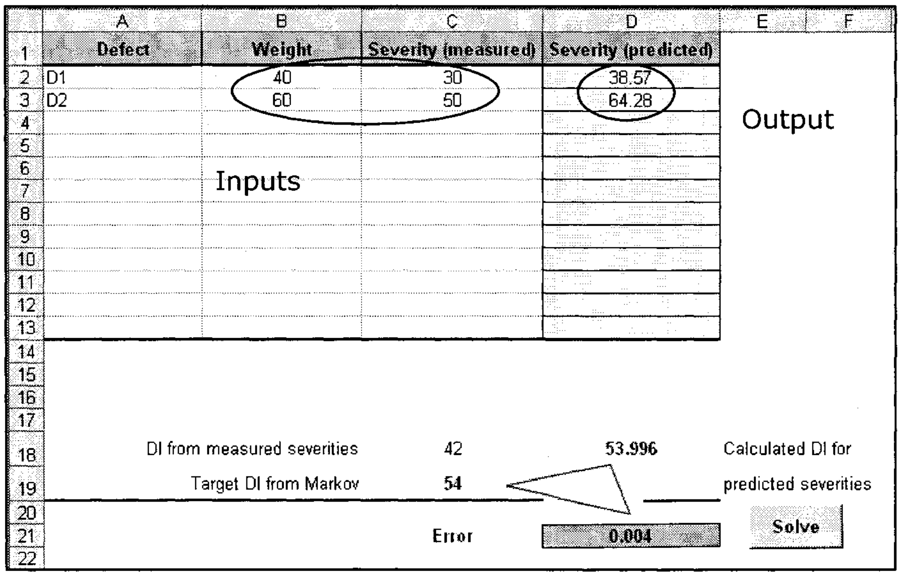
Predicted Deterioration Index = 54

Defect	Weight	Unknown severity
D_1 D_2	40 60	**S_1 =?** **S_2 =?**
		DI_3 = 54

$$\frac{S_1\times 40 \;+\; S_2\times 60}{100} = 54$$

$S_1 \leq 100$; $S_2 \leq 100$ → get S1 & S2 at year 3

Figure 4.10: Severity prediction challenges



	A	B	C	D	E	F
1	Defect	Weight	Severity (measured)	Severity (predicted)		
2	D1	40	30	38.57		
3	D2	60	50	64.28		

DI from measured severities 42 53.996 Calculated DI for predicted severities

Target DI from Markov 54

Error 0.004

Figure 4.11: Prediction model stage 3: future severities

Once the future severities are determined, they represent specific and defined defects that need to be repaired. The future extents of specific defects are not just a condition state or a numeric index, but represent the specific deterioration behavior of an instance, rather than an average behavior.

4.3 Repair Selection for Building Components: An Optimization Model

In this section, an optimization model is introduced to determine the best repair option for each instance for each year in the planning horizon. For each year, the deterioration model covered in section 4.2 predicts the deterioration level (DI) for any instance along with the specific severities for its defects (Figure 4.12). These severities represent the extent of problems that need to be repaired in that year.

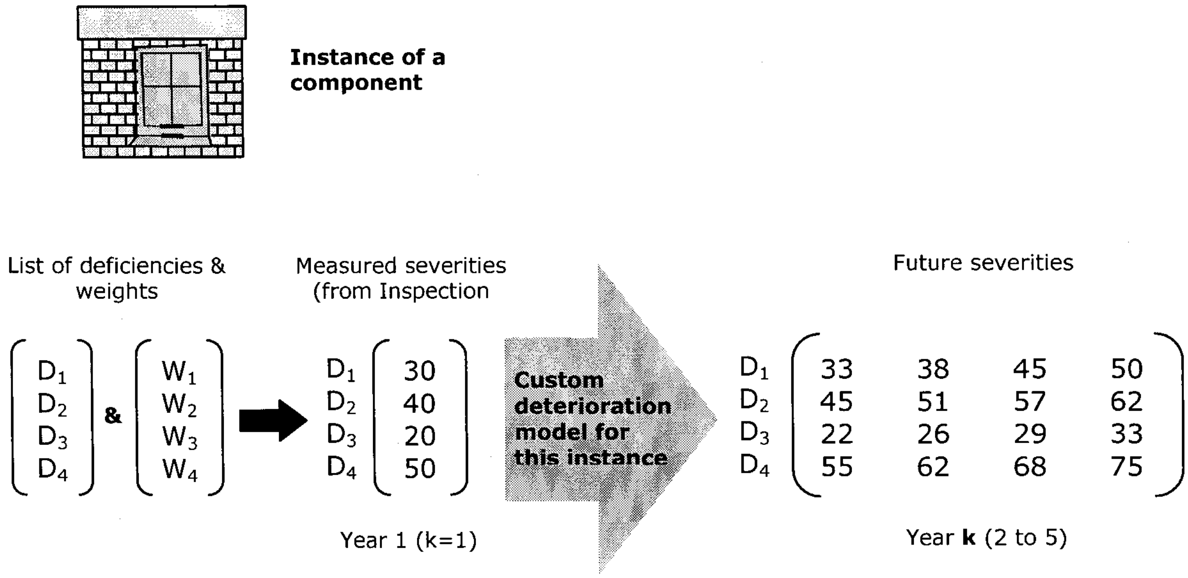


Figure 4.12: Results from the 3-stage deterioration model

Since it is possible to repair some of these defects but not the others, it is possible to generate various repair options (scenarios) by enumerating all possibilities of repair. However, among the various scenarios, it is necessary to identify the cheapest repair scenario that brings the condition of the instance to an acceptable level. Details of this analysis are discussed in the following sub-sections.

4.3.1 Creating Repair Scenarios

Repair scenarios (RSs) can be generated for any component using its list of deficiencies. These scenarios involve decisions whether to repair or not to repair any combination of the deficiencies (Figure 4.13). For example, repair scenario 2 in a binary form (1, 1, 0, 1) in Figure 4.13 implies repairing deficiencies 1, 2, and 4 and not 3. As such, for a component with (d) deficiencies, then the possible number of repair scenarios equals (2^d).

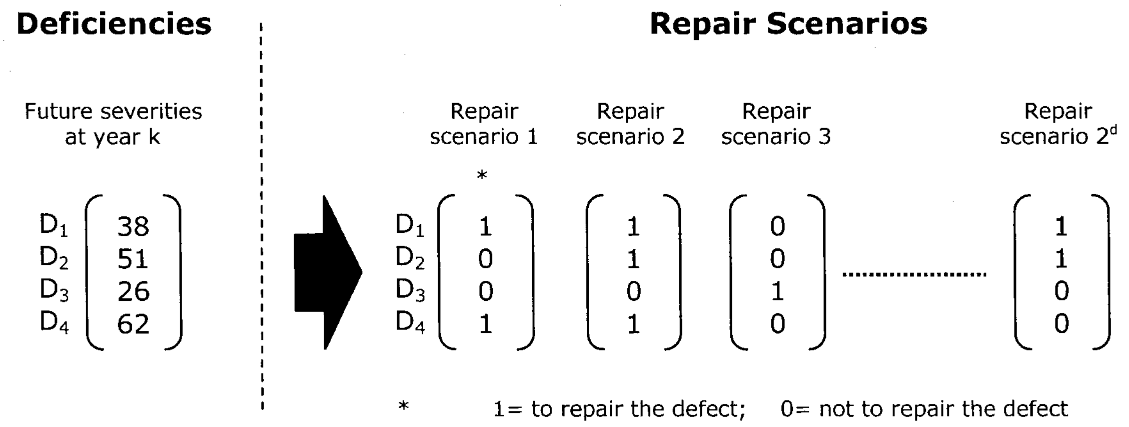


Figure 4.13: Possible repair scenarios

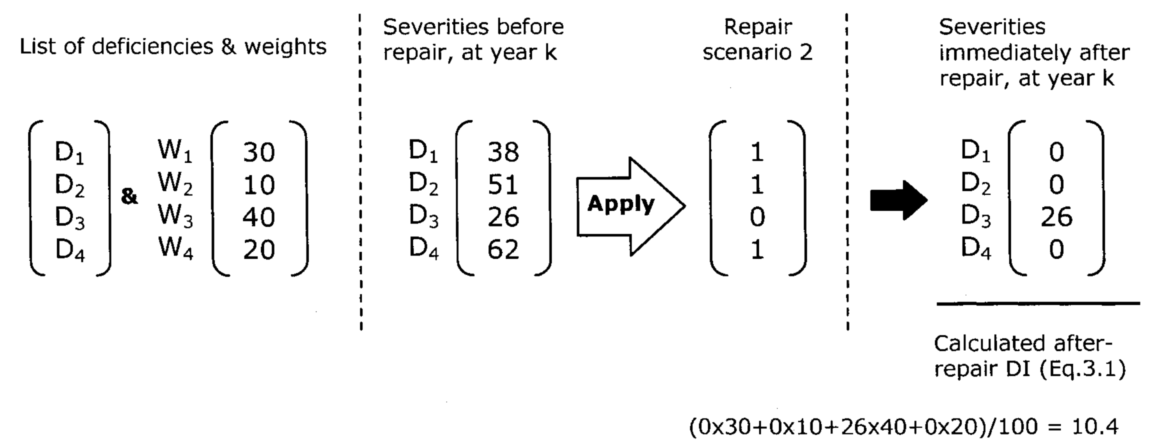
While all possible scenarios are assumed to be valid and feasible, a filter to check the constructability of any repair scenario can be easily incorporated. In general, however, each of these repair scenarios will have its implication in terms of cost, as well as, level of improvement to the condition of the instance when the repair scenario is carried out in its planning year.

4.3.2 Condition Improvement Due-To Repair

Once a repair scenario is selected, the (DI) immediately after repair can be easily calculated, as shown in Figure 4.14. In this calculation, the severities of the repaired defects (defects with 1s in the repair scenario) become 0s (i.e., repaired), while the not repaired defects will have the same severities as the before case. Once the after-repair severities are determined, the after-repair deterioration index (DI_{AR}) for the instance is then calculated using Equation (3.1) to be 10.4. In a mathematical representation, the after repair severities for any repair scenario are expressed as: S*(1-RS). Accordingly, the after-repair deterioration index (DI_{AR}) is directly expressed as:

$$DI_{AR}^{(k)} = \frac{\sum_{i=1}^{d} W_i \cdot S_{ik} \cdot (1 - RS_i)}{100} \tag{4.7}$$

($DI_{AR}^{(k)}$) is the After-Repair-Deterioration Index in year (k). As per Equation 4.7, the improvement depends on the future severities at year (k) and the repair scenario used.



List of deficiencies & weights
Severities before repair, at year k
Repair scenario 2
Severities immediately after repair, at year k
D1 D2 D3 D4
W1 30 W2 10 W3 40 W4 20
&
D1 38 D2 51 D3 26 D4 62
Apply
1 1 0 1
D1 0 D2 0 D3 26 D4 0
Calculated after-repair DI (Eq.3.1)
(0x30+0x10+26x40+0x20)/100 = 10.4

Figure 4.14: Condition improvement due-to certain repair scenario

4.3.3 Repair Cost Estimate

Any repair scenario is expected to have its associated cost, expressed as a percentage of full replacement cost for that instance. For example, repair scenario (1 0 1 1) costs 62% of the cost of the full replacement scenario (1 1 1 1).

For any repair scenario (RS), it is possible to estimate its relative repair cost percentage (%RC) using two simple assumption:

1. Deficiency weight is a good representative of the relative repair cost of the defect; and

2. Repairing one defect individually will cost more than the cost to repair it within a full replacement (i.e., repairing all defects at the same time).

Based on these two assumptions and an assumed 25% cost increase in the case of repairing the individual defects, the cost to repair a single defect (i), as a percentage is expressed as 1.25*W_i.

Therefore, the total percentage cost for any repair scenario (RS) can be determined by summing the costs of repairing all the defects that are decided to be repaired in this repair scenario, as follows (Figure 4.15):

$$\%RC = \sum_{i=1}^{d} 1.25 \times W_i \times RS_i \qquad (4.8)$$

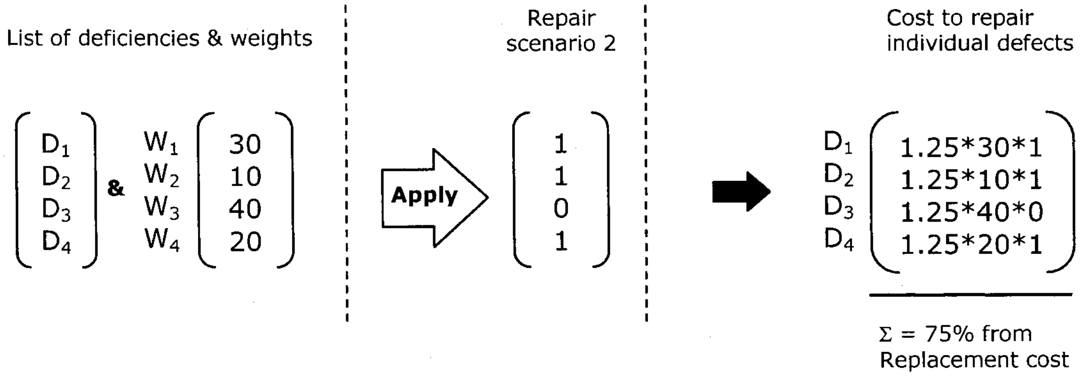


Figure 4.15: Percentage repair cost for a certain repair scenario

4.3.4 Optimum Repair Scenario

At this stage, a building instance will have its repair scenarios (2^d) defined along with their costs and improvement implications when these repair scenarios are carried out in a certain year. Selecting the best repair scenario has been modeled as an optimization problem to reach to a desirable condition (DDI) with minimum repair cost, as follows:

Objective Function:

$$\text{Min } (\%RC) = \text{Min} \sum_{i=1}^{d} 1.25 \times W_i \times RS_i \qquad (4.9)$$

Subject to:

$$DI_{AR} = \frac{\sum_{i=1}^{d} W_i \times S_i \times (1 - RS_i)}{100} \leq DDI \tag{4.10}$$

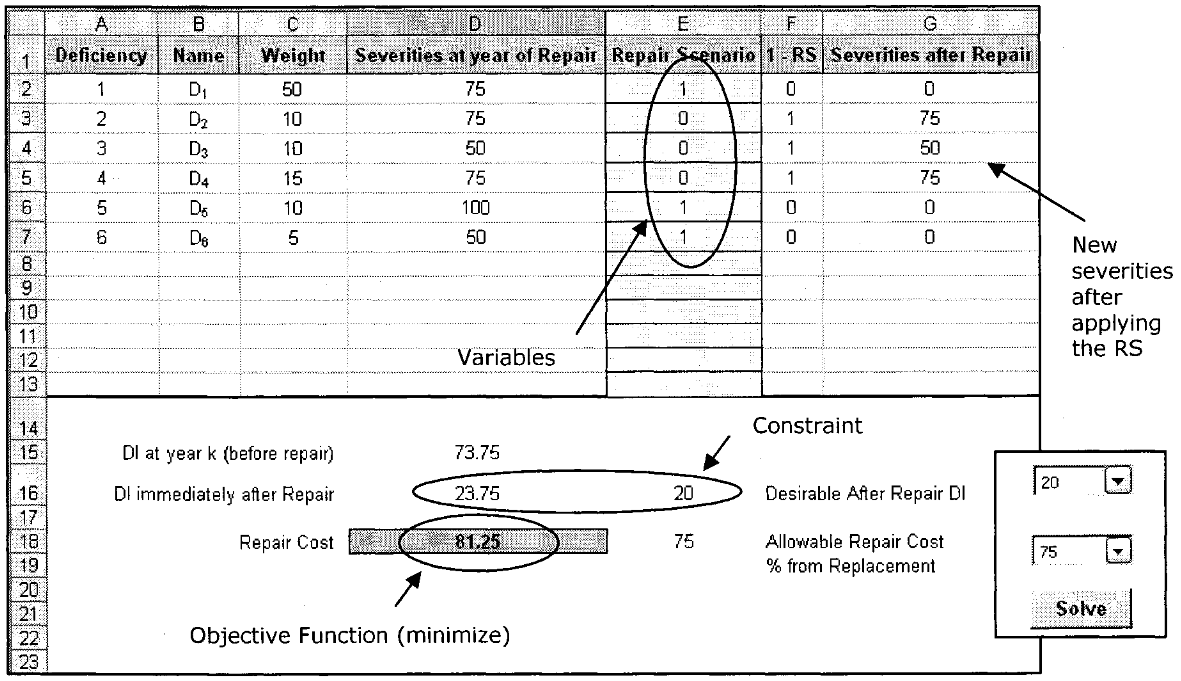
Variables: The RS_i are binary variables: either 1 (to repair defect i) or 0 (to ignore the repair);

$i = 1, 2 \ldots d$ (4.11)

This optimization problem can be classified as an Integer Programming (IP) problem and has been solved by Excel SOLVER using the branch and bound technique described in Appendix A.

For practicality, once the optimum scenario is determined, its cost is compared with the full-replacement cost, before a final decision is made. In this study, a repair scenario costing more than 75% of replacement cost will justify replacement instead.

The optimization model is implemented on a spreadsheet program as shown in Figure 4.16.



Deficiency	Name	Weight	Severities at year of Repair	Repair Scenario	1 - RS	Severities after Repair
1	D_1	50	75	1	0	0
2	D_2	10	75	0	1	75
3	D_3	10	50	0	1	50
4	D_4	15	75	0	1	75
5	D_5	10	100	1	0	0
6	D_6	5	50	1	0	0

DI at year k (before repair) 73.75

DI immediately after Repair 23.75 20 Desirable After Repair DI

Repair Cost 81.25 75 Allowable Repair Cost % from Replacement

Figure 4.16: Optimization model to determine the best repair scenario

It is noted that the optimization model needs the user to specify a desirable (DI) limit. This limit is called the desirable after-repair DI or (DDI), which can be defined as the minimum acceptable level for the component's condition after-repair.

The condition constraint (DDI) can be easily set as a desired value (e.g., 30) as a fixed value that the component is not allowed to deteriorate beyond. However, instead of a fixed (DDI) value to use for all components, a more practical approach is to set (DDI) relative to the importance of the component (using the relative importance factor (RIF) defined in section 3.4). As discussed in Chapter 3, the importance factor ranges from 100 (most important) to 0 (least important) and reflects the component's impact on safety, functionality, and other components. It is assumed, therefore, that the higher the importance of a component, the more desire to repair the component to a better condition (i.e., lower DDI). As such, the (DDI) constraint for each component is set as:

$$\boldsymbol{DDI = 100 - RIF} \qquad (4.12)$$

Once the (DDI) is known, the optimization model of Figure 4.16 selects the best repair scenario and, accordingly, its repair cost percentage (%RC) becomes known. To convert the (%RC) into real dollars, the following equations are used:

- Instance Repair Cost ($IRC) = {(%RC) * Instance Replacement Cost ($)}; and
- Instance Replacement Cost ($) = {Instance Relative Size (Z) * Replacement Cost of all instances of the component ($CRC)}.

where, %RC is the instance percentage cost obtained from the optimization model, (Z) is the instance relative size (e.g., 20% of roof area will be repaired), and the $CRC is the total replacement cost of the component (e.g., all the roof). The previous two equations can be combined as follows:

$$\$IRC_j = \%RC_j \times Z_j \times \$CRC \qquad (4.11)$$

Typically, for standard buildings such as hotels, schools, and hospitals, the cost per usable area is used as a base for cost calculations. Accordingly, the $CRC can easily be calculated from cost tables that list the cost per educational area in the school. To facilitate the calculations, a simple database can be used to store data of typical $/education area for various components along with data related to the educational area in every school.

The sizing factor (Z) can also be stored as characteristic information of any instance that is pre-calculated as the relative size, area, count, or capacity of the instance, compared to the rest of instances that form a component. For example, if a building has four window instances with number of windows as follows: Instance 1 (10 windows on east side); Instance 2 (15 windows on west side); Instance 3 (10 windows on north side); and Instance 4 (20 windows on south side), then (Z) becomes (10/55), (15/55), (10/55), and (20/55) = 18.2%, 27.3%, 18.2%, and 36.3% for instances 1, 2, 3, and 4, respectively.

As an example to the cost calculations, let's consider that the best repair scenario obtained from optimization for a certain instance has a repair cost of 56.25%. If this instance has a relative size of 30% and the component replacement cost is $100,000, then the repair cost of this instance is estimated to be (56.25% x 30% x $100,000) = $16,875.

4.4 Deterioration after Repair

Given a best repair scenario for a building instance, the after-repair (DI) of this instance is expected to be improved. However, in the years after the repair, the instance condition will still deteriorate. In this study, it is assumed that after-repair deterioration will follow the same custom deterioration curve obtained from Markov for that instance, as shown in Figure 4.16. The example in Figure 4.17 shows that a best repair scenario is selected because it improves (DI) to 18.5 in the second year of the planning horizon (lower than DDI of 20).

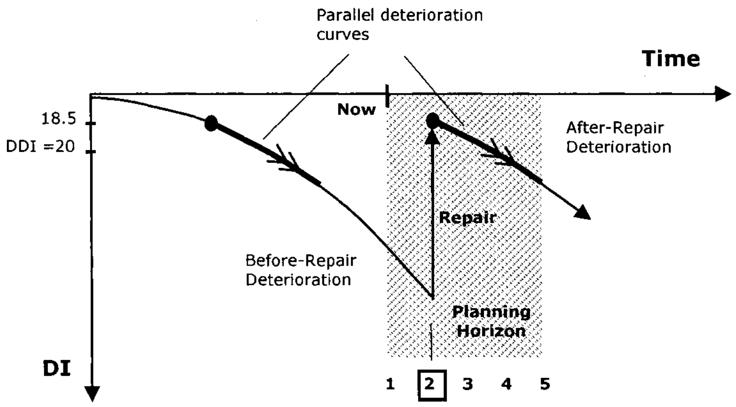

Parallel deterioration curves
Time
18.5
DDI =20
Now
After-Repair Deterioration
Repair
Before-Repair Deterioration
Planning Horizon
DI
1
2
3
4
5

Figure 4.17: After-repair deterioration

Using the custom deterioration curve of that instance, after-repair deterioration become parallel to the before-repair segment that starts with a (DI) of 18.5.

4.5 Implementation: Sequential Optimization Analysis Program (SOAP)

As concluded from the previous sections, both the deterioration model and the repair model largely use optimization, one after the other. Hence, both models are combined in one module (an analysis module) in the proposed asset management framework. This module is called SOAP or Sequential Optimization Analysis Program. SOAP is applied at the instance level to develop a custom deterioration curve by optimizing the transition probabilities for Markov chain. Afterwards, it determines the best repair scenario for each year in the planning horizon. The detail of this model is shown in Figure 4.18.

4.6 Conclusions

In this chapter a dynamic approach for instance-level LCCA has been introduced. The proposed approach depends mainly on accurate condition assessment surveys of the asset inventory, considering component deficiencies. For each building component, the condition assessment data is used to generate a dynamic deterioration model, which is then used in a repair selection model. The repair selection model determines, for each year of the planning horizon, the cheapest repair strategy that raises the condition of the component to a user-defined minimum acceptable level.

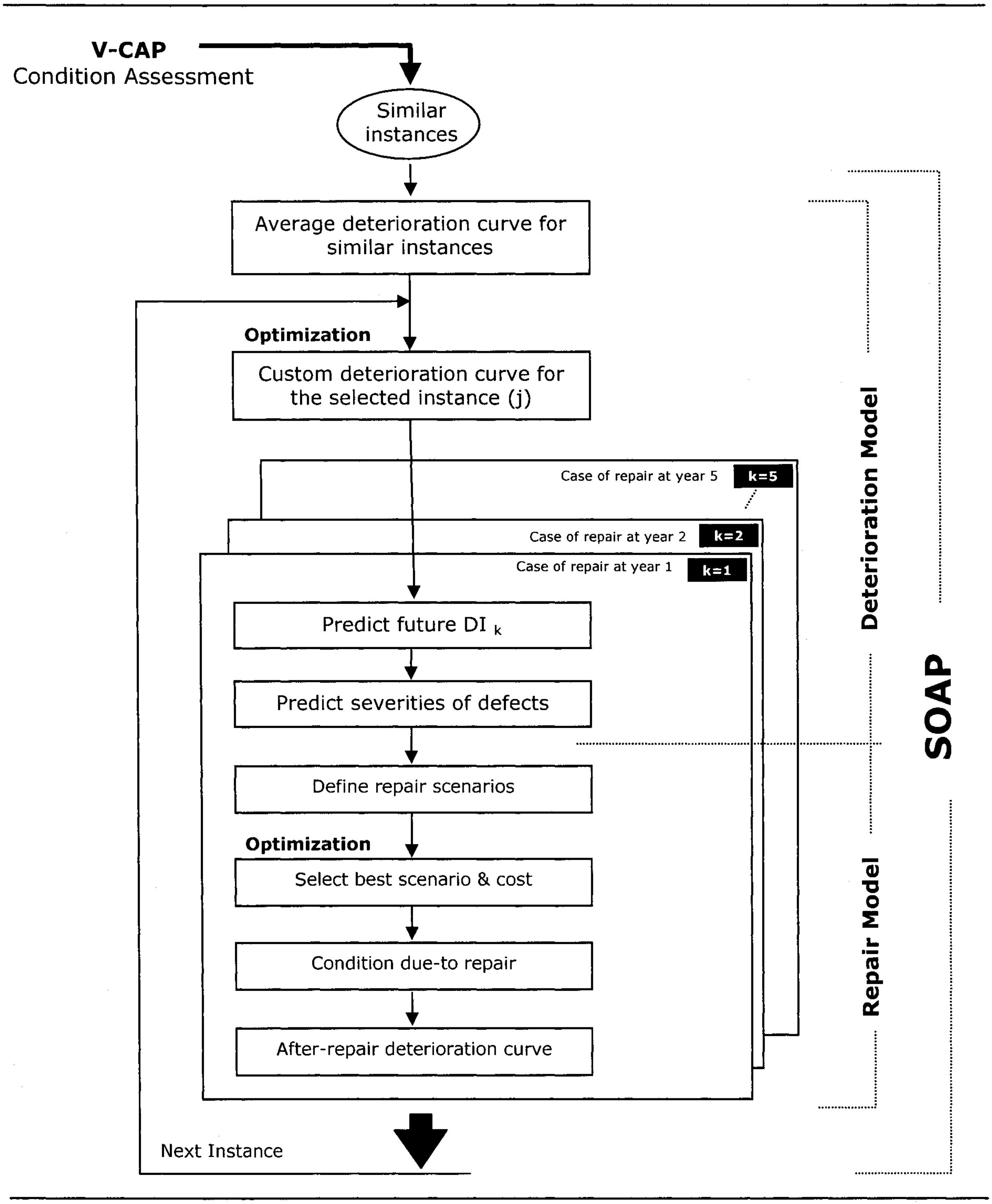


Figure 4.18: Instance LCCA using SOAP

Chapter 5

FORMULATION FOR NETWORK-LEVEL OPTIMIZATION & REPAIR FUND ALLOCATION

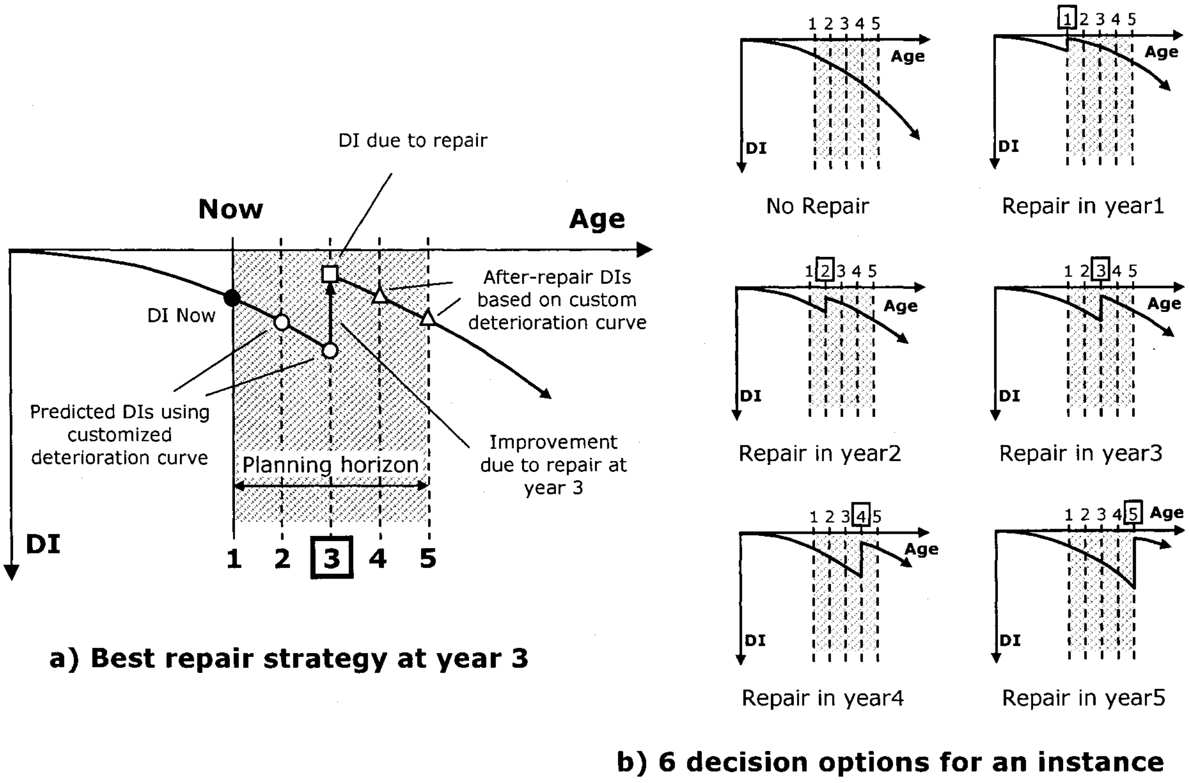
5.1 Introduction

In this chapter, a network-level optimization is presented to facilitate repair prioritization and fund allocation decisions. The optimization model determines, for each instance in a network of instances, whether it should be repaired during the planning horizon and at which year. The optimization is formulated to consider the limited available budget, in addition to any organizational preferences. The optimization objective function attempts to maximize the overall improvement to the network by minimizing the overall network deterioration index, considering limited yearly budgets. Alternative formulations for the optimization problem are discussed in this chapter, while actual implementation is presented along the description of a case study in Chapter 6.

5.2 From Instance-Level to Network-Level Analysis

In the previous chapter, SOAP analysis was carried out at the instance level to provide the asset management framework with the instance's optimal life cycle costs corresponding to each year in the planning horizon. The analysis outputs are summarized in Figure 5.1.

Figure 5.1 (a) shows the results of SOAP analysis when repairing an instance at year 3 of its planning horizon, using the best repair scenario for that year. Since SOAP analysis is carried out for each year in the planning horizon, the final output of the analysis for any instance is a set of six curves that represent six decision options for that instance (Figure 5.1 b). Each decision option is well defined in terms of its year of repair and the consequent implications (repair cost, due-to repair improvement, and before-repair and after-repair conditions). In the next subsection, the decision options are evaluated from a network-level prospective to determine the optimum year to repair each instance (i.e., 0, 1, 2, 3, 4, or 5). It is noted here that, a no-repair decision option is possible, which corresponds to year 0.



a) Best repair strategy at year 3

b) 6 decision options for an instance

Figure 5.1: Summary of SOAP results

5.2.1 Expected Performance of Repair Options over the Planning Horizon

The six decision options for repairing an instance, just mentioned earlier, need to be differentiated not only in terms of cost but also in terms of a single value to represent the expected performance over the planning horizon. One way of determining the expected performance of a repair option is to calculate the area under the deterioration curve, such as, the highlighted area in Figure 5.2 (a). However, this area representation is not suitable because it does not consider the effect of a repair action in some years in the planning horizon. For example, the repair action at year 5, shown in Figure 5.2 (b), has the same deterioration area as the case of no-repair. Therefore, a suitable representation of the expected performance over the planning horizon is to use the average of the yearly (DI) values (shown on the curves in Figure 5.2). It is important to note that the no-repair case has only five points for calculating the expected performance (Figure 5.2 a), while the case of a repair at any year has six points (e.g., Figure 5.2 b or c).

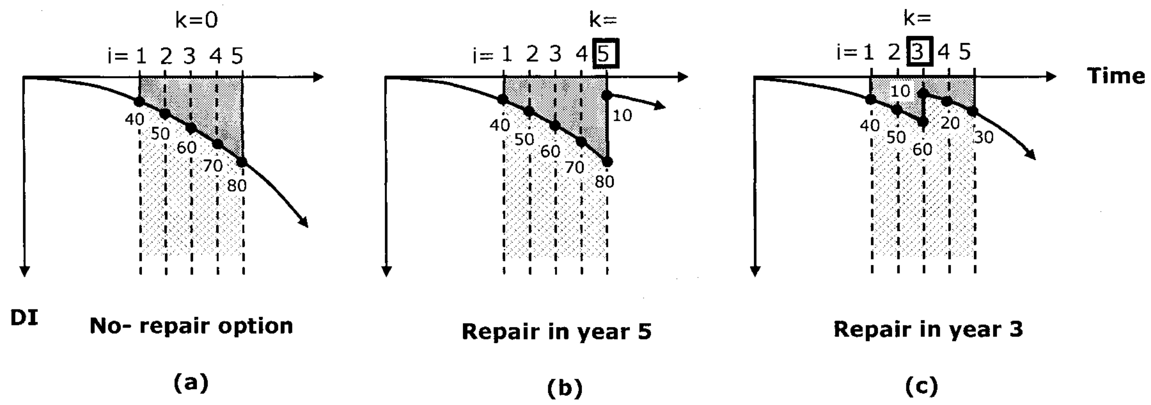


Figure 5.2: Expected repair performance over the planning horizon

The expected performance (EP) over the planning horizon, as such, represents the average deterioration (average DIs) and can be mathematically represented as:

$$EP_j^{(k)} = Average\,(DI_i)_j^{(k)} \quad \forall\, i \in Planning\ Horizon \qquad (5.1)$$

where, $EP_j^{(k)}$ represents the expected performance of instance (j) that is repaired at year (k); and $(DI_i)_j^{(k)}$ represents the (DI) value at year (i) in the planning horizon, when instance (j) is repaired at year (k).

In this manner, the calculations for the (EP) in the three cases of Figure 5.2 are as follows:

- Case the instance is not repaired in any year (Figure 5.2 a), the $EP_j^{(0)}$ equals to (40+50+60+70+80) / 5 = 60;

- Case of repairing the instance at year 5 (Figure 5.2 b), the $EP_j^{(5)}$ equals to (40+50+60+70+80+10) / 6 = 51.67; and

- Case of repairing the instance at year 3 (Figure 5.2 c), the $EP_j^{(3)}$ equals to (40+50+60+10+20+30) / 6 = 35.

The difference among these values clearly shows the implications of repair decisions on the performance of the instance. The case of no-repair logically has the highest value (worst performance) during the planning horizon, but it has no repair cost. The other two cases show two different repair strategies with different cost implications and expected performances (the repair in year 3 improves EP from 60 to 35, while the repair in year 5 improves EP from 60 to 51.67).

Through this representation of expected performance, all the six repair decisions produced by SOAP (which are known in terms of specific actions to repair specific defects) can be clearly defined in terms of costs and EPs. Since the network of components involves hundreds or thousands of these instances, SOAP analysis provides a large pool of data regarding six decision options for each instance. This data lends itself well to optimization modeling to determine for each instance its optimum decision option (year of repair) from the network prospective.

5.2.2 Overall Network Condition

The expected performance explained in the previous section expresses the condition of each instance over the planning horizon, considering the repair action. At the network level, however, an overall network condition over the planning horizon can be determined from the (EPs) of the individual instances in the network. The overall network condition represents a deterioration index (DI_N), ranging from 0 to 100, where 0 is the best, and is calculated by averaging the instances' EPs, weighted by their relative importance factors (discussed in section 3.4), expressed as follows:

$$DI_N = \frac{\sum_j RIF_j \times EP_j^{(k)}}{\sum_j RIF_j} \qquad \forall\ j \in \textit{the Network}\ (N) \qquad (5.2)$$

where, (EP_j) is the expected performance of instance (j) that is repaired at year (k) and (RIF_j) is the relative importance factor of instance (j).

The network condition (DI_N), as such, is a function of the repair year of each individual instance in the network. Therefore, the (DI_N) represents an evaluation for the network condition after deciding the timing of repair for each instance. It is important also to note that, as the set of decisions regarding the year of repair of each instance directly impact the network (DI_N), it accumulates repair costs for each year of the planning horizon.

5.3 Prioritization and Fund Allocation: A Network–Level Optimization

Optimally allocating a limited budget among various instances is a difficult task that requires structured decision support. Prioritization and repair fund allocation means determining whether an instance should be repaired in the current planning horizon (5-year plan) or not, and at which year should this repair happen. This process has been formulated as an optimization problem, with the year of repair (0 to 5) as the optimization variables for each instance in the network. The optimization attempts to improve the overall network condition by optimally allocating the available budget constraint in each year.

The objective function is set to attain the best overall condition for the whole network of building instances, at the end of the planning period (i.e., best value for the repair money). The overall network condition is represented by the network deterioration index (DI_N) discussed in the previous section. Since the (DI_N) ranges from 0 to 100 with 0 represents the best condition, then the optimization process minimizes the objective function value (seeking a better condition), as follows:

$$Min\ (DI_N) = \underset{jk}{Min} \left(\frac{\sum_j RIF_j \times EP_j^{(k)}}{\sum_j RIF_j} \right) \qquad \forall\ j \in \textit{the Network}\ (N) \qquad (5.3)$$

where, (DI_N) is the (DI) for the whole network at the end of the planning horizon; (RIF_j) represents the relative importance factor for instance (j); and ($EP_j^{(k)}$) is the expected performance of instance (j) based on repair at year (k).

Mathematically, the variables in Equation (5.3) can be represented as binary variables (Y_{jk}: 1 or 0) where 1 corresponds to the repair year (k), as illustrated in the decision matrix Equation (5.4):

$$\textit{\textbf{Variables}} = \begin{array}{r|cccccc|l} & 0 & 1 & . & k & . & 5 & \textbf{(repair year)} \\ \text{Instance}\ \ 1 & Y_{1,0} & Y_{1,1} & Y_{1,2} & Y_{1,3} & Y_{1,4} & Y_{1,5} & \\ 2 & Y_{2,0} & Y_{2,1} & Y_{2,2} & Y_{2,3} & Y_{2,4} & Y_{2,5} & \\ . & . & . & . & . & . & . & \\ \mathbf{j} & . & . & . & Y_{jk} & . & . & \\ . & . & . & . & . & . & . & \\ J & Y_{J,0} & Y_{J,1} & Y_{J,2} & Y_{J,3} & Y_{J,4} & Y_{J,5} & \end{array} \qquad (5.4)$$

With the binary representation, it is possible to express the objective function as follows:

Objective Function:

$$Min\ (DI_N) = Min \left(\frac{\sum_j \sum_k Y_{jk} \times RIF_j \times EP_{jk}}{\sum_j RIF_j} \right) \qquad (5.5)$$

where, (EP_{jk}) is the element (j,k) in the expected performance matrix for all instances and all repair years, as shown in Equation (5.6):

$$EP = \begin{array}{c|cccccc|l} & 0 & 1 & . & k & . & 5 & \textbf{(repair year)} \\ \hline 1 & EP_{1,0} & EP_{1,1} & EP_{1,2} & EP_{1,3} & EP_{1,4} & EP_{1,5} & \\ 2 & EP_{2,0} & EP_{2,1} & EP_{2,2} & EP_{2,3} & EP_{2,4} & EP_{2,5} & \\ 3 & . & . & . & . & . & . & \\ j & . & . & . & \boldsymbol{EP_{jk}} & . & . & \\ . & . & . & . & . & . & . & \\ J & EP_{J,0} & . & . & . & . & EP_{J,5} & \end{array} \qquad (5.6)$$

Subject to:

$$\left(\sum_{j} \$IRC_{jk} \times Y_{jk} \right) \leq \$B_k \qquad \forall\, k \qquad (5.6)$$

$$\sum_{k} Y_{jk} = 1 \qquad \forall\, j \qquad (5.7)$$

Y_{jk} binary (0 or 1); k = 0, 1, 2, 3, 4, 5 (5.8)

Equation (5.6) shows the total yearly repair costs for the instances repaired in year (k) which should be less than the yearly budget in that year ($\$B_k$), where $\$IRC_{jk}$ is the instance's repair cost in year (k).

It is noted that the $\$IRC_{jk}$ represents an element (j,k) in the repair cost matrix for all instances in the network for all repair years, as shown in Eq. 5.9:

$$\$IRC = \begin{array}{c|cccccc|l} & 0 & 1 & . & k & . & 5 & \textbf{(repair year)} \\ \hline 1 & \$IRC_{1,0} & \$IRC_{1,1} & \$IRC_{1,2} & \$IRC_{1,3} & \$IRC_{1,4} & \$IRC_{1,5} & \\ 2 & \$IRC_{2,0} & \$IRC_{2,1} & \$IRC_{2,2} & \$IRC_{2,3} & \$IRC_{2,4} & \$IRC_{2,5} & \\ 3 & . & . & . & . & . & . & \\ \mathbf{j} & . & . & . & \boldsymbol{\$IRC_{jk}} & . & . & \\ . & . & . & . & . & . & . & \\ J & \$IRC_{J,0} & . & . & . & . & \$IRC_{J,5} & \end{array} \qquad (5.9)$$

With the fact that the network of instances is extremely large, the binary representation for the variables will result in a large optimization problem, as each instance has six binary decisions. Accordingly, an alternative representation of the variables as integers has been introduced in an effort to reduce problem size. In this integer representation, each instance (j) in the network will have only one integer variable (Y_j) between 0 and 5 (i.e., 0, 1, 2, 3, 4, or 5). The mathematical formulation in this case is as follows:

Objective Function:

$$Min\ (DI_N) = Min \left(\frac{\sum_j RIF_j \times EP_j}{\sum_j RIF_j} \right) \qquad (5.10)$$

Subject to:

- Expected performance values selected based on the variables Y_j

$$EP_j = \begin{cases} EP_{j1} & ; if\ Y_j = 0 \\ EP_{j2} & ; if\ Y_j = 1 \\ EP_{j3} & ; if\ Y_j = 2 \\ EP_{j4} & ; if\ Y_j = 3 \\ EP_{j5} & ; if\ Y_j = 4 \\ EP_{j6} & ; if\ Y_j = 5 \end{cases} \qquad \forall\ j \qquad (5.11)$$

- Yearly sum of cost $\leq$yearly budget

$$\left(\sum_j \$IRC_{jk} \right) \leq\ \$B_k \qquad for \quad k = 1, 2, 3, 4, and\ 5 \qquad (5.12)$$

$$\text{where, } \$IRC_{jk} = \begin{cases} \$IRC_{jk} & ; if\ Y_j = k \\ 0 & ; if\ Y_j \neq k \end{cases}$$

Y_j integer; 0, 1, 2, 3, 4, 5 (5.13)

The (EP_j) is the expected performance from the expected performance matrix shown in Equation (5.14), and the $\$IRC_j$ is the repair cost from the cost matrix of Equation (5.15).

$$EP = \begin{array}{c} 1 \\ 2 \\ 3 \\ j \\ . \\ J \end{array} \left[\begin{array}{cccccc} EP_{1,1} & EP_{1,2} & EP_{1,3} & EP_{1,4} & EP_{1,5} & EP_{1,6} \\ EP_{2,1} & EP_{2,2} & EP_{2,3} & EP_{2,4} & EP_{2,5} & EP_{2,6} \\ . & . & . & . & . & . \\ . & . & . & \boldsymbol{EP_{j,Y+1}} & . & . \\ . & . & . & . & . & . \\ EP_{J,1} & . & . & . & . & EP_{J,6} \end{array} \right] \tag{5.14}$$

$$\$IRC = \begin{array}{c|ccccc} & 1 & 2. & k & . & 5 \\ \hline 1 & \$IRC_{1,1} & \$IRC_{1,2} & \$IRC_{1,3} & \$IRC_{1,4} & \$IRC_{1,5} \\ 2 & \$IRC_{2,1} & \$IRC_{2,2} & \$IRC_{2,3} & \$IRC_{2,4} & \$IRC_{2,5} \\ 3 & . & . & . & . & . \\ \mathbf{j} & . & . & \boldsymbol{\$IRC_{jk}} & . & . \\ . & . & . & . & . & . \\ J & \$IRC_{J,1} & . & . & . & \$IRC_{J,5} \end{array} \tag{5.15}$$

Although the complexity in the integer formulations it produces less number of variables and constraints. Generally, the relationship between the variables and the objective function and between the variables and the constraint in both the binary and integer models is illustrated in Figure 5.3.

More details about solving IP problems can be found in Appendix A.

5.3.1 Experimenting with Alternative Formulations

Once the mathematical formulation for the binary and the integer models were made, the two alternative models were implemented in a spreadsheet format, as shown Figure 5.4 and Figure 5.5, respectively. Accordingly, various optimization tools, and strategies were experimented with, with respect to different sizes of networks.

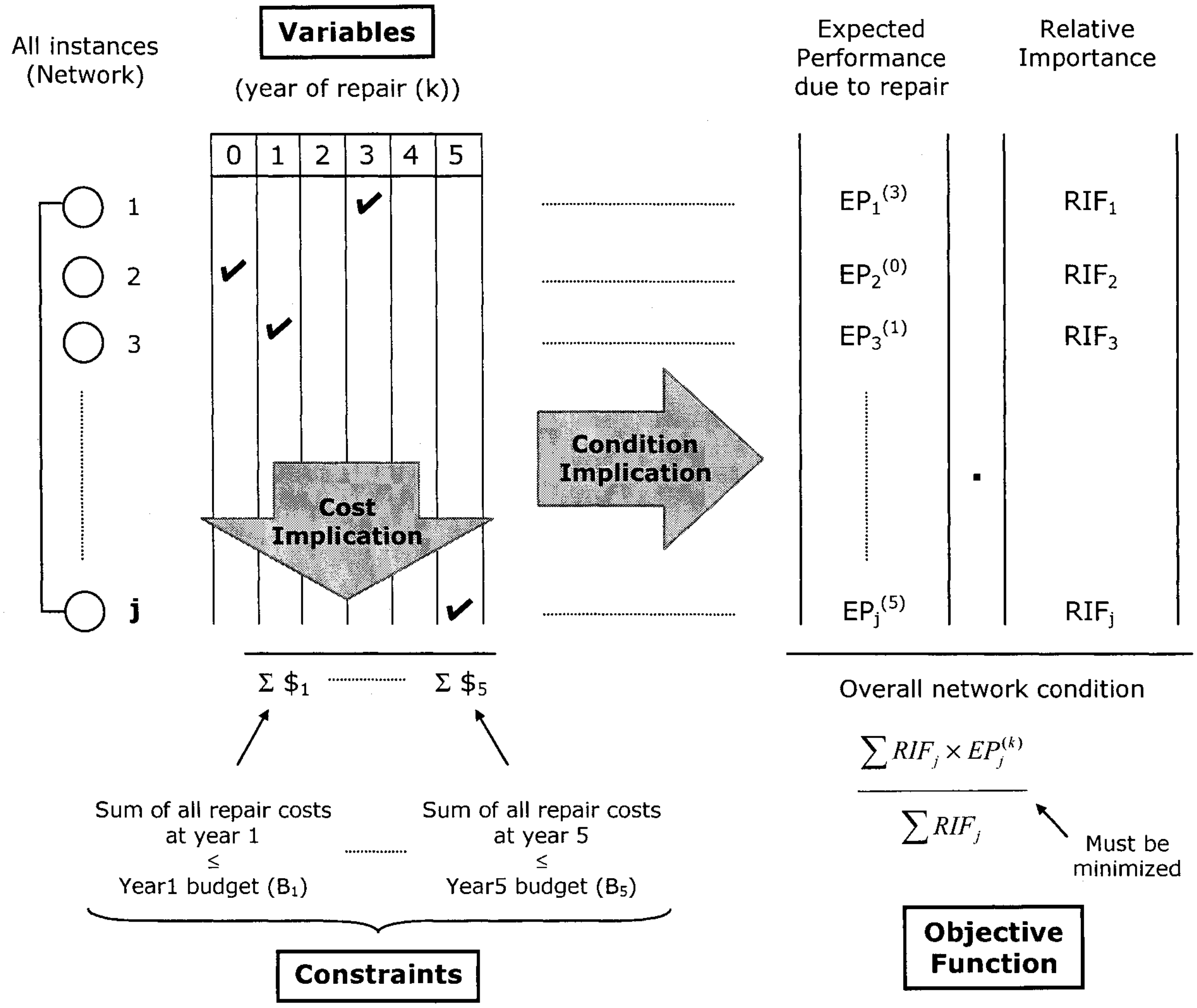
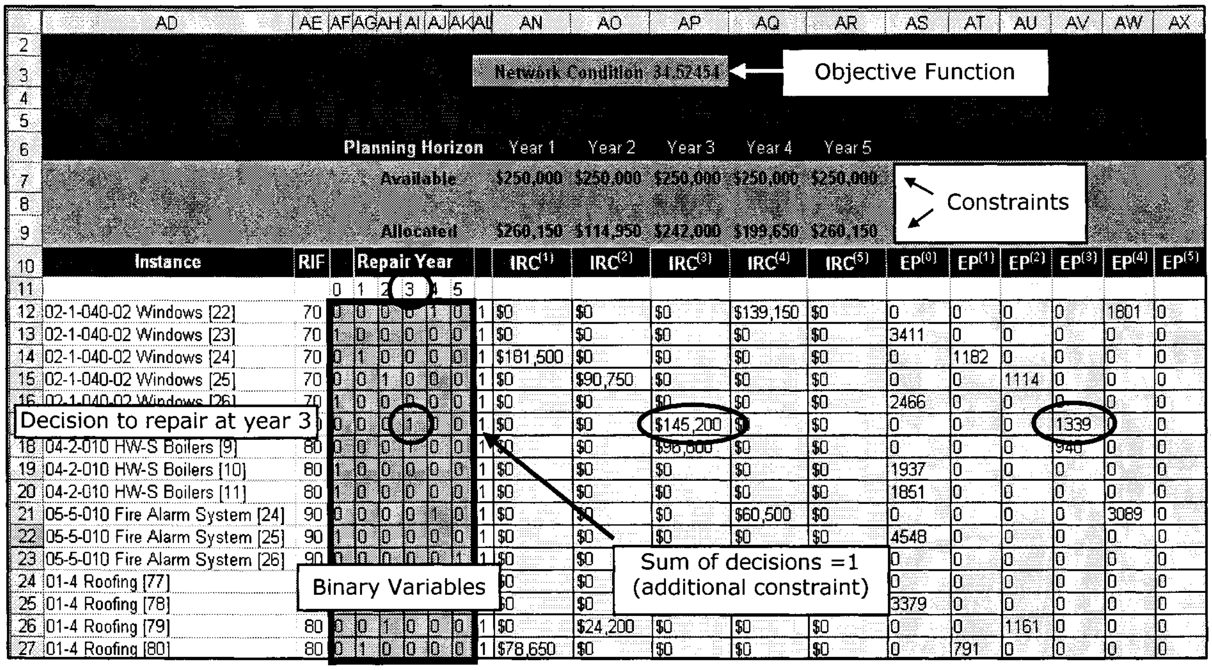


Figure 5.3: Optimization parameters for repair prioritization

In terms of implementation, the two models vary in the number of variables, constraints, and the level of complexity. Some implementation issues to the two models are as follows:

Integer Model: For each instance (row in the model), "LOOKUP" functions are used to determine the instance's data (best repair scenario, cost, and EP) corresponding to the value of the instance's repair year (k) in the variable cell. The data is obtained from the SOAP database. Once the proper costs and EPs corresponding to (k)s are determined for all instances, the spreadsheet performs two network calculations: (1) cost calculations for various years in the plan (i.e., 1, 2, 3, 4, and 5), using the conditional-sum function "SUMIF"; and (2) network condition calculation according to Equation (5.2).

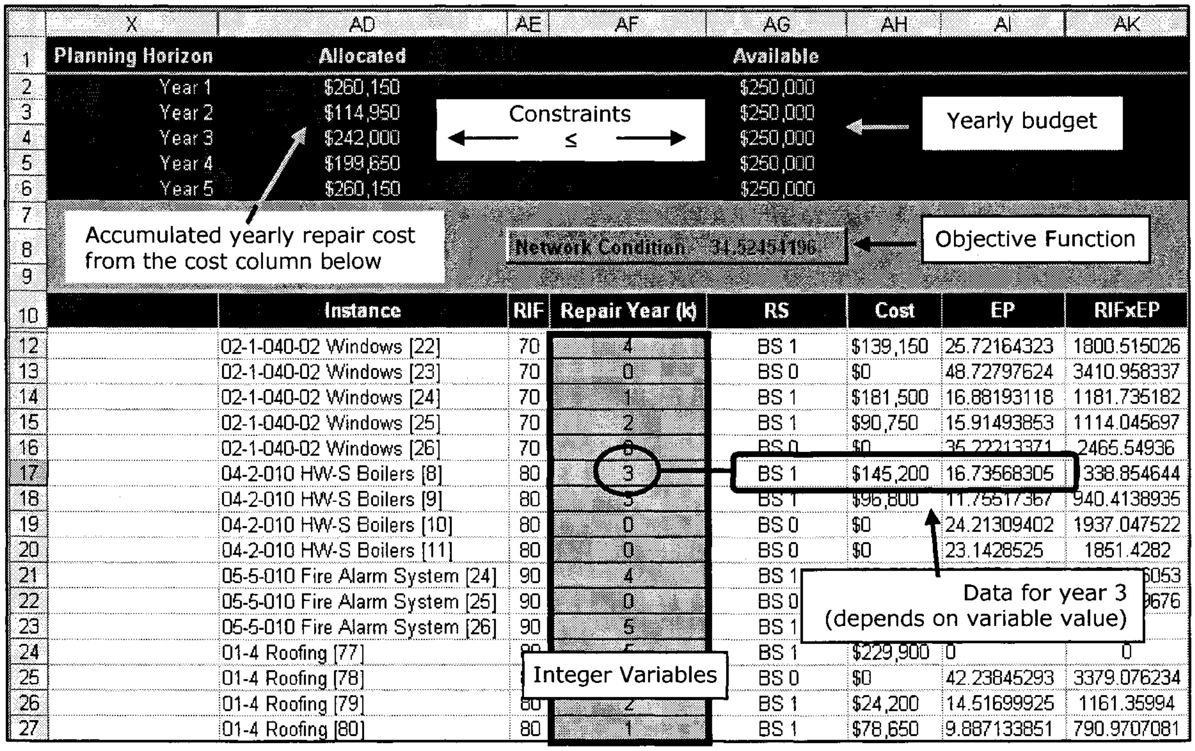


Network Condition 34.52454

Planning Horizon	Year 1	Year 2	Year 3	Year 4	Year 5
Available	$250,000	$250,000	$250,000	$250,000	$250,000
Allocated	$260,150	$114,950	$242,000	$199,650	$260,150

	Instance	RIF	Repair Year 0	1	2	3	4	5		IRC(1)	IRC(2)	IRC(3)	IRC(4)	IRC(5)	EP(0)	EP(1)	EP(2)	EP(3)	EP(4)	EP(5)
12	02-1-040-02 Windows [22]	70	0	0	0	0	1	0	1	$0	$0	$0	$139,150	$0	0	0	0	0	1801	0
13	02-1-040-02 Windows [23]	70	1	0	0	0	0	0	1	$0	$0	$0	$0	$0	3411	0	0	0	0	0
14	02-1-040-02 Windows [24]	70	0	1	0	0	0	0	1	$181,500	$0	$0	$0	$0	0	1182	0	0	0	0
15	02-1-040-02 Windows [25]	70	0	0	1	0	0	0	1	$0	$90,750	$0	$0	$0	0	0	1114	0	0	0
16	02-1-040-02 Windows [26]	70	1	0	0	0	0	0	1	$0	$0	$0	$0	$0	2466	0	0	0	0	0
17	[illegible]	[illegible]	0	0	0	1	0	0	1	$0	$0	$145,200	$0	$0	0	0	0	1339	0	0
18	04-2-010 HW-S Boilers [9]	80	0	0	0	1	0	0	1	$0	$0	$96,800	$0	$0	0	0	0	940	0	0
19	04-2-010 HW-S Boilers [10]	80	1	0	0	0	0	0	1	$0	$0	$0	$0	$0	1937	0	0	0	0	0
20	04-2-010 HW-S Boilers [11]	80	1	0	0	0	0	0	1	$0	$0	$0	$0	$0	1851	0	0	0	0	0
21	05-5-010 Fire Alarm System [24]	90	0	0	0	0	1	0	1	$0	$0	$0	$60,500	$0	0	0	0	0	3089	0
22	05-5-010 Fire Alarm System [25]	90	1	0	0	0	0	0	1	$0	$0	$0	$0	$0	4548	0	0	0	0	0
23	05-5-010 Fire Alarm System [26]	90	0	0	0	0	0	1	1	$0	$0	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0
24	01-4 Roofing [77]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	$0	$0	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0
25	01-4 Roofing [78]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	$0	$0	[illegible]	[illegible]	[illegible]	3379	0	0	0	0	0
26	01-4 Roofing [79]	80	0	0	1	0	0	0	1	$0	$24,200	$0	$0	$0	0	0	1161	0	0	0
27	01-4 Roofing [80]	80	0	1	0	0	0	0	1	$78,650	$0	$0	$0	$0	0	791	0	0	0	0

Figure 5.4: Binary formulation of optimization variables



	Planning Horizon	Allocated		Available
2	Year 1	$260,150		$250,000
3	Year 2	$114,950		$250,000
4	Year 3	$242,000		$250,000
5	Year 4	$199,650		$250,000
6	Year 5	$260,150		$250,000

Network Condition 34.52454196

	Instance	RIF	Repair Year (k)	RS	Cost	EP	RIFxEP
12	02-1-040-02 Windows [22]	70	4	BS 1	$139,150	25.72164323	1800.515026
13	02-1-040-02 Windows [23]	70	0	BS 0	$0	48.72797624	3410.958337
14	02-1-040-02 Windows [24]	70	1	BS 1	$181,500	16.88193118	1181.735182
15	02-1-040-02 Windows [25]	70	2	BS 1	$90,750	15.91493853	1114.045697
16	02-1-040-02 Windows [26]	70	0	BS 0	$0	35.22213371	2465.54936
17	04-2-010 HW-S Boilers [8]	80	3	BS 1	$145,200	16.73568305	1338.854644
18	04-2-010 HW-S Boilers [9]	80	5	BS 1	$96,800	11.75517367	940.4138935
19	04-2-010 HW-S Boilers [10]	80	0	BS 0	$0	24.21309402	1937.047522
20	04-2-010 HW-S Boilers [11]	80	0	BS 0	$0	23.1428525	1851.4282
21	05-5-010 Fire Alarm System [24]	90	4	BS 1	[illegible]	[illegible]	[illegible]
22	05-5-010 Fire Alarm System [25]	90	0	BS 0	[illegible]	[illegible]	[illegible]
23	05-5-010 Fire Alarm System [26]	90	5	BS 1	[illegible]	[illegible]	[illegible]
24	01-4 Roofing [77]	80	5	BS 1	$229,900	0	0
25	01-4 Roofing [78]	[illegible]	[illegible]	BS 0	$0	42.23845293	3379.076234
26	01-4 Roofing [79]	80	2	BS 1	$24,200	14.51699925	1161.35994
27	01-4 Roofing [80]	80	1	BS 1	$78,650	9.887133851	790.9707081

Figure 5.5: Integer formulation of optimization variables

With the use of lookup and conditional-sum functions the relationship between the source of data and the variables, the variables and the constraints, and the variables and the objective function become complex. However, the number of variables equals the number of instances in the network, while the number of constraints is only five (the budget constraints).

Binary Model: As opposed to the integer model, the binary model uses six cells for each instance to model the year of repair (one cell for each year option). These cells represent the variable cells and should be filled with 1s or 0s. As the selection of the year of repair should be a single value then, the sum of binary decisions must add to 1 (i.e., activating only one option). With this representation, the cost calculation for any year in the plan can be easily performed by multiplying the decision column for that year (holding 1s or 0s) by the corresponding cost column in the SOAP database, and then added up to represents the total spending in that year, as shown in Figure 5.4. Also, the network condition is calculated after the simple multiplication of the decision column by the corresponding (EP) column and the (RIF) column. The final network condition is calculated by adding up all weighted EPs then divide by the sum of (RIFs).

As explained, only simple multiplications and summations are used to in the binary model link the data, variables, constraints, and objective function, which greatly simplify the optimization process. However, the number of variables is increased six times than the integer model. Also, new constraints (equivalent to the number of instances, to ensure that the summation of decisions for each instance equal to 1) are added to the five budget constraints. This might limit the model's ability to handle large-scale networks.

As the search space for optimal or close-to-optimal solutions largely depends on the network size and the complexity of formulation, the two optimization models are expected to behave differently. The search space for the integer model can simply be expressed as 6^J as compared to 2^{6*J} for the binary model, where (J) represents the total number of instances in the network. With either the binary or the integer representation of variables, still the optimization problem is expected to be large. As such, in addition to tradition mathematical optimization, the genetic algorithms technique will be examined for its ability to arrive at near optimum solutions to large-scale problems. Also, an alternative step-wise strategy will be examined. In this strategy, a large-scale optimization will be applied only to each year of the planning horizon separately and consecutively. Comparison of the results for all these options in order to establish a reasonable strategy to handle the prioritization / fund allocation problem, will be discussed in Chapter 6, however, a summary of the results is provided in Table 5.1.

Table 5.1: Improvement for the Overall Network Condition

Tool	Size	Strategy	Integer Model		Binary Model	
			Condition Improvement	Comments	Condition Improvement	Comments
Mathematical Optimization using (Solver)	100 Instances	All-years	**Model is not suited for mathematical optimization**		From: 55 To: 29	600 variables
		Step-wise			From: 55 To: 29	5 optimizations of 200 variables
	200 Instances	All-years			From: 54 To: 31	1200 variables
		Step-wise			From: 54 To: 31	5 optimizations of 400 variables
	300 Instances	All-years			From: 54 To: 32	1800 variables
		Step-wise			From: 54 To: 31	5 optimizations of 600 variables
	400 Instances	All-years			**Failed** (too many variables, 2400)	
		Step-wise			From: 53 To: 31	5 optimizations of 800 variables
	800 Instances	All-years			**Failed** (too many variables, 4800)	
		Step-wise			From: 54 To: 32	5 optimizations of 1600 variables
Genetic Algorithms using (Evolver)	100 Instances	All-years	From: 55 To: 44	100 variables	From: 55 To: 39	600 variables
		Step-wise	From: 55 To: 29	5 optimizations Max. of 100 variables	From: 55 To: 31	5 optimizations of 200 variables
	800 Instances	All-years	From: 54 To: 44	800 variables	**Binary model is less beneficial with GAs, as such, is not used**	
		Step-wise	From: 54 To: 32	5 optimizations Max. of 800 variables		

As illustrated in Table 5.1, mathematical optimization using (Solver) couldn't handle 2400 variables using the all-years strategy, which means at step-wise optimization will soon fail to handle large networks. The GAs tool, on the other hand, shows the ability to provide acceptable near-optimal solutions, using the integer model with a step-wise strategy. Therefore, GAs is more reliable in optimizing the prioritization problem for large-scale networks of building components, and will be used with the integer formulation in the proposed framework for asset management.

5.4 Optimization with Decision Maker's Preferences

This section presents an approach to consider the decision maker's preferences and constraints (rather than the yearly budget) that might affect the repair prioritization process. As opposed to having all instances in the network in a single pool, this approach divides the network into sub-networks by applying sets of user-defined filtering criteria. The key benefit of having sub-networks is to make sure that a certain category of instances (e.g., buildings with historical values) receive reasonable funds without competition with the larger pool. In addition, each sub-network becomes small in size and the optimization can better allocate the repair funds.

The filtering criteria for creating sub-networks can relate to any attribute of any level in the asset hierarchy. For example, a sub-network can be created for all mechanical instances (at the system level). Another example could be the separation of the instances that show the exterior appearance of school buildings in tourist areas or close to locations where important events are usually held (e.g., Olympic games). The proposed sub-network approach, as such, enables school boards to respect political and economical considerations.

One important consideration in the sub-network approach is determining a suitable yearly budget limitation. Given a suitable criteria, the sub-network can be defined and often need to be dealt with at the beginning of the planning horizon (at year 1). Since year 1 repair costs represent the cheapest repair before further deterioration occurs, the portion of the total budget to be allocated to the sub-network can be simply calculated as the total repair needs of sub-network relative to that of the whole network, as follows.

$$\$b_n^{(k)} = \$B_N^{(k)} \times \frac{need\ \ for\ sub-network\ (n)\ at\ year1}{need\ \ for\ whole\ network\ (N)\ at\ year1} = \$B_N^{(k)} \times \frac{\sum\limits_{j'\in n} IRC_{j'}^{(1)}}{\sum\limits_{j\in N} IRC_j^{(1)}} \qquad (5.5)$$

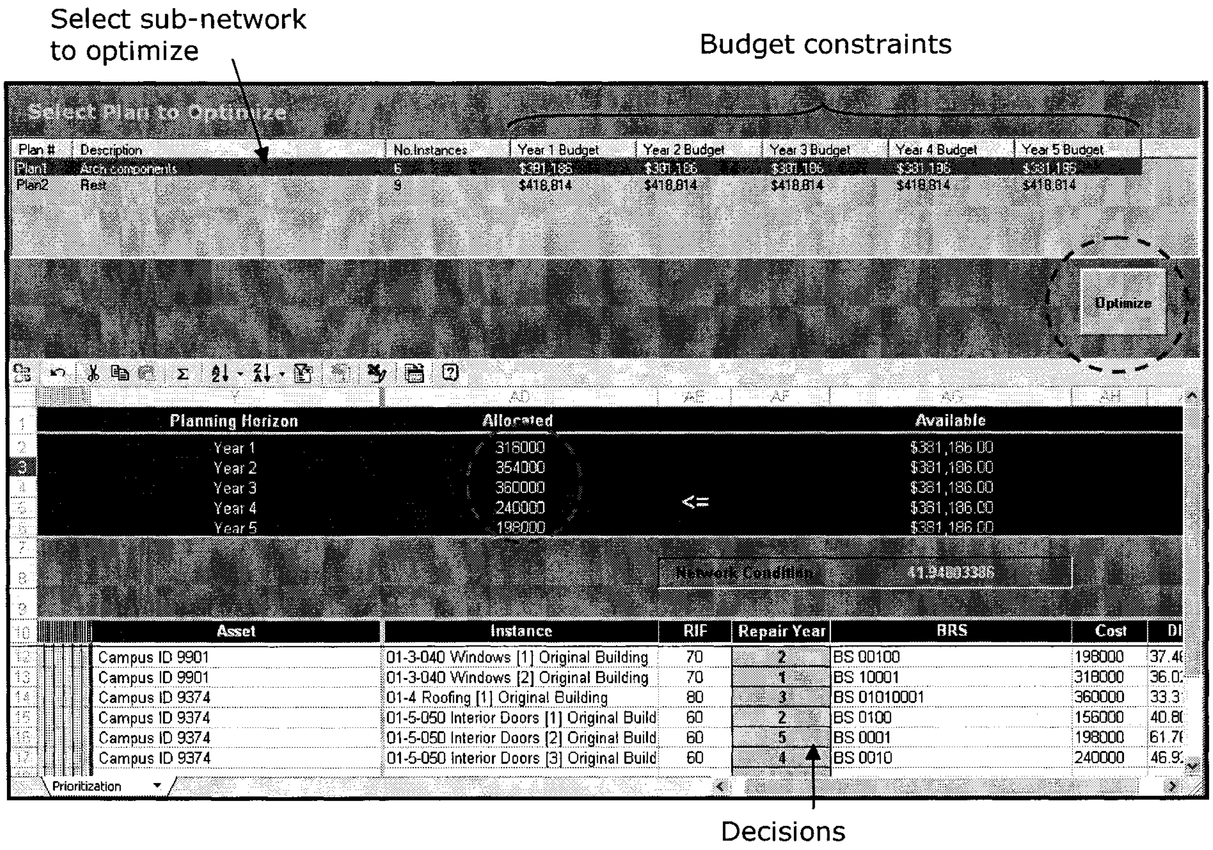
where, $\$b_n^{(k)}$ is the budget for sub-network (n) for year (k) in the planning horizon; $\$B_N^{(k)}$ is total budget for the whole network (N) for year (k) in the planning horizon; $IRC_{j'}^{(1)}$ is the cost of repair at year 1 for the sub-network related instance (j'); and $IRC_j^{(1)}$ is the cost of repair at year 1 for the whole network related instance (j).

For example, if the repair cost for all windows in year1 equals \$5,000,000, compared to \$200,000,000 to repair all components in year 1, then windows can be separated as a sub-network and given (5/200) from the available budgets at each year in the planning horizon. As such if a budget limit of \$50,000,000/ year is available, then windows will receive \$1,250,000 and the rest of component will receive \$48,750,000. Dealing with the windows sub-network separately, ensures that windows, not other components, will receive their share.

The optimization models discussed in the previous section is still valid with all formulations and can be used to improve the sub-network condition within available budgets for that sub-network.

5.5 Implementation: Asset Fund Allocation Program (A-FAP)

To implement the asset prioritization module, an Asset Fund Allocation Program (A-FAP) has been incorporated in the proposed framework. A-FAP is linked to SOAP output file and provides the user with user-friendly interface to set the yearly budget limits and any desirable sub-networks. A-FAP then creates an integer prioritization model for any selected sub-network on a separate sheet by adjusting the list of instances and yearly budgets. Also, it sets the ranges of variables and constraints to be used in the optimization, which is fully automated (Figure 5.6). More features and screen shots are provided later in Chapter 6.



Plan #	Description	No.Instances	Year 1 Budget	Year 2 Budget	Year 3 Budget	Year 4 Budget	Year 5 Budget
Plan1	Arch components	6	$381,186	$381,186	$381,186	$381,186	$381,186
Plan2	Rest	9	$418,814	$418,814	$418,814	$418,814	$418,814

Planning Horizon	Allocated		Available
Year 1	318000		$381,186.00
Year 2	354000		$381,186.00
Year 3	360000		$381,186.00
Year 4	240000	<=	$381,186.00
Year 5	198000		$381,186.00

Network Condition: 41.94803386

Asset	Instance	RIF	Repair Year	BRS	Cost	DI
Campus ID 9901	01-3-040 Windows [1] Original Building	70	2	BS 00100	198000	37.4
Campus ID 9901	01-3-040 Windows [2] Original Building	70	1	BS 10001	318000	36.0
Campus ID 9374	01-4 Roofing [1] Original Building	80	3	BS 01010001	360000	33.3
Campus ID 9374	01-5-050 Interior Doors [1] Original Build	60	2	BS 0100	156000	40.8
Campus ID 9374	01-5-050 Interior Doors [2] Original Build	60	5	BS 0001	198000	61.7
Campus ID 9374	01-5-050 Interior Doors [3] Original Build	60	4	BS 0010	240000	46.9

Figure 5.6: Asset Fund Allocation Program (A-FAP)

5.6 Conclusions

Various formulations for network-level optimization are presented in this chapter to consider for the possible variable representations, problem size considerations, and decision maker's preferences in the fund allocation process. Details of the implementations of these models and a case study are presented in the next chapter.

Chapter 6

INTEGRATION AND TESTING

6.1 Introduction

In the previous three chapters, models for condition assessment, deterioration prediction, repair selection, and asset prioritization were introduced and discussed. These models form the basis for a comprehensive framework for asset management that suit the building infrastructure. In this chapter, an implementation prototype program, D-TRACKER, is introduced which combines and integrates the various models of the proposed framework. Features of D-Tracker are presented and a case study is used to validate the prototype and demonstrate its usefulness.

6.2 D-TRACKER: A Building Asset Management Framework

The proposed asset management framework, presented in the previous three chapters, has unique focus on tracking the dynamics of defects in various building components and on optimally repairing these defects. The condition assessment accurately evaluates the current severity of defects associated with the inspected components. Based on the current condition assessment, the component-dependent deterioration model of the framework predicts the future severities (defects) for each year in the planning horizon. As such, both the present and the future defects are tracked by the framework, and used by the repair model to create practical repair options. With its dependence on deficiencies, a Deficiency Tracker (D –TRACKER) prototype system has been developed as a comprehensive asset management system. D-Tracker is a Visual Basic application program which is designed to be customizable to any organizational needs in terms of the asset hierarchy, the deficiency lists, and the inventory of components. Once D-Tracker is adapted for any organization, it can be used to manage the assets for repair and rehabilitation purposes. D-Tracker has modules for condition assessment, Life-cycle-cost analysis (deterioration prediction and repair optimization), and asset prioritization (Figure 6.1). The main screen of D-Tracker is shown in Figure 6.2, which provides access to the various functions of D-Tracker, as explained in the following subsections.

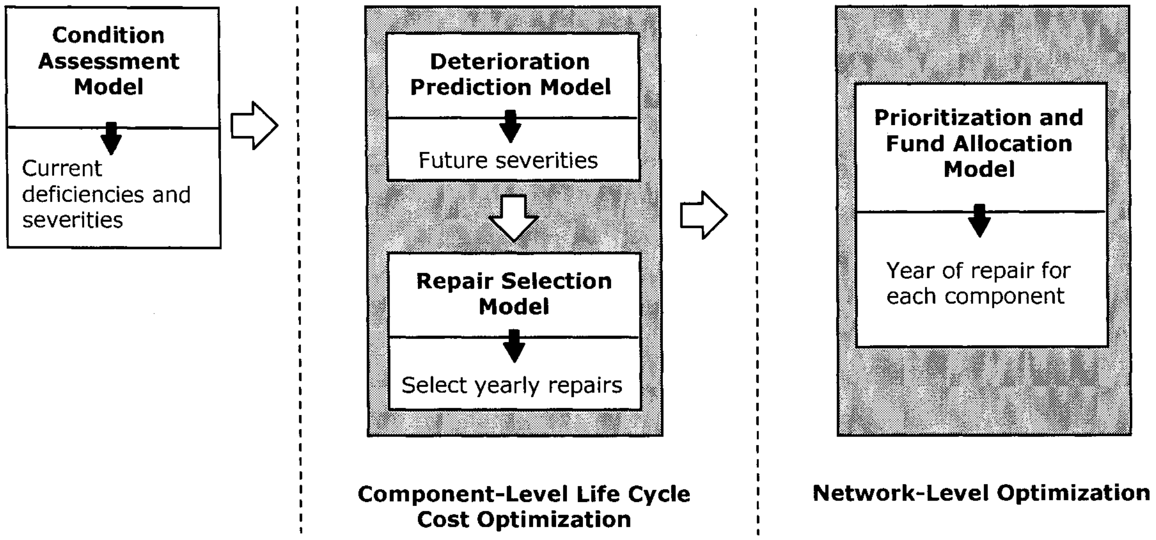

Condition Assessment Model
Current deficiencies and severities
Deterioration Prediction Model
Future severities
Repair Selection Model
Select yearly repairs
Prioritization and Fund Allocation Model
Year of repair for each component
Component-Level Life Cycle Cost Optimization
Network-Level Optimization

Figure 6.1: D-TRACKER design

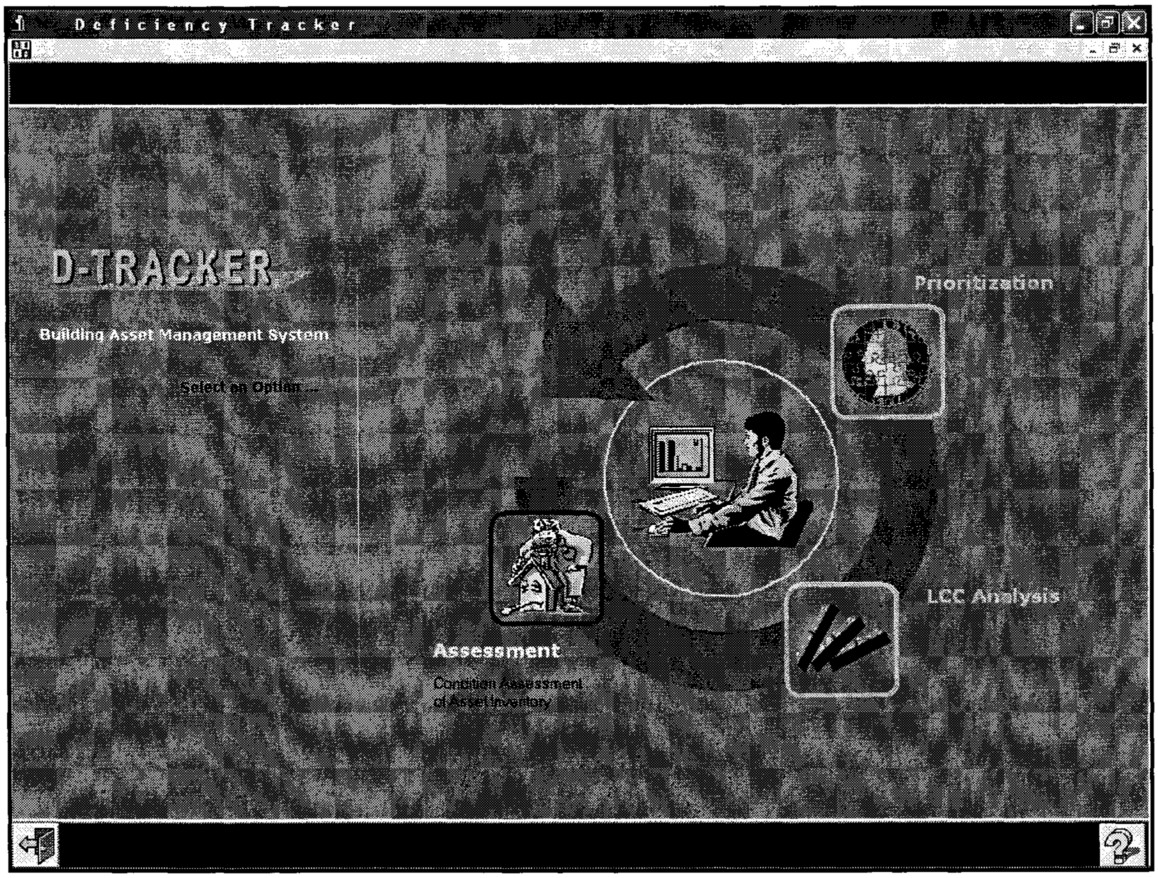


Figure 6.2: D-TRACKER main options

6.2.1 Condition Assessment Module

The condition assessment module uses the Visual Condition Assessment Program (V-CAP) described in Chapter 3, which handles three main tasks: (1) inspection; (2) revision; and (3) reporting (Figure 6.3), as follows:

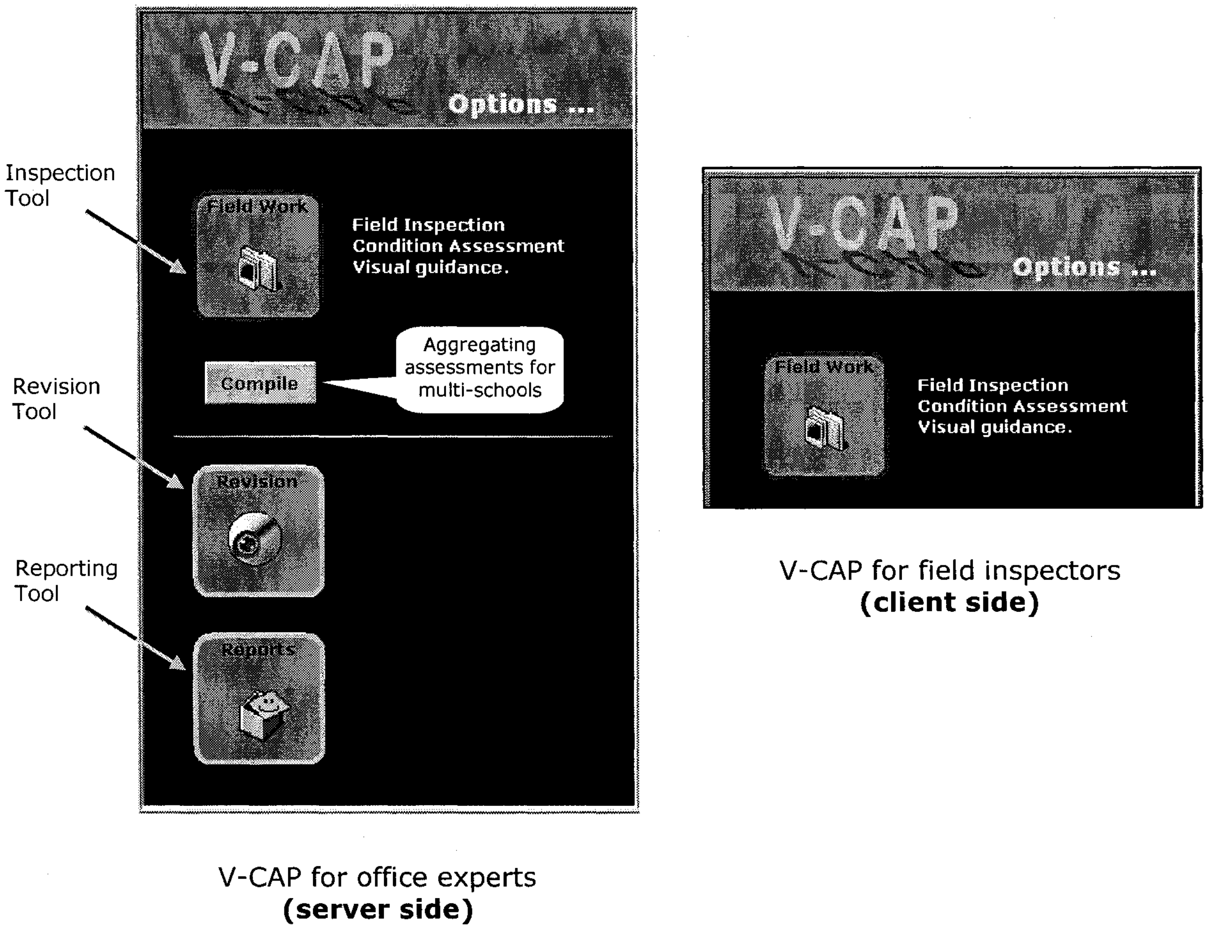
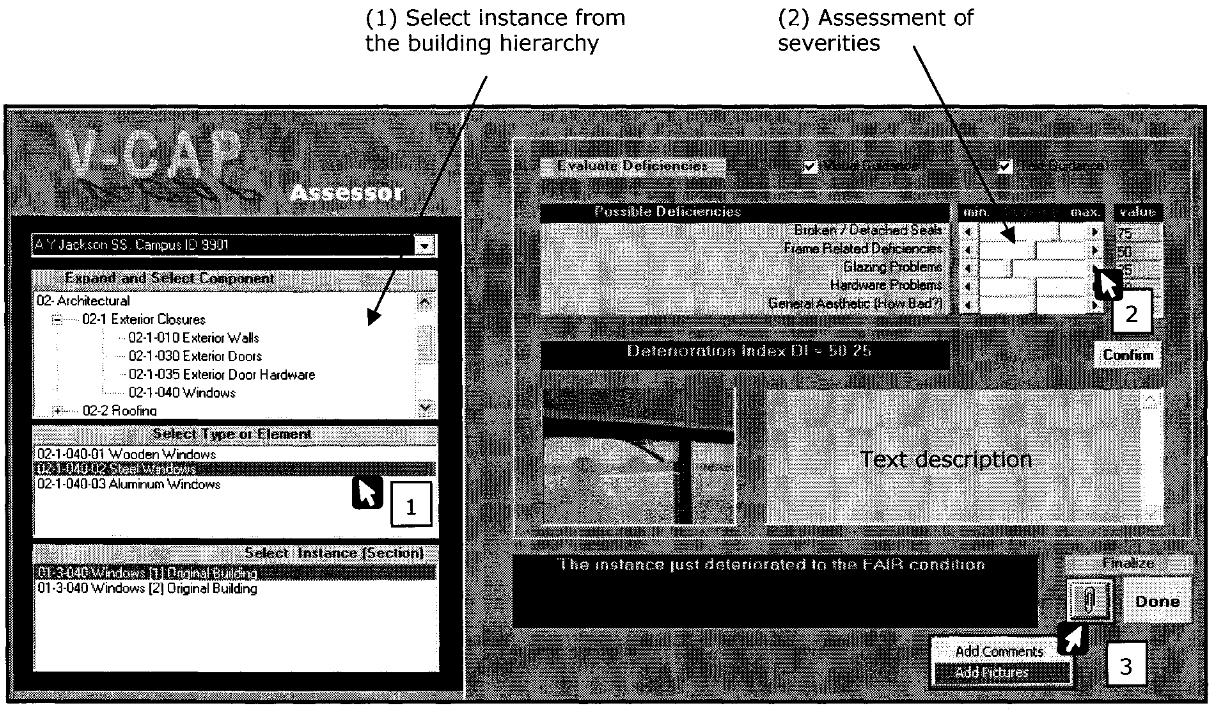

Inspection Tool
Revision Tool
Reporting Tool
V-CAP
Options ...
Field Work
Field Inspection Condition Assessment Visual guidance.
Compile
Aggregating assessments for multi-schools
Revision
Reports
V-CAP for office experts (server side)
V-CAP
Options ...
Field Work
Field Inspection Condition Assessment Visual guidance.
V-CAP for field inspectors (client side)

Figure 6.3: V-CAP functions

1. V-CAP Inspection Tool: this tool is mainly for inspectors to perform field inspections with the help of the visual guidance system discussed in Chapter 3. Inspectors use a client version of the condition assessment module and then send the assessment files to the office experts who have a server version that allows them to compile the inspection files for various buildings (Figure 6.3). The inspection tool allows the inspector to start a new inspection, save inspection files, and continue working on an existing inspection. To minimize inspection errors, the program is designed with a user-friendly interface for the user to follow the building hierarchy to select the instances for inspection (Figure 6.4).



(1) Select instance from the building hierarchy

(2) Assessment of severities

V-CAP Assessor

A Y Jackson SS, Campus ID 9901

Expand and Select Component

02- Architectural
02-1 Exterior Closures
02-1-010 Exterior Walls
02-1-030 Exterior Doors
02-1-035 Exterior Door Hardware
02-1-040 Windows
02-2 Roofing

Select Type or Element

02-1-040-01 Wooden Windows
02-1-040-02 Steel Windows
02-1-040-03 Aluminum Windows

Select Instance (Section)

01-3-040 Windows [1] Original Building
01-3-040 Windows [2] Original Building

Evaluate Deficiencies

☑ Visual Guidance ☑ Text Guidance

Possible Deficiencies	min. max.	value
Broken / Detached Seals		75
Frame Related Deficiencies		50
Glazing Problems		25
Hardware Problems		
General Aesthetic (How Bad?)		

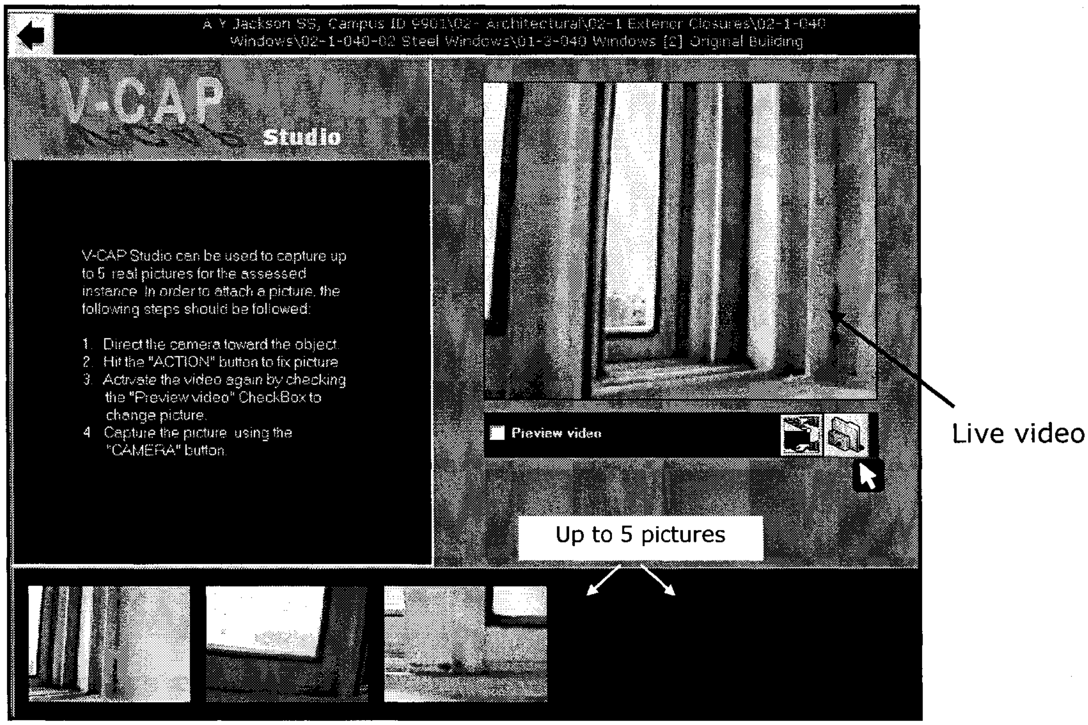
Deterioration Index DI = 50 25

Text description

The instance just deteriorated to the FAIR condition

Figure 6.4: Selecting an instance and assessing its deficiencies

During the inspection, the program allows the user to attach up to five real pictures for each inspected instance, in addition to text comments, as shown in Figure 6.5. Once the inspection is completed for all components in the building, the program provides an e-mail feature for inspectors to send the inspection files to the main office (Figure 6.6).

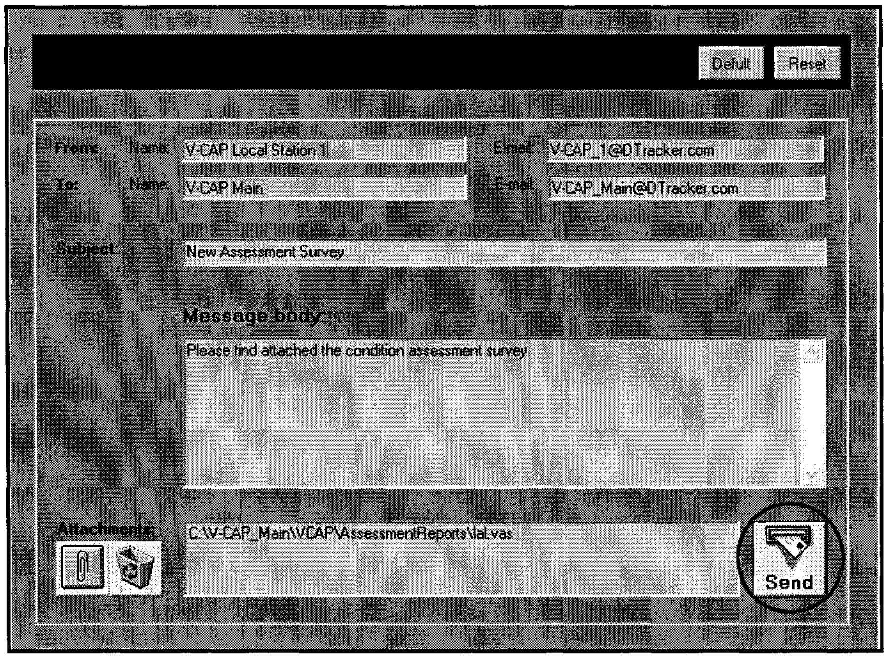


A Y Jackson SS, Campus ID 9901\02- Architectural\02-1 Exterior Closures\02-1-040 Windows\02-1-040-02 Steel Windows\01-3-040 Windows [2] Original Building

V-CAP Studio can be used to capture up to 5 real pictures for the assessed instance. In order to attach a picture, the following steps should be followed:

1. Direct the camera toward the object.
2. Hit the "ACTION" button to fix picture
3. Activate the video again by checking the "Preview video" CheckBox to change picture.
4. Capture the picture using the "CAMERA" button.

Figure 6.5: V-CAP Studio



From: Name: V-CAP Local Station 1 E-mail: V-CAP_1@DTracker.com

To: Name: V-CAP Main E-mail: V-CAP_Main@DTracker.com

Subject: New Assessment Survey

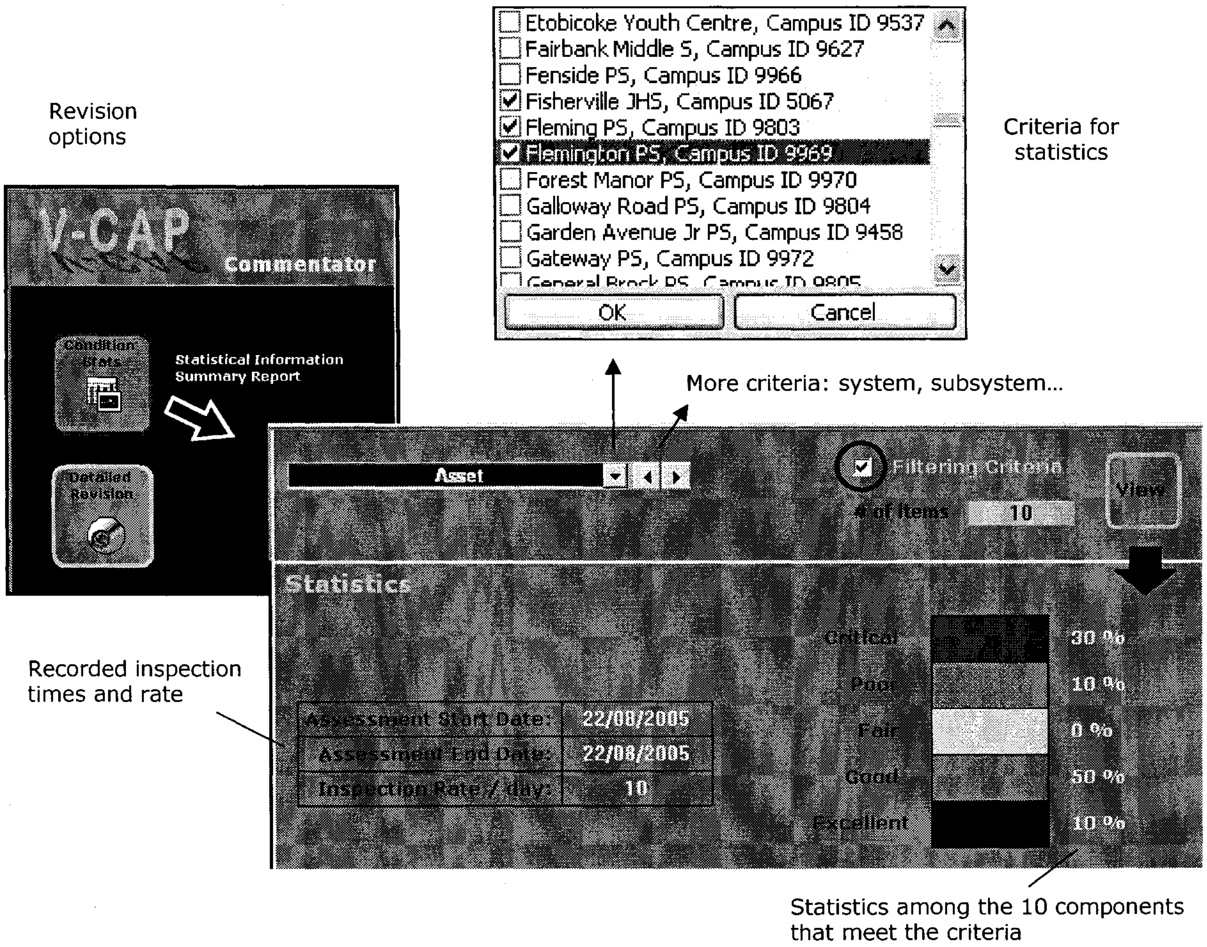
Message body:

Please find attached the condition assessment survey

Attachments: C:\V-CAP_Main\VCAP\AssessmentReports\lal.vas

Figure 6.6: Sending inspection data to main office as e-mail attachment

2. V-CAP Revision Tool: this tool is for managers and office users only. This tool enables managers to view / check / modify the compiled inspection data. Based on the various inspection files received, the program performs useful statistics about the inspected components (Figure 6.7).



Revision options

☐ Etobicoke Youth Centre, Campus ID 9537
☐ Fairbank Middle S, Campus ID 9627
☐ Fenside PS, Campus ID 9966
☑ Fisherville JHS, Campus ID 5067
☑ Fleming PS, Campus ID 9803
☑ Flemington PS, Campus ID 9969
☐ Forest Manor PS, Campus ID 9970
☐ Galloway Road PS, Campus ID 9804
☐ Garden Avenue Jr PS, Campus ID 9458
☐ Gateway PS, Campus ID 9972
☐ General Brock PS, Campus ID 9805

Criteria for statistics

More criteria: system, subsystem...

Assessment Start Date:	22/08/2005
Assessment End Date:	22/08/2005
Inspection Rate / day:	10

Critical	30 %
Poor	10 %
Fair	0 %
Good	50 %
Excellent	10 %

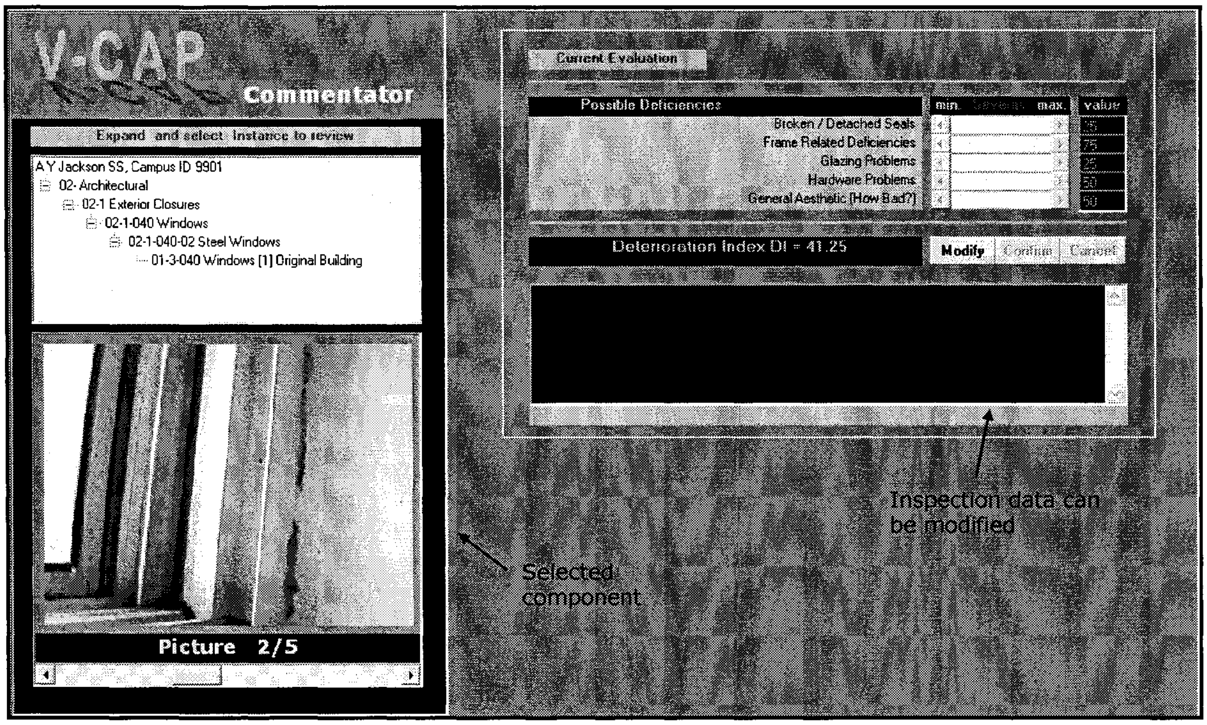
Recorded inspection times and rate

Statistics among the 10 components that meet the criteria

Figure 6.7: Inspection statistics

For example, the inspection data for the three selected schools in Figure 6.7 shows that 50% of their components are in good condition. In addition to condition statistics, the program shows the inspection start and finish times (for the filtered assts). Accordingly, it calculates and shows the inspection rate that indicates the productivity of the inspectors. The ability to filter inspection data at any level makes it possible to only check a certain inspector, component, and/or system, for example.

In addition to the statistics option, the program allows the user in the office to view all inspection data such as pictures, text comments, and severities evaluations. Accordingly, the user can add / change any inspection / assessment data (Figure 6.8).



A Y Jackson SS, Campus ID 9901
- 02- Architectural
  - 02-1 Exterior Closures
    - 02-1-040 Windows
      - 02-1-040-02 Steel Windows
        - 01-3-040 Windows [1] Original Building

Possible Deficiencies	min. Severity max.	value
Broken / Detached Seals		35
Frame Related Deficiencies		75
Glazing Problems		25
Hardware Problems		50
General Aesthetic [How Bad?]		50

Deterioration Index DI = 41.25

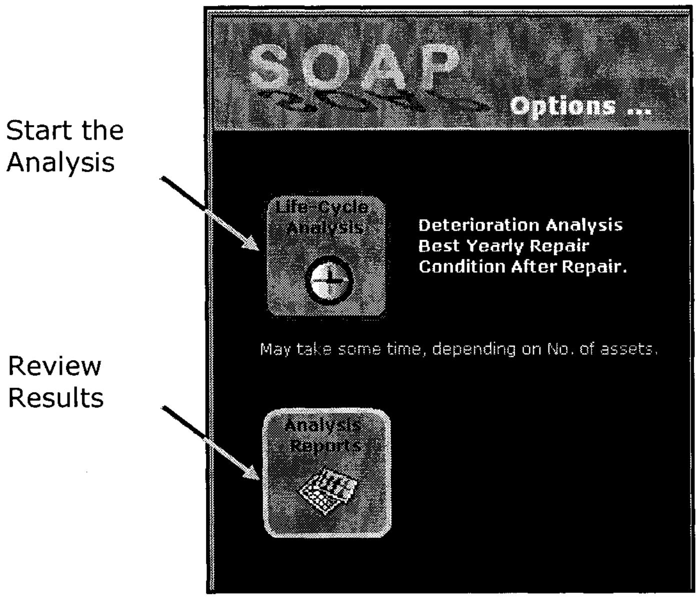
Figure 6.8: Revision tool

Once revision is completed, the user approves the inspection data, which become ready for further use by other system modules.

3. **V-CAP Reporting Tool:** it exports inspection data to an external Excel sheet.

6.2.2 LCC Analysis Module

The LCC analysis module uses the Sequential Optimization Analysis Program (SOAP) discussed in Chapter 4, which provides two options, as shown in Figure 6.9:



Start the Analysis

Review Results

Deterioration Analysis
Best Yearly Repair
Condition After Repair.

May take some time, depending on No. of assets.

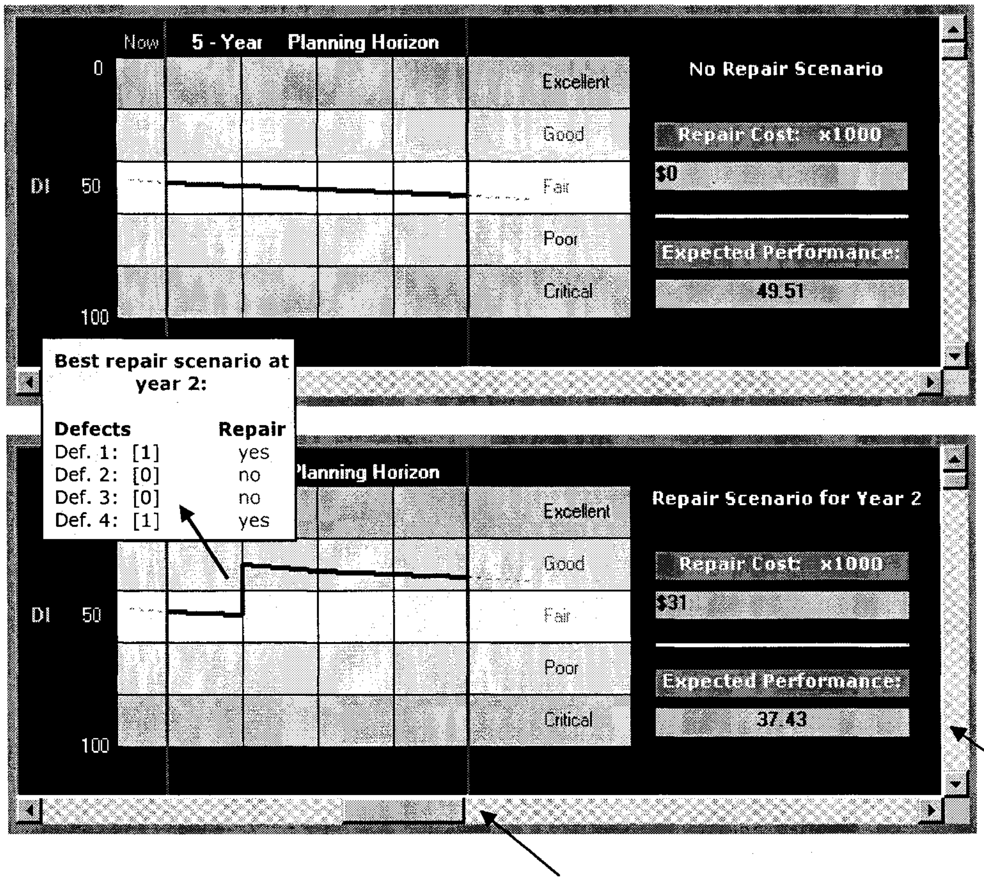
Figure 6.9: SOAP main options

1. Start the LCC Analysis: the first option is to start loading a compiled inspection file after revision and approval in order to perform the LCC analysis. The analysis is done by the sequential optimizations discussed in Chapter 4. Once the analysis is done, the program saves an analysis as a file that is ready for further steps; and

2. Review already Analyzed File: Figure 6.10 shows some of the graphical analysis results of SOAP program, which can be exported to MS Excel.

For a selected instance, the figure shows the calculated deterioration curves for the two cases of no-repair during the planning horizon (top) and the case of repair in year 2 using the least costly option in that year. The no-repair curve shows that the condition will remain as "Fair" at the end of the five years while it will improve to "Good" in the second strategy. It is noted that, the repair in the second year improves the expected performance from 49.5 to 37.4 at a cost of \$31k. As shown in the screen in Figure 6.10, the horizontal scroll bar can be used to view the impact of repair strategies at different years while the vertical scroll bar can be used to view other instances.

As a summary of the analysis results over the planning horizon, Figure 6.11 shows another graphical analysis result of SOAP, which shows the impact of repair year on both expected performance and cost.



No Repair Scenario

Repair Cost: x1000

$0

Expected Performance:

49.51

Best repair scenario at year 2:

Defects	Repair
Def. 1: [1]	yes
Def. 2: [0]	no
Def. 3: [0]	no
Def. 4: [1]	yes

Repair Scenario for Year 2

Repair Cost: x1000

$31

Expected Performance:

37.43

Deterioration curve for the case of no repair

Deterioration curve for the case of repair at year 2

Check other components

Change year of repair

Figure 6.10 SOAP Results

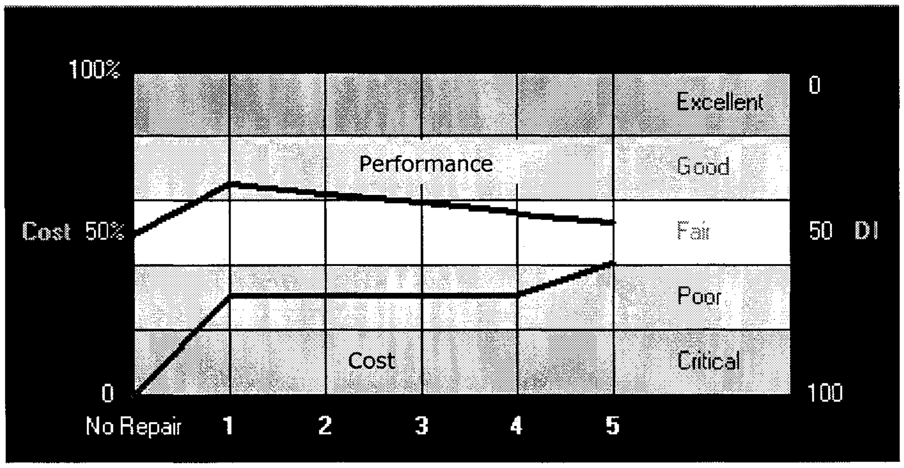


Figure 6.11: SOAP summary charts

The expected performance curve in Figure 6.11 indicates that whether the repair is performed in the first year or the second year, the expected performance will remain in "Good" condition. However, if repair is delayed after that, the performance will decrease to "Fair". Also, the cost curve shows that the cost of repair will not change if repair is performed between the first and the fourth year, then it will increase. As a conclusion, if the available funds in year one is not enough, this instance can be planned for repair in the second year without noticeable changes to the cost of repair or to the expected performance. Also, if the performance is acceptable as "Fair", then the no-repair option should be selected as it saves the repair cost.

6.2.3 Prioritization Module

The prioritization module uses the Asset Fund Allocation Program (A-FAP) discussed in Chapter 5. A-FAP main options are shown in Figure 6.12. When activated to create a 5-year plan, A-FAP presents the user with two screens to enter the yearly budget limit, and specifying sub-networks that reflect decision maker's preferences in fund allocation. Figure 6.13 shows the screen for setting the yearly budget limits with a yearly contingency percentage if desired. If the user selects to create sub-networks from the radial buttons in the top, the second screen for creating sub-networks appears as shown in Figure 6.14. The user then can select the criteria to filter all available components to create the sub set of desirable characteristics. Figure 6.14 for example, shows a filter to select all the architectural components in a sub-network.

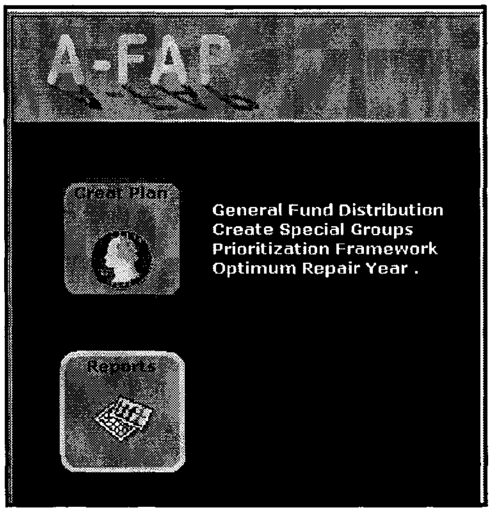
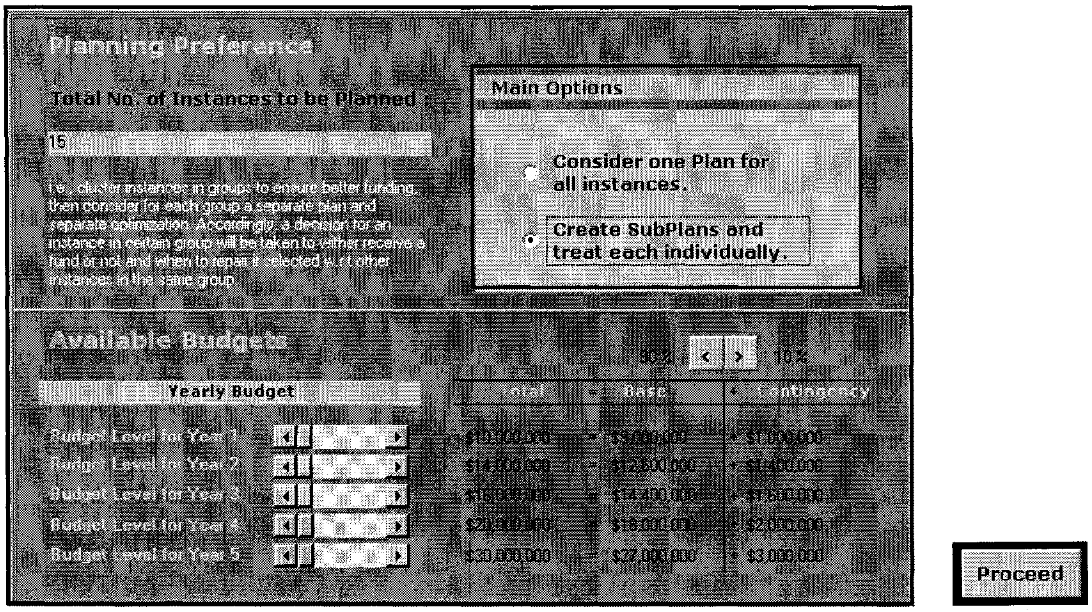


Figure 6.12: A-FAP options



Planning Preference

Total No. of Instances to be Planned :

15

i.e., cluster instances in groups to ensure better funding, then consider for each group a separate plan and separate optimization. Accordingly, a decision for an instance in certain group will be taken to wither receive a fund or not and when to repair if selected w.r.t other instances in the same group.

Main Options

○ Consider one Plan for all instances.

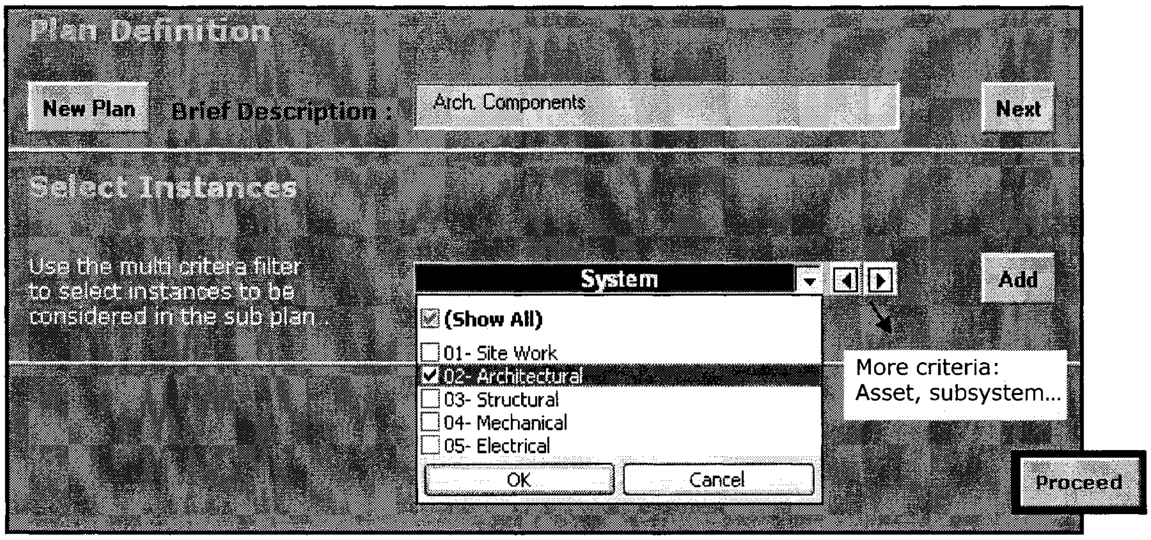
● Create SubPlans and treat each individually.

Available Budgets

90 % < > 10 %

Yearly Budget	Total	= Base	+ Contingency
Budget Level for Year 1	$10,000,000	= $9,000,000	+ $1,000,000
Budget Level for Year 2	$14,000,000	= $12,600,000	+ $1,400,000
Budget Level for Year 3	$16,000,000	= $14,400,000	+ $1,600,000
Budget Level for Year 4	$20,000,000	= $18,000,000	+ $2,000,000
Budget Level for Year 5	$30,000,000	= $27,000,000	+ $3,000,000

Figure 6.13: A-FAP budget and preferences settings



Plan Definition

Brief Description : Arch. Components

Select Instances

Use the multi critera filter to select instances to be considered in the sub plan..

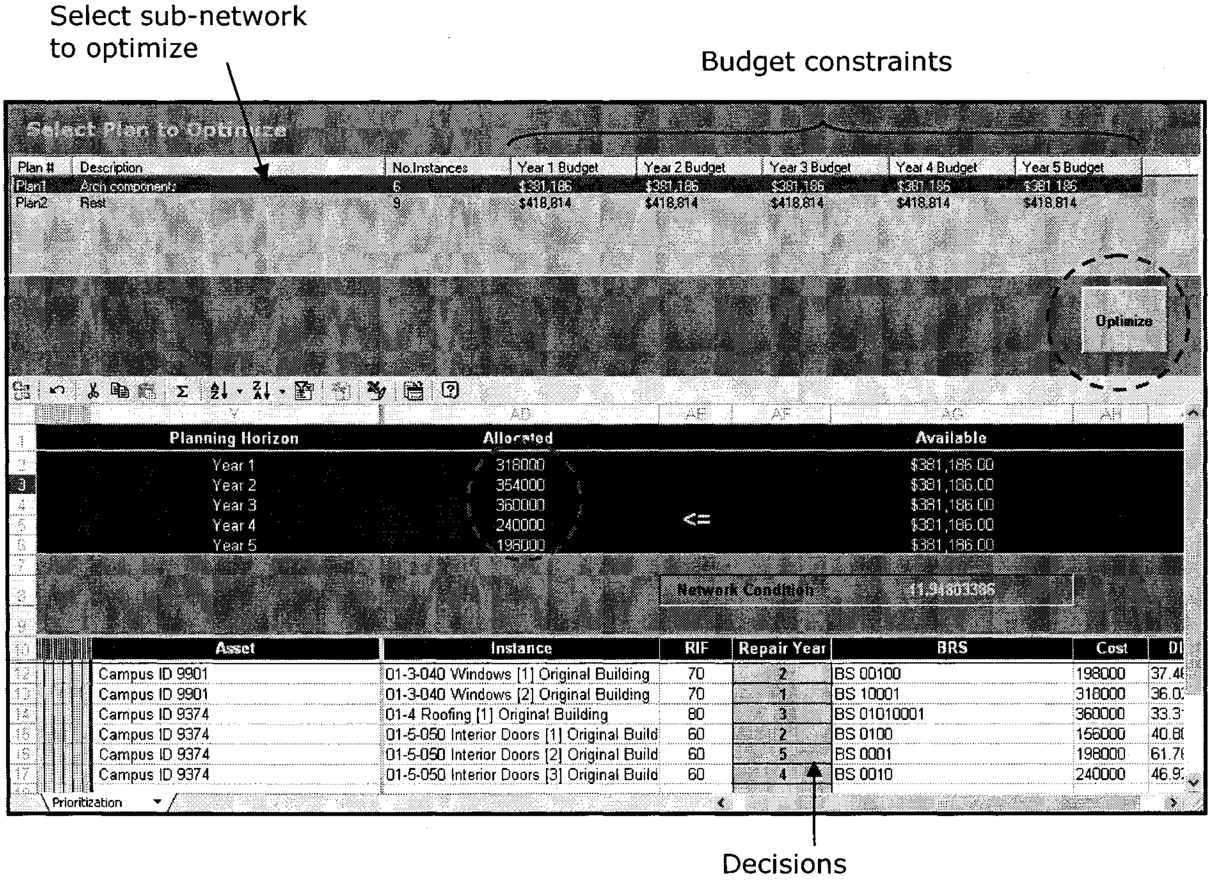
System

☑ (Show All)
☐ 01- Site Work
☑ 02- Architectural
☐ 03- Structural
☐ 04- Mechanical
☐ 05- Electrical

More criteria: Asset, subsystem...

Figure 6.14: Creating sub-networks

Once all the desired sub-networks are created the user can proceed with the optimization and accordingly, a spreadsheet model will be set up with the variables, constraints, and objective function that suit each sub-network individually. Afterwards, the optimization runs in an automated manner to determine near-optimum decisions for the repair years and accumulated costs. The example in Figure 6.15 shows the decision for a sub-network of six instances where the column of allocated costs meets the available yearly budget.



Plan #	Description	No.Instances	Year 1 Budget	Year 2 Budget	Year 3 Budget	Year 4 Budget	Year 5 Budget
Plan1	Arch components	6	$381,186	$381,186	$381,186	$381,186	$381,186
Plan2	Rest	9	$418,814	$418,814	$418,814	$418,814	$418,814

Planning Horizon	Allocated		Available
Year 1	318000		$381,186.00
Year 2	354000		$381,186.00
Year 3	360000	<=	$381,186.00
Year 4	240000		$381,186.00
Year 5	198000		$381,186.00

Network Condition 11.94303386

Asset	Instance	RIF	Repair Year	BRS	Cost	DI
Campus ID 9901	01-3-040 Windows [1] Original Building	70	2	BS 00100	198000	37.4
Campus ID 9901	01-3-040 Windows [2] Original Building	70	1	BS 10001	318000	36.0
Campus ID 9374	01-4 Roofing [1] Original Building	80	3	BS 01010001	360000	33.3
Campus ID 9374	01-5-050 Interior Doors [1] Original Build	60	2	BS 0100	156000	40.8
Campus ID 9374	01-5-050 Interior Doors [2] Original Build	60	5	BS 0001	198000	61.7
Campus ID 9374	01-5-050 Interior Doors [3] Original Build	60	4	BS 0010	240000	46.9

Figure 6.15: Optimization model for prioritization and fund allocation

6.3 Case Study

To experiment with the developed framework, real data obtained from one of the largest school boards in North America have been used. The data relate to 14,857 instances that are planned to be repaired from 2003 to 2007. For this large number of instances, the data obtained for each instance is as follows:

- Building ID (for the school holding the instance);
- Construction year for the building;
- Instance ID;
- Priority value for repair purposes (calculated by the asset management system used);
- Suggested year of repair (2003 – 2007, suggested by the inspector during the field inspection);
- Estimated cost of repair; and
- Text description for the instance condition and repair needs.

Various comments can be made on these data. First, detailed condition assessment data is not provided either in direct rating (good, poor, etc) or in severities of defects. Rather, a calculated priority value was provided (depends in part on the condition assessment which was not given). Second, despite of the fact that the system used by the school board is one of the sophisticated and commonly used large software systems, the year of repair is subjectively determined by inspectors in the field, without a structured mechanism that considers deterioration and repair needs.

6.3.1 Testing of Available Data

The received data have been extensively analyzed before being used with the proposed system. By examining the text comments of the instances, it was noticed that inspectors suggest one of three repair related actions: (1) replacement; (2) repair; and (3) further analysis. The frequency of these actions is shown in Figure 6.16, clearly indicating that the asset management system uses a replacement strategy for repairs, regardless of the suggested year of repair or the calculated priority value.

Discussing the obtained data with facility managers at the school board, they indicate that the provided data are sufficient to create a yearly repair plan by considering the instances that are suggested to be repaired in each year (e.g., 2004), then sorting these instances in a descending order by their priority values for fund allocation. The highest priority instances are allocated their repair needs and so on until the available budget in that year is exhausted.

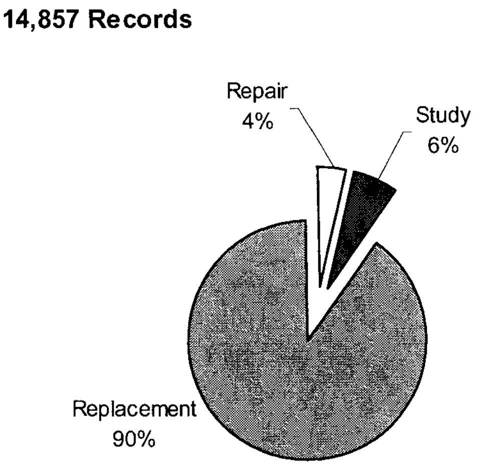


Figure 6.16: Repair types provided in the provided data

As explained, the priority values (which partially depend on condition) are the basis for funding decisions. As such, these priority values are examined for being logical in terms of age versus repair representation and fund allocation use. These are discussed as follows:

1. Relationships among Condition, Age, and Repair Year: It is expected that an instance with high priority value to be planned early for repair and vise versa. To examine this logical relationship, Figure 6.17 shows the inherent relationship between priority value and year-of-repair in the data. Unexpectedly, the data do not exhibit a logical trend. This shows that the lack of link between the manners by which the priority values and the year-of-repair are determined.

Another logical relationship is examined in the data, which is the age versus priority value. It is expected that as component ages, its priority value should get higher (the shaded trend in Figure 6.18). The relationship between the age of instances and the priority value and between the age and the repair year are then plotted, as shown in Figures 6.18 and 6.19, respectively. As shown in the previous three figures, the data still do not exhibit logical trends.

From Figures 6.17 to 6.19, it is clear that the priority value is calculated without respecting the instance age or expected repair year. This implies, it can not reflect the deterioration behavior of components.

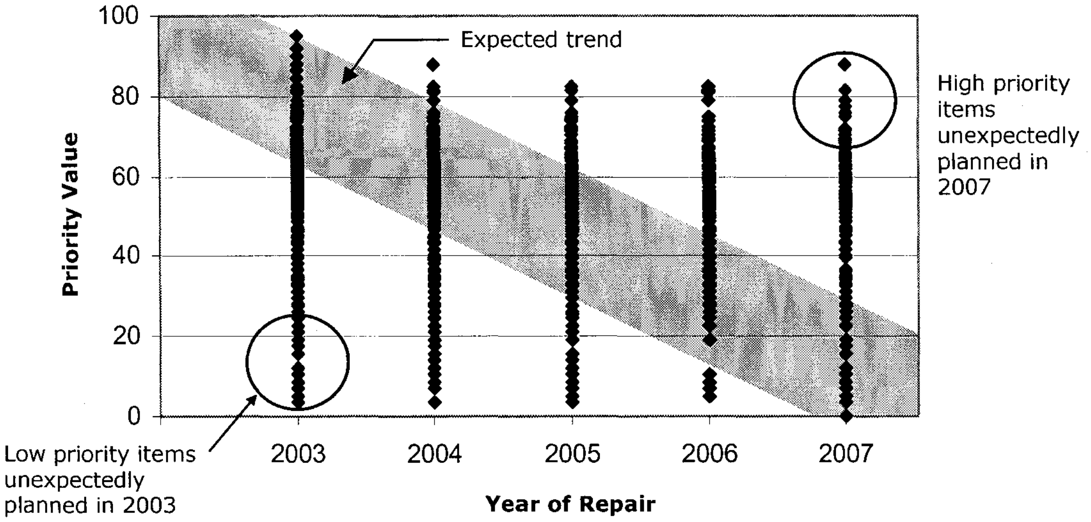


Figure 6.17: Repair year vs. priority value

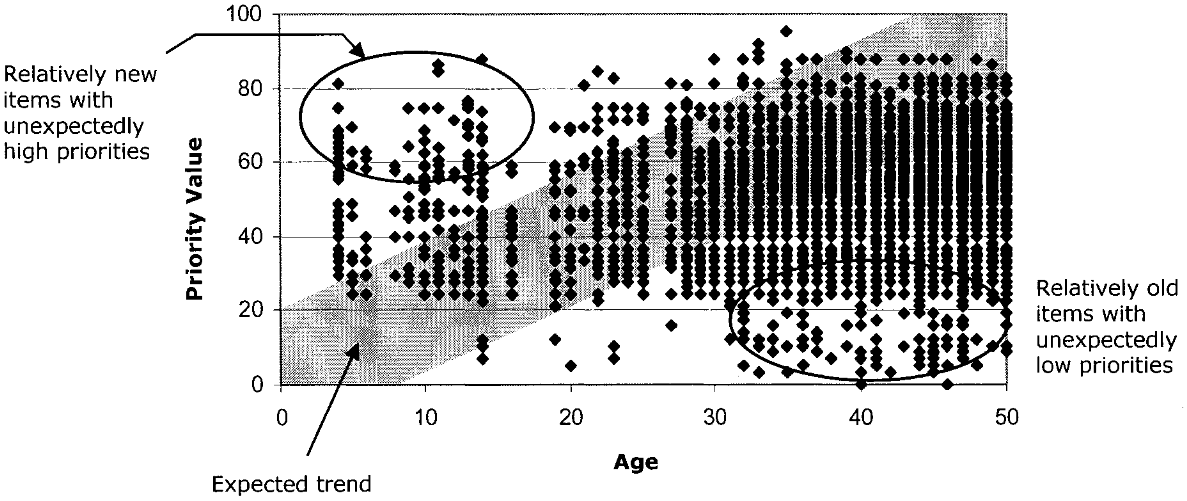


Figure 6.18: Age vs. priority value

2. Suitability of Priority Value for Fund Allocation: In the obtained data, the assigned priority values to various instances of certain components (e.g., windows, roofs, and boilers) are analyzed. First, the frequencies of instances with various priority values are plotted. For the window component, a clustering has been observed, as shown in Figure 6.20, where more than 56% of the window instances have the same priority value (59.3). This indicates a potential problem during the fund allocation to a large number of instances having the same priority.

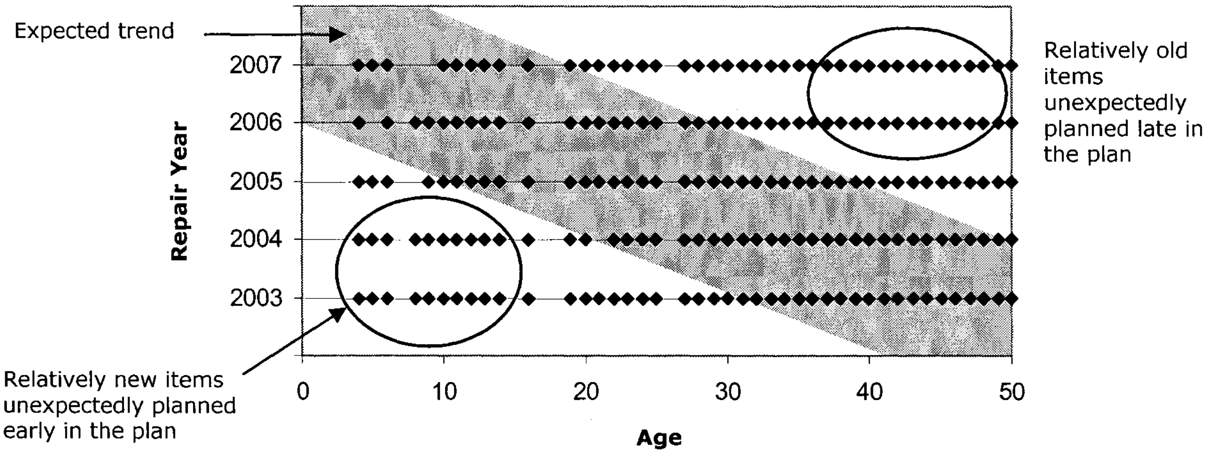


Figure 6.19: Age vs. repair year

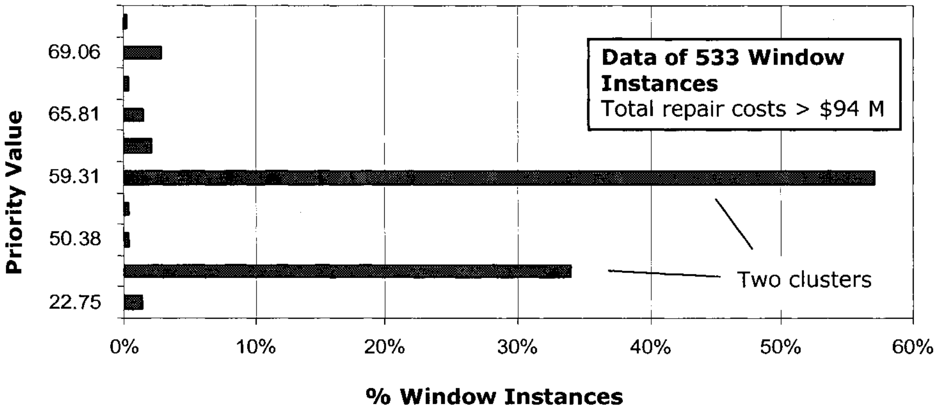


Figure 6.20: Distribution for the priority values for window instances

The same observation can be made with boilers and roofs, as shown in Figures 6.21 and Figure 6.22, respectively.

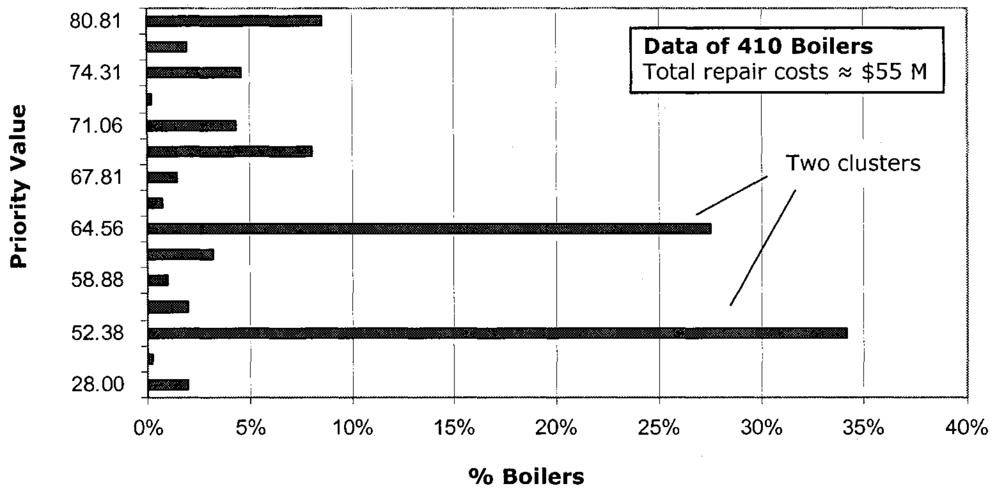


Figure 6.21: Distribution for the priority values for boilers

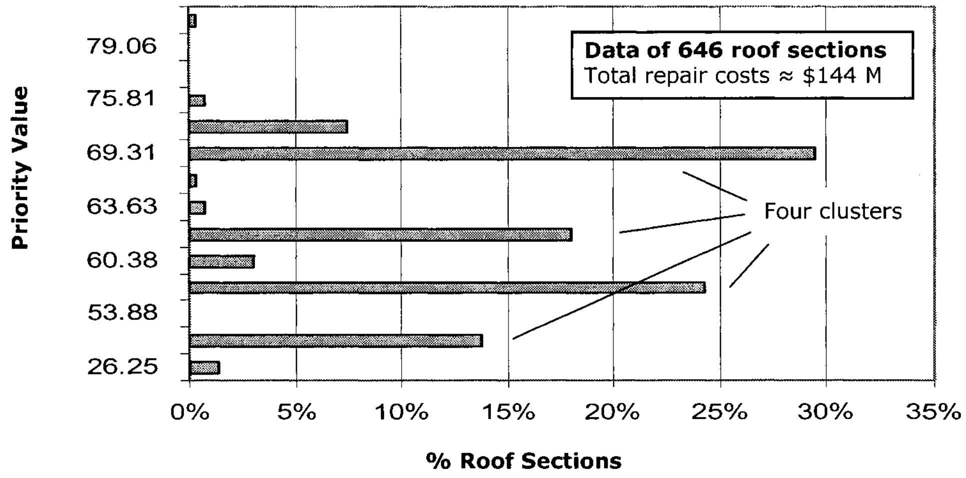


Figure 6.22: Distribution for the priority values for roof sections

In another analysis considering all components, the S-curve for accumulated repair costs is plotted against the priority values, as shown in Figure 6.23.

The enlarged portion of Figure 6.23 shows a potential fund allocation problem since the vertical lines indicate many components having the same priority value and no clear mechanism for their prioritization.

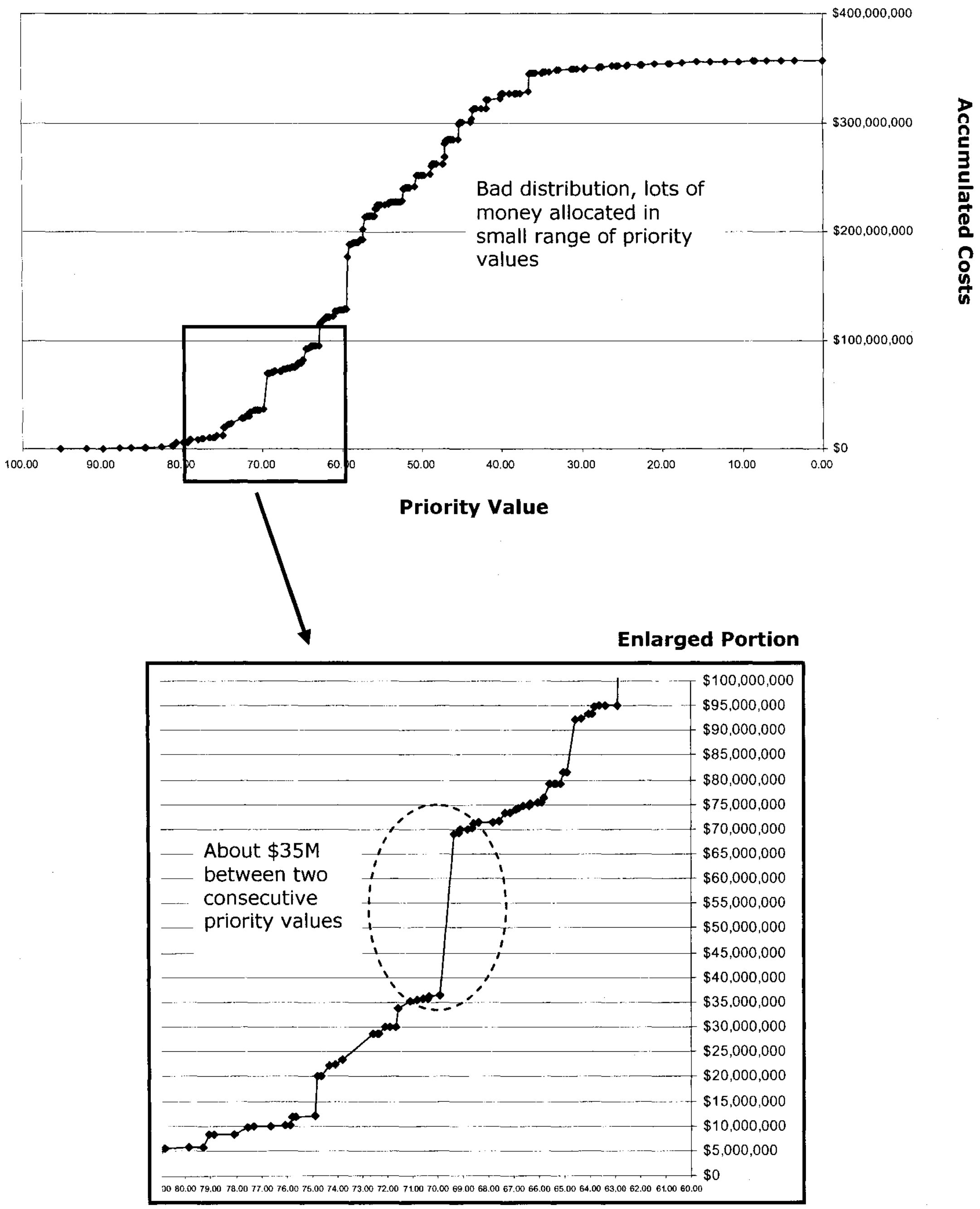


Figure 6.23: Accumulated spending vs. priority values

In doing a similar analysis at the level of a single component in one year, Figure 6.24 shows the accumulated costs for only roofs planned to be repaired in 2003. The figure shows that the priority factor is not a sufficient prioritization factor.

As a conclusion, the obtained data, in its original form (reflective of the lack in decision support of existing tools) can not be readily used for comparison purposes within the proposed developments. As such, some data preparation is needed as discussed in the next subsection.

It is noted that the relationships used to examine the data gave an example of quality control tests on some of the outputs of existing asset management systems.

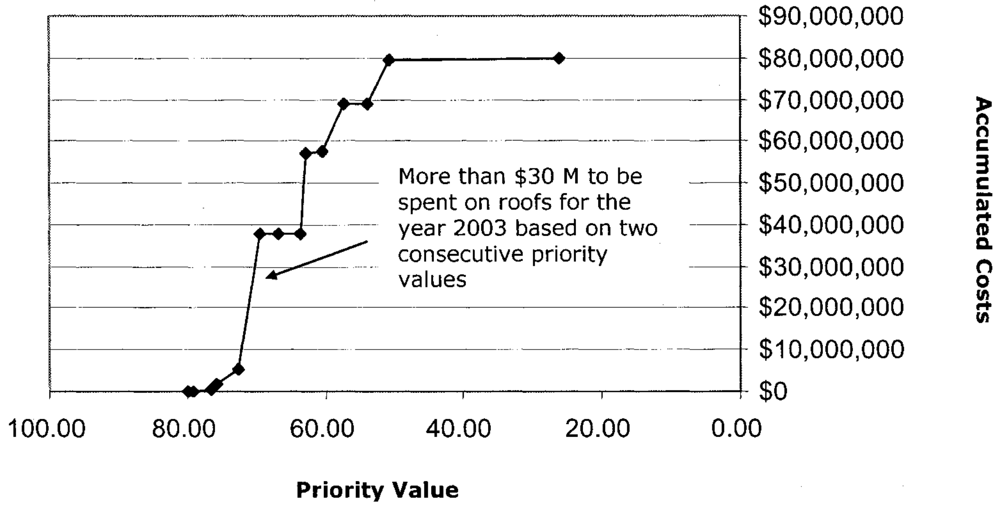


Figure 6.24: Accumulated spending in 2003 for roof sections

6.3.2 Data Preparation

Since the proposed framework essentially depends for its decision support on repairing severities of defects (which are not available in the obtained data), these data have to be generated. Also, since it is very time consuming to generate severity data for all the 14,857 records of building components, only the four major components of a building are considered: (1) roof; (2) boiler; (3) window; and (4) fire alarm system.

With components' deficiencies not given in the data, for simplicity, any instance of the four components is assumed to have a single deficiency, representing its overall condition (DI). The process of generating reasonable DI values for instances followed two steps (Figure 6.25):

1. Establishing a reasonable deterioration curve for each component, showing the expected DIs at various ages. The curve has to show logical trends, as shown at the bottom of Figure 6.25; and

2. Considering each instance, one at a time, and according to its age, determine its DI, randomly within a range of (+/- 10) points from the expected DI at that age (from the curve in step 1).

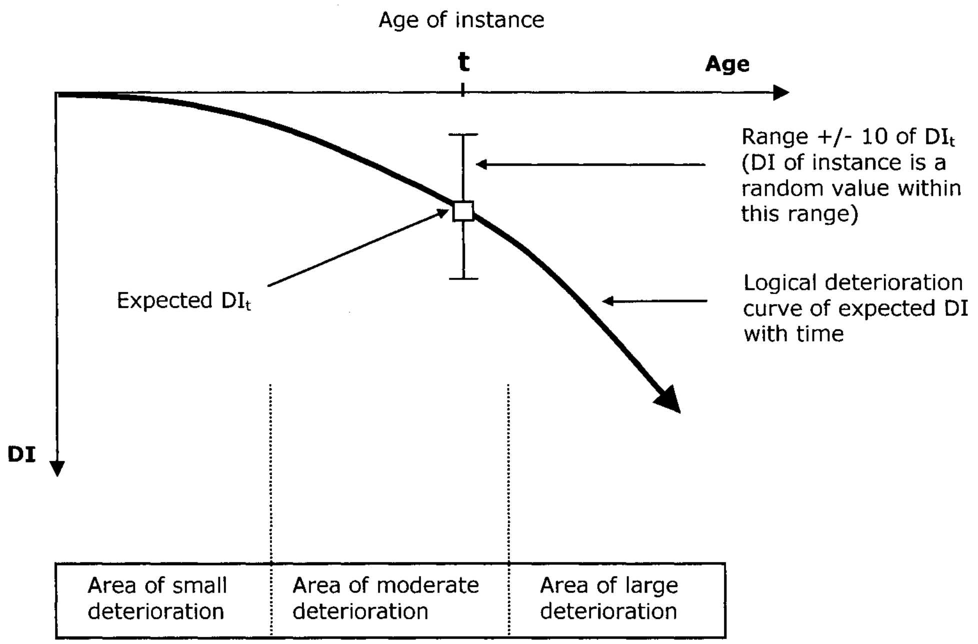

Age of instance
t
Age
Range +/- 10 of DI$_t$ (DI of instance is a random value within this range)
Expected DI$_t$
Logical deterioration curve of expected DI with time
DI
Area of small deterioration
Area of moderate deterioration
Area of large deterioration

Figure 6.25: Process of Generating DI values

These two steps are explained in the following:

1. Component Expected Deterioration Curve: Since the original data contained priority values for each instance (with known age), these data are used to generate the expected deterioration curve. To do that, some statistics are conducting among the data as shown in Table 6.1 for roofs. The table shows the frequency of roof instances that expected to be in good condition, fair condition, and poor condition, respectively. The table also shows the minimum, average, and maximum ages for these groups of instances. The calculations for the reasonable ages corresponding to various condition classifications (i.e., DI of 40, 60, and 80) were performed in the same table using the weighted average of various age categories. These calculations were done based on the expected logic of small deterioration (good condition) which should happen at the early ages (minimum), moderate at the average ages, and severe at the maximum ages.

Table 6.1: Minimum, Average, and Maximum Ages for Roof s

Classification	# of instances	Minimum age	Average age	Maximum age	Age based on weighted average
Good (DI = 40)	1	40			(40*1+28*14)/ 15 = **29**
	14	28			
Fair (DI = 60)	30		36.33		(36.3*30+41*2+41.6*66+ 38.5*96+37.9*11) / 205 = **39**
	2		41		
	66		41.62		
	96		38.51		
	11		37.91		
Poor (DI = 80)	3			46	(46*3+47*1)/4 = **46**
	1			47	

The expected deterioration curve for roofs, as a result of the previous calculations, is shown in Figure 6.26 with values as shown to the left.

3. **Generating DIs for all Instances:** A reasonable DI value for any instance of a component is generated as a random value that is +/- 10 points around the expected DI value for all instances at the same age. The curve in Figure 6.26, as such is used to generate for any roof section a reasonable DI value. For example, according to Figure 6.26, an instance with age of about 30 has an expected DI of about 40, this instance, as such, will be assigned a random DI value between 30 and 50 (i.e., +/- 10 points from DI = 40). Figure 6.27 shows the generated DI values for all roof sections with ages between 0 and 50.

DI	Age
0	0
20	14
40	29
60	39
80	46
100	50

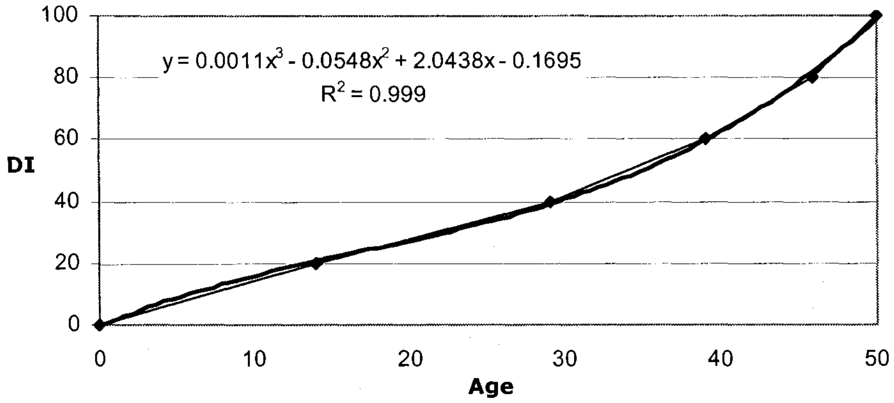


Figure 6.26: Expected deterioration curve for Roofs

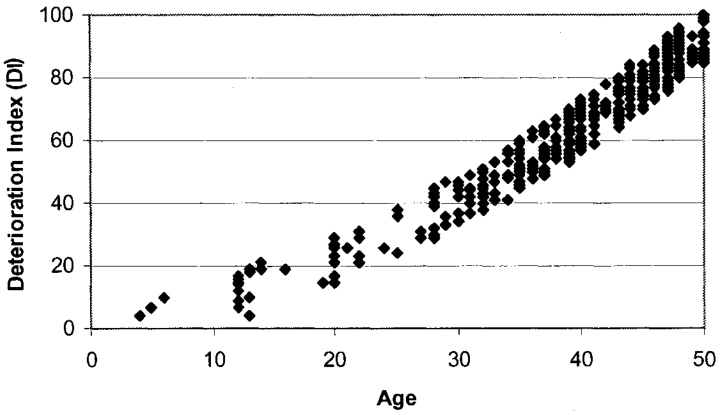


Figure 6.27: Randomly generated DIs for Roofs

The same process is followed to determine reasonable deterioration curves for the remaining three components (window, boiler, and fire alarm system). The generated DIs and the trend of deterioration can be shown in the following figures.

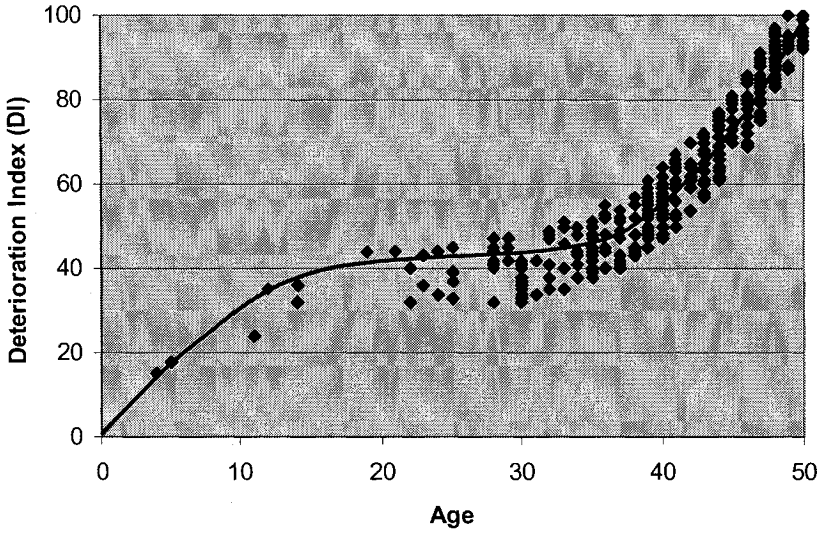


Figure 6.28: Randomly generated DIs for Windows

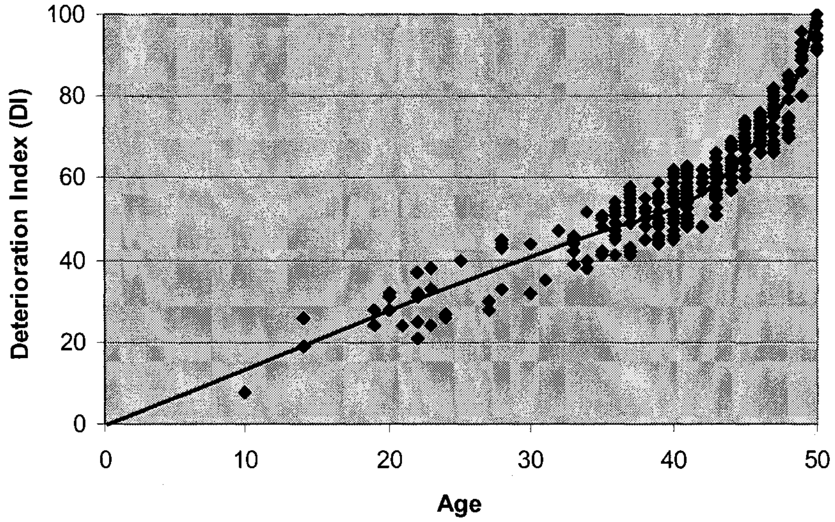


Figure 6.29: Randomly generated DIs for Boilers

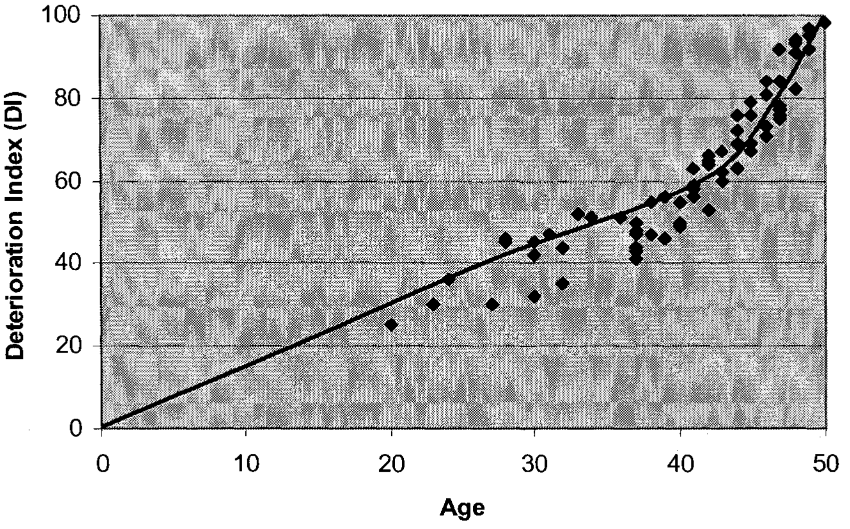


Figure 6.30: Randomly generated DIs for Fire alarm systems

With the determination of reasonable DI values for the 800 instances of roofs, windows, boilers, and fire alarm systems, the data can now be fed into D-Tracker to continue the analysis of the case study, as described in the next subsections.

6.3.3 Feeding the Data into D-Tracker

With the DI's calculated, and the assumption that each instance has one deficiency (weight = 100%), then the severity of that deficiency can be calculated using Equation 3.1, and the value becomes the same as the DI. As such, all data become ready for use in D-Tracker.

Analyzing Replacement Options using SOAP:

In order to analyze the data, the repair model was set to respect only the replacement option (as the cost to replace any instance is known from original data). Also, the cost of replacement for any instance is considered as the one provided with the original data. Accordingly, the SOAP analysis uses its Markov chain model to predict the deterioration of each instance from its current DI value in the future and considers the impact of a replacement strategy each year. With the completion of the SOAP analysis, six decision options become available for each instance of the 800 instances. These options include the no-repair option with zero cost, and five other options of replacement at different years in the planning horizon with associated costs. The condition after replacement becomes a new condition (i.e., DI = 0).

Analyzing Best Replacement Time using A-FAP:

The asset fund allocation program (A-FAP) was then activated to analyze the results of SOAP. The yearly budget to repair the 800 instances was limited to $10,000,000. This value was set based on the relative need approach discussed in section 5.4. The value is calculated after consulting the school board (owning the data) and knowing that the total budget for all instances (i.e., the 14,857 instance) is expected to be $50,000,000/ year during this 5-year planning horizon. The program then created the integer spreadsheet model for asset prioritization. The spreadsheet was then exported to MS-Excel to facilitate the experimentations.

Results of Experimentations:

As mentioned in Chapter 5, many optimization experimentations were carried out to compare among various formulations for the repair-fund allocation problem. Various sizes of networks were formed from the 800 instances by randomly eliminating some instances from the four types of components (roof, window, boiler, and fire alarm system). Therefore, networks with 100, 200, 300, 400, and 800 instances were created and allocated yearly budgets of $1,300,000, $2,500,00, $3,500,00, $4,640,000, and $10,000,000, respectively based on relative need. It is worth to know that the total need for the 800 instances is $155,112,010 and the available limit is $10,000,000/ year (i.e., $50,000,000 total, covering less than 1/3 of the need).

The main challenge as such, is how to maximize the overall network condition (minimum deterioration index) during the five years, given these limited budgets. In order to do that, various optimization tools and strategies were examined to ensure close-to-optimal solution for any size network. With the failure of the integer model using mathematical optimization tools (Excel Premium Solver), the binary model was used to provide the best solutions for the sizes of networks it can handle. Genetic Algorithms (GAs), on the other hand, are used on both the integer and the binary models, with different strategies to improve its accuracy to reach to acceptable optimal solutions. With GAs being suitable for large-scale problems, the best optimization strategy can help provide better solutions to any size of network. The GA tool used in this study is a commercial Excel add-in software, Evolver.

Mainly, two strategies were used for the optimization, as follows: (1) all-years strategy; and (2) step-wise strategy (year by year), as discussed in Chapter 5 (Table 5.1 provides a summary of all results). The results in terms of improvement to the overall condition (i.e., the new condition after optimization) were as follows:

Experiment 1: Smallest size network (100 instances)

Yearly budget = $1,300,000; original network condition (DI_N) = 55

- Mathematical Optimization using Solver (optimal):
  - All-years strategy: improved DI = 29.
  - Step-wise strategy: improved DI = 29.
- Genetic Algorithms using Evolver (near optimal):
  - All-years strategy: improved DI = 44 (integer) and 39 (binary).
  - Step-wise strategy: improved DI = 29 (integer) and 30 (binary).

Comments:

The step-wise strategy provides with either the integer model or the binary model close to optimal results, almost the same results if compared with mathematical optimization. The experimentation with the binary model, however, was noticed to be slower than the integer model, as the number of variables and constraints are larger.

Experiment 2: Largest size network (800 instances)

This experiment is to check the ability to provide reasonably accurate solutions for large networks.

Yearly budget = \$10,000,000; original network condition (DI_N) = 54

- Mathematical Optimization using Solver (optimal):
  - All-years strategy: failed due to the large size (4800 variables).
  - Step-wise strategy: improved DI = 32.
- Genetic Algorithms using Evolver (near optimal):
  - All years strategy: improved DI = 44 (integer model).
  - Step-wise strategy: improved DI = 32 (integer model).

Comments:

The accuracy of GAs didn't change with the size of network, although it is noted that, the time to reach the solution increased, especially to reach to a very close value to Solver results. The presented results are rounded which mean there is a very small difference between Solver and Evolver results. Another important comment is that, Solver failed to handle the problem with the all-years strategy because the number of variables increased from 600 variables in case of 100 instances to 4800 variables in case of 800 instances. To check the limit for the available mathematical optimization tool

(Solver), further experimentations have been carried out using various size networks, as explained next.

Experiment 3: 200 Instances

Yearly budget = $2,500,000; original network condition (DI_N) = 54

- Solver:
  - All-years strategy: improved DI = 31; variables (1200).
  - Step-wise strategy: improved DI = 31

Experiment 4: 300 Instances

Yearly budget = $3,500,000; original network condition (DI_N) = 54

- Solver:
  - All-years strategy: improved DI = 32; variables (1800).
  - Step-wise strategy: improved DI = 31

Experiment 5: 400 Instances

Yearly budget = $4,640,000; original network condition (DI_N) = 53

- Solver:
  - All-years strategy: failed to handle; variables (2400).
  - Step-wise strategy: improved DI = 31

Comments:

Solver couldn't handle 2400 variables, which means the step-wise strategy will soon fail to handle large networks. Accordingly, it is expected that, the maximum number of variables that can be handled by mathematical optimization is less than 2400. Using this limit on the step-wise model, the maximum network size that can be handled is (2400/2 = 1,200 instances).

As illustrated by the results of the various experiments, the GAs tool is more reliable in optimizing the prioritization problem for large-scale networks of building components, using the integer model with a step-wise strategy.

Experiment 6: Comparison between Optimization and Simple Ranking for Fund Allocation

One important comparison was performed to compare the results of optimization versus the simpler approach of ranking (scoring), used in existing software systems. In all the previous optimization experiments, the selection of instances for repair (and accordingly the budget allocation) use a single criteria to prioritize the instances, which is the expected performance (EP) due to instance repair, weighted by the importance factor of the instance. Using the ranking approach, however, two possible ways can be followed:

1. Use the year of repair manually suggested by inspectors to allocate funds for instances, and within each year prioritize the instances by any criteria, such as the one used in the optimization experiments. This approach of constraining the year, however, is expected to perform poorly as it is mainly subjective and lacks any optimization. This approach is not comparable to the approach used in the optimization and as such, is not suitable for comparison; and

2. Ignore the repair year suggested by inspectors and rank the instances based on their weighted EPs (same as the optimization experiments). Accordingly, the instances to repair each year are the ones with highest priorities, until the year's budget is exhausted. This scenario is considered for the network of 800 instances and the result shows improvement from an initial network condition of 54 to a network condition of only 45.

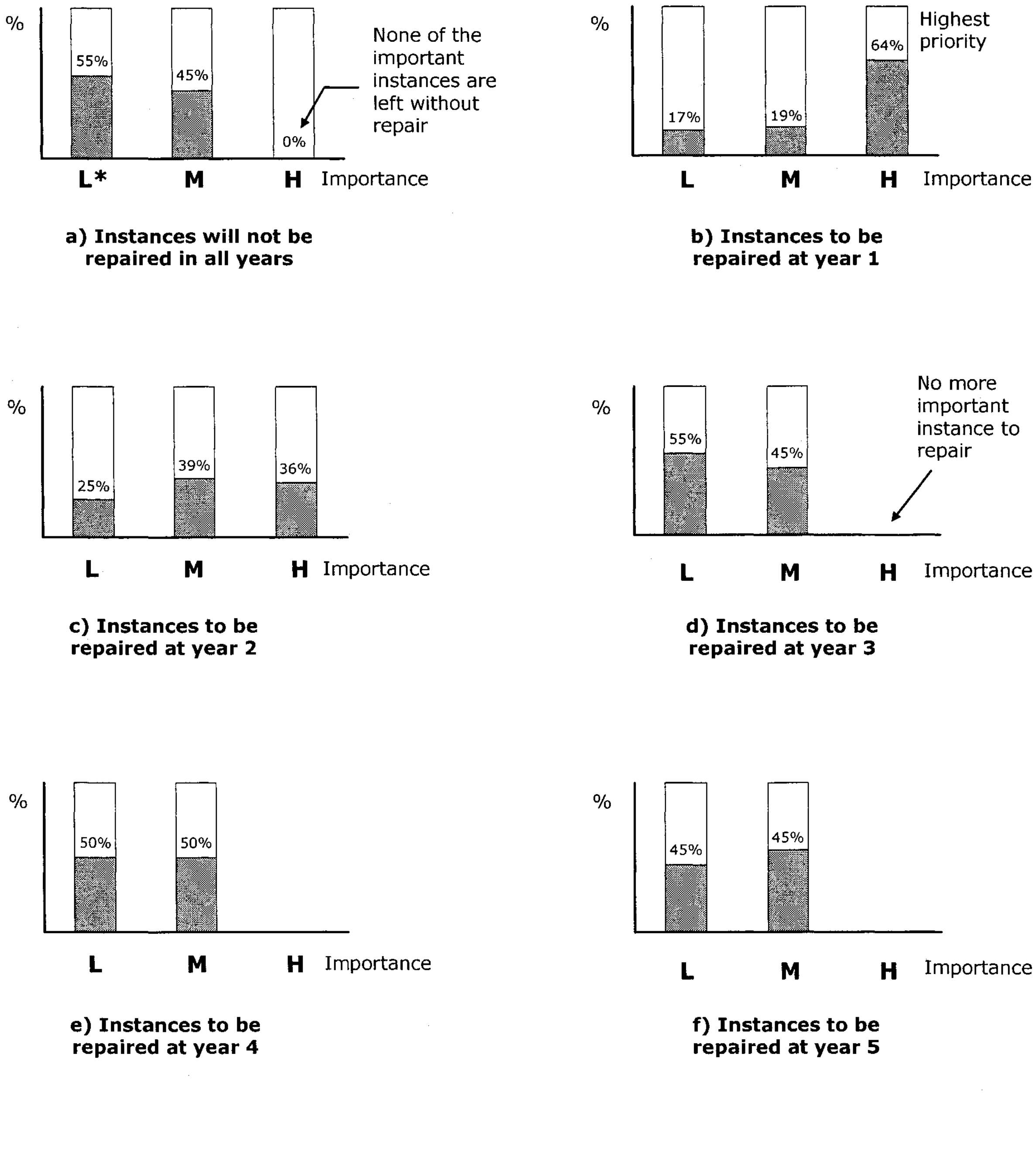
Such poor performance for ranking as compared to optimization results is due to the fact that ranking can not consider the network as a whole and focuses only on one instance at a time.

6.3.4 Comments on Optimization Results

As the best solution for the prioritization problem is determined using optimization, the resulted 5-year plan is considered and tested against a set of logical trends.

Figure 6.31 shows an analysis of the decisions (priorities) made in each year. The figure shows the percentages of instances with high, medium, and low importance. As shown in Figure 6.31 (a) for example, none of the instances with high importance are omitted. Same trend is also observed in Figure 6.31 (b) where the highest percentage has high importance. Once, all the high importance instances are completed in the repair of year 1 and year 2, all other years consider the lower importance instances. It can be concluded that the optimization provides reasonable selections for the instances to be repaired in various years of the plan.

In another representation, Figure 6.32 shows an analysis of the instances at the same importance factors. For example, the pie graph for RIF = 70 (relatively low importance) shows that 23%, 10%, 13%, 9%, and 6% of the instances will be repaired in years 1, 2, 3, 4, and 5, respectively, while the remaining 32% will not be repaired due to budget shortage. The pie for RIF = 90 (relatively high importance), also shows that all the instances with this high importance will be repaired in the first two years only.



a) Instances will not be repaired in all years

b) Instances to be repaired at year 1

c) Instances to be repaired at year 2

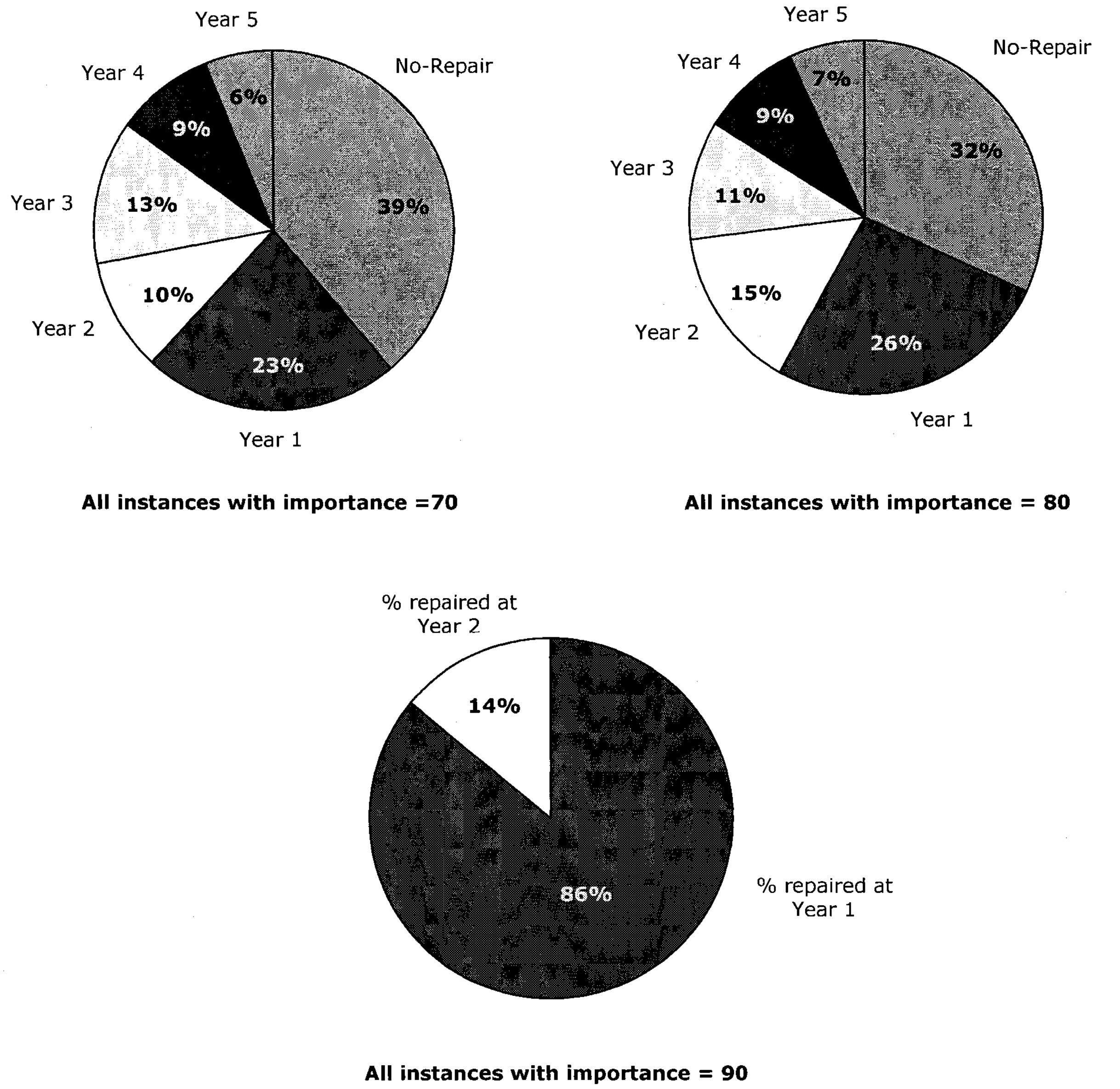
d) Instances to be repaired at year 3

e) Instances to be repaired at year 4

f) Instances to be repaired at year 5

* L = Low relative importance =70; M = Medium importance = 80; and H = High importance = 90

Figure 6.31: Analysis of year-by-year decisions



All instances with importance =70

All instances with importance = 80

All instances with importance = 90

Figure 6.32: Analysis of instances at various importance values

In terms of prioritization capacity, Figure 6.33 shows the relation between the instance age and the prioritization criteria (performance criteria RIF x $EP^{(0)}$) used in the optimization. As shown in the figure, the higher the age of the instance, the higher its prioritization criteria (i.e., eligibility for repair), which is a logical trend.

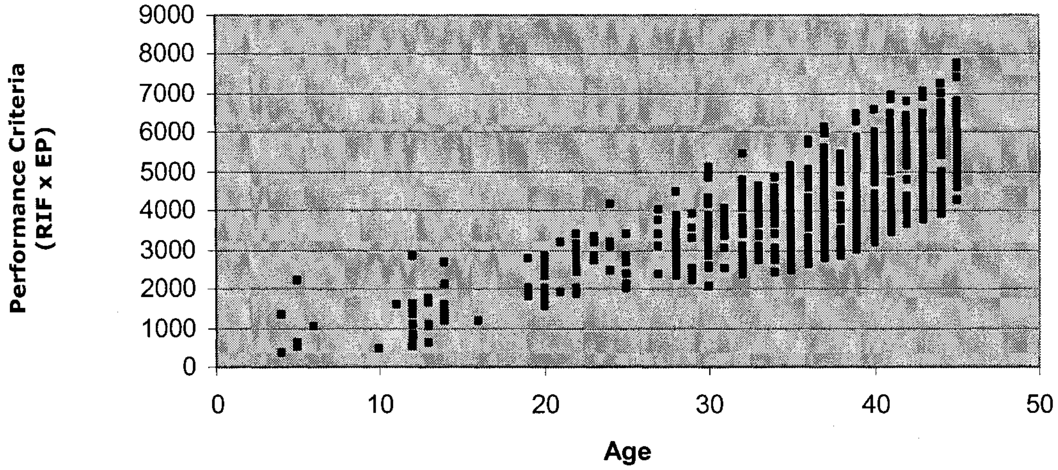


Figure 6.33: Age versus prioritization criteria

In addition, as noticed in Figure 6.33, the prioritization criteria used has a large range (0 to about 8000), which ensures a wide enough range to distinguish among instances for funds allocation.

In terms of yearly network improvements, Figure 6.34 (a) shows the number of instances repaired at various years, against the prioritization criteria before repair (representing condition and importance). Figure 6.34 (b), on the other hand, shows for each year, the instance's prioritization criteria (performance) after repair. As shown, the result of optimization shows that yearly in the plan, the instances with high importance are considered. Also, the result of the repair in early years is a large impact on the network (the range before repair in year 1 is compressed much after repair).

6.4 Conclusions

This chapter provided a brief overview of the capabilities of the proposed framework as well as the results of various optimization experiments. Based on the results, the framework is shown to provide logical decision support. Mathematical optimization was shown to perform adequately up to 1,200 instances, while the GAs can work on much larger networks to provide near optimum results. Analysis of the fund allocation results of optimization show logical trends of repairing the instances with high importance early and a manner that achieve maximum network benefit.

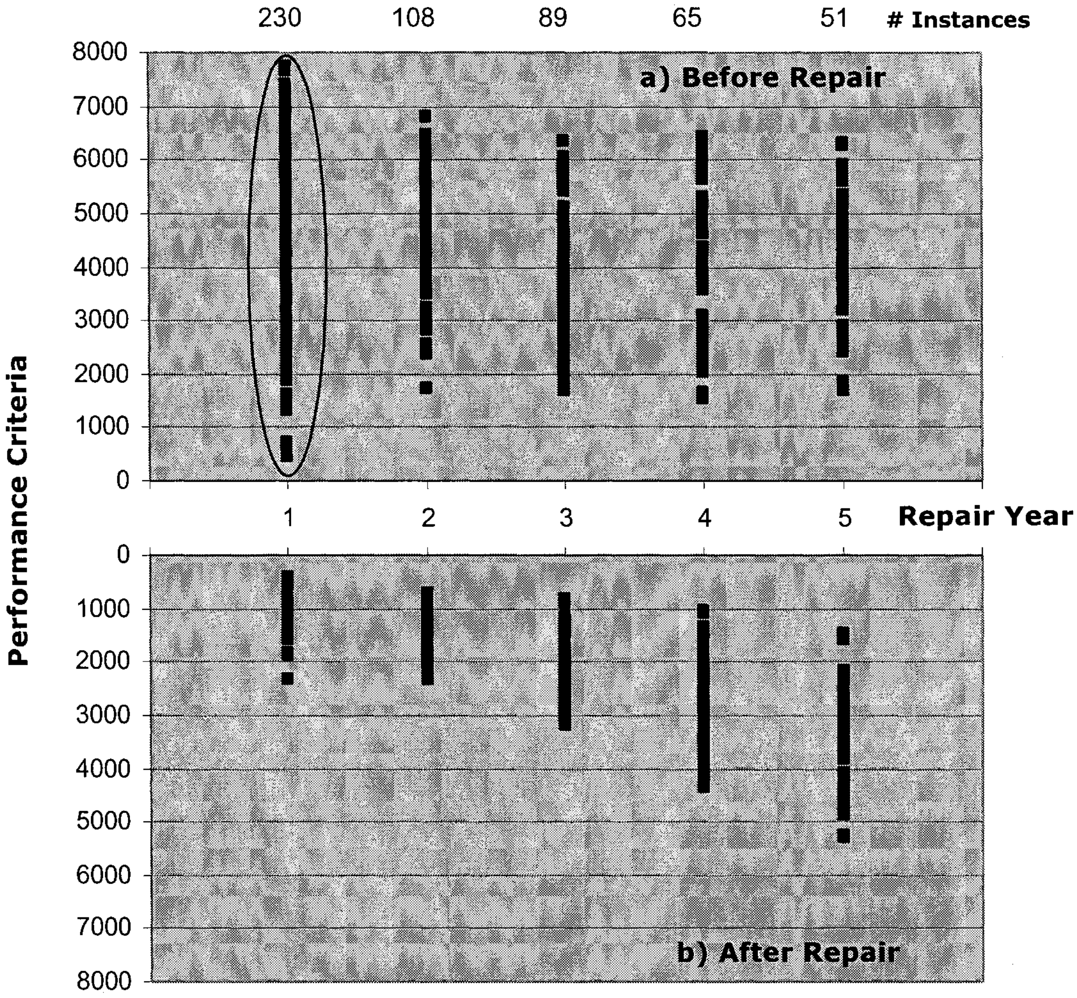


Figure 6.34: Optimization impact on the overall network improvement

Chapter 7

CONCLUSIONS AND FUTURE RESEARCH

7.1 Conclusions

With the infrastructure getting old and being in an increasing demand, government and large owner organizations are under constant pressure to sustain safety and operability of the infrastructure with minimum cost. Condition assessment, deterioration modeling, repair selection, asset prioritization, and fund allocation, therefore, are crucial processes to the management of large network of infrastructure assets. The main objective of this thesis is to develop an integrated asset management framework that can overcome the drawbacks of traditional practices in asset management for the building infrastructure domain.

Building networks are complex in nature due to the large number of diverse interrelated components and systems involved. As such, fundamental changes should take place in many areas related to condition assessment, deterioration modeling, and repair selection. Traditional approaches exhibit high level of subjectivity in the condition assessment, static and less-accurate deterioration models, and inadequate processes for repair selection and asset prioritization. Current approaches and their underlying assumptions (that might have suited other infrastructure domains), as such, are not valid for building assets. Therefore, this research introduced improved approaches that track the dynamics of components' defects. As such, the research provides adequate decision support for the identification, tracking, and repairing the specific defects in building components. Models were, therefore, developed to perform condition assessment, custom deterioration prediction, best repair selection, and network-level prioritization. These individual models were integrated in a comprehensive asset management framework.

The developed framework is expected to help re-engineering the traditional asset management processes of the building infrastructure and the decision-making process related to budgeting and repair strategies. Providing decision support at the detailed defects level, however, is complex and has been innovatively handled in this research through three successive optimizations related to: (1) custom deterioration modeling; (2) best repair selection; and (3) network-level prioritization. This approach is practical and can handle large-scale networks since it reduces the complexity of formulating the whole process in one optimization model. The first optimization is responsible for determining accurate and custom predictions for future deficiencies for each instance in the network. As these future defects represent potential problems that might need repair, the second optimization determines, for each year in the planning horizon, the best repair strategy to improve the condition of the component to a user-defined acceptable level, with minimum cost. These two optimizations, as such, provide the best repair option in each year in the planning horizon. The third optimization, therefore, uses these options at the network level to determine the best repair year for each component, considering budget limits and other user constraints.

The proposed framework was implemented in a prototype program called D-Tracker which was validated using a real-life case study. Using the capabilities of D-Tracker, various optimization experiments were experimented with, which showed the ability to provide managers with logical decision support. The experimentation results illustrated that mathematical optimization can optimize decisions for up to 1,200 components, while the Genetic Algorithms (GAs) technique can handle much larger networks to provide near-optimum results.

Based on further experiments, it was obvious that optimization provided excellent solutions compared to simple ranking. The adopted optimization was capable of improving the overall network condition (deterioration index) from an initial value of 54 to an improved value of 32, while the simple ranking, for the same data, improved the condition to only 45.

This research has made a number of contributions within its individual modules and also in the integrated asset management framework itself. The details are as follows:

1. **Condition Assessment:** A new system (V-CAP) is presented to support not only the visual inspection process, but also the revision process to ensure the quality of collected data. The inspection tool uses an interactive visual database system to help reduce the subjectivity and improve the accuracy of assessing various components' defects. This also results in a process that is speedy and inexpensive, in addition to being suitable for less experienced individuals. Consequently, it becomes possible to electronically distribute the inspection system locally at each facility on hand-held devices to conduct the condition assessment simultaneously at all facilities, to save time and cost.

   The revision services provided in V-CAP also enables managers in the office to verify / change the inspection data, and to create useful statistics using a variety of filtering criteria. By filtering the data of one component (e.g., roofs), the statistics show an anatomy of condition in various roofs, inspection details, and inspectors' rates of field assessment work;

2. **Deterioration Modeling:** The proposed model for deterioration prediction is novel in its formulation and its integration with the condition assessment. The link from condition assessment to prediction modeling provides reliable data to make accurate deterioration predictions. The assessment data embodies the maintenance history of the specific organization. Using this data, the proposed deterioration model dynamically produces a custom deterioration curve for each instance in the network using optimization. As such, the model considers the implicit impact of the specific working environment of each instance and any other unforeseen parameters that

affect the instance's rate of deterioration, including the interaction with other components. This customization is achieved through the optimized Markov chain model, which generates a custom transition probability matrix for each instance.

The custom deterioration curve is not only useful for predicting the overall condition of an instance in any year in the planning horizon, but also translating this future condition into predicted severities of defects, which define specific repair needs. The deterioration model, therefore, incorporates systematic steps to determine these severities for each year in the planning horizon;

3. **Repair Modeling:** The proposed methodology of examining all possible strategies to repair the expected severities of defects is more accurate and practical than the traditional way of assigning subjective and predefined repair options (e.g., minor repair, major repair, etc.). This is because the model is more focused on the defects, repairs, costs, and the impact on condition improvement. The model, as such, is novel in its use of optimization to determine the least cost strategy to repair the expected severities of defects in a given year to reach a minimum acceptable level.

   The model also incorporates a procedure to calculate the condition improvement after repair without the need for repeating the condition assessment. The focus on the detailed defects and the specific repair actions is also useful in setting up repair contracts with minimum efforts; and

4. **Prioritization and Fund Allocation:** The presented model for network-level optimization is capable of determining the best overall repair decisions and allocating appropriate funds for realistic 5-year plans. The formulation in this stage is simplified and reduced into a reasonable size of linear formulation due to the fact that the repair model provided a defined list of least-cost repair options along the planning horizon. As such, the prioritization model determines only the best year of repair for each instance in the network. One additional contribution in this aspect is the development of a sub-network optimization approach to allocate funds efficiently among a group of assets that are of particular interest to the decision maker.

   In terms of fund allocation, the research highlighted a simple approach to decide on the portion of funds to be allocated to any sub set of network based on repair needs. The same simple approach is general enough for use at the system level, the organization level, or the ministry level (e.g., allocating funds to various infrastructure sectors).

In addition to the above contributions, this thesis provided insights to examine the quality of outputs of any asset management system through a set of tests on the outputs of the system to examine if logical relationships exist in the decision. By applying these tests on the outputs of the proposed framework, the optimization results proved to follow logical trends of repairing the instances with high importance early in the plan and in a manner that achieves maximum network benefit.

Finally, this research has successfully structured a comprehensive asset management system that can accurately allocate funds and prioritize assets for repair purposes. Although the focus has been on asset management for educational buildings, the proposed system can easily be adapted to other types of building assets such as hospitals, and to other asset domains such as bridges, highways, and airports. The proposed research is expected to aid consultants and owner organizations, such as municipalities and government agencies, to make appropriate decisions that ensure the sustainable operation of the infrastructure assets with the least cost and optimum operational condition.

4.1. Future Research

There are several potential improvements to the developed asset management system presented in this study and other areas of future research related to the developed system. These include:

- Adding a fourth module to the developed system as a feed back and control module to record / update actual costs and repair actions. Such module will help in constructing a constructability filter for future repair selection. Also, it will help in doing better estimates for repair costs;

- Testing the condition assessment module individually in an actual assessment survey to determine its performance. Also, expanding the visual guidance system with additional pictures of various deteriorating components;

- Examining the applicability of the system on other types of assets and determine improvements for generalization;

- Doing a more detailed study on repair cost calculations for individual deficiencies;

- Linking the developed framework to other systems for execution planning such as BAL (Hegazy 2004); and

- Using GIS and visualization techniques to present the system inputs and outputs such as condition indices, level of funding, backlog, and actual versus planned performances.

REFERENCES

1. Abdelrahim, A. M., and George, K. P. (2000). "Artificial neural network for enhancing selection of pavement maintenance strategy." Transportation Research Record: *Journal of the Transportation Research Board*, No.1669, TRB, Washington, D.C., pp. 16-22.

2. Abraham D. M., Iseley, T., Prasanth, R. K., and Wirahadikusumah, R. (1997). "Integrating Sensing Technologies for Underground Utility Assessment." *Proceedings of Infrastructure condition assessment: art, science, and practice*, ASCE, Boston, Massachusetts, August 25-27, pp.316-325.

3. Al-Battaineh, H. T., AbouRizk, S. M., Siu, K. L., and Allouch, M., (2005). "The Optimization of Infrastructure Budget Allocation Using Genetic Algorithms." *Proceedings of 1st CSCE Specialty Conference on Infrastructure Technologies, Management, and Policies*, CSCE, Toronto, Ontario, Canada, June 2-4, FR-173.

4. ASCE (2005 a). "http://www.asce.org/reportcard/2005/index.cfm" Accessed October 31, 2005.

5. ASCE (2005 b). "http://www.asce.org/reportcard/index.cfm?reaction=full&page=6" Accessed October 31, 2005.

6. Bandara, N., and Gunaratne, M. (2001). "Current and future pavement maintenance prioritization based on rapid visual condition evaluation." *Journal of Transportation Engineering*, 127(2), 116-123.

7. BUILDER (2002). "BUILDER User Guide." U.S. Army, *Engineering Research and Development Centre - Construction Engineering Research Laboratory (ERDC-CERL)*, Champaign, IL, USA, May 2002.

8. Butt, A.A., Shahin, M.Y., Feighan, K.J., and Carpenter, S.H. (1987). "Pavement Performance Prediction Model Using the Markov Process." Transportation Research Record: *Journal of the Transportation Research Board*, No. *1123*, TRB, Washington, D.C., pp. 12-19.

9. Choate, P. and Walter, S. (1981). "America in Ruins: The Decaying Infrastructure." Duke Press, Durham, N.C.

10. Collins, L. (1972). "An introduction to Markov chain analysis." CATMOG, Geo Abstracts Ltd., Univ. of East Anglia, Norwich, U.K.

11. Coullahan, R. and Siegfried, C. (1996). "Facilities Maintenance Using Life Cycle Asset Management." *Facilities Engineering Journal*, Mar/Apr, Association for Facilities Engineering, Cincinnati, Oh (http://www.afe.org/).

12. DeStepano, P.D., and Grivas, D.A. (1998). "Method for Estimating Transition Probability in Bridge Deterioration Models." *Journal of Infrastructure Systems*, 4 (2), pp.56-62.

13. Dogaki, M., Furuta, H., Tsukiyama, I., and Fraqngpool, D. (2000). "Optimal Maintenance Planning of Reinforced Concrete Decks on Highway Network." *Conference proceeding, US-Japan Workshop on Life-Cycle Cost Analysis and Design of Civil Infrastructure Systems*, ASCE, Honolulu, Hawaii.

14. Earl, R. (1997). "The Condition Assessment Survey: A Case Study for Application to Public Sector Organizations." *Proceedings of Infrastructure condition assessment: art, science, and practice*, ASCE, Boston, Massachusetts, August 25-27, pp 277-286.

15. Elhakeem, A. and Hegazy, T. (2005). "Improving Deterioration Modeling using Optimized Transition Probability Matrices for Markov Chains." *Proceedings of the 84th Transportation Research Board Annual Meeting*, TRB, Washington, D.C., USA, Jan. 9-13, No.:05 - 2656.

16. Elhakeem, A. and Hegazy, T. (2005). "Towards a Visual Guidance System for Condition Assessment of the Building Infrastructure." *Proceedings of 1st CSCE Specialty Conference on Infrastructure Technologies, Management, and Policies*, CSCE, Toronto, Ontario, Canada, June 2-4, FR-159.

17. Engineering Research and Development Centre - Construction Engineering Research Laboratory (ERDC-CERL) in Champaign, IL, USA.

18. Evolver (1998). "The Genetic Algorithms Super Solver: Users Manual." Palisade Corporation, 31 Decker Road, Newfield, NY USA 14867.

19. Federal Highway Administration FHWA (1999). "Asset Management Primer." A Report prepared by the U.S. Department of Transportation.

20. Federation of Canadian Municipalities (1999). "Quality of life infrastructure program." FCM, Ottawa, Ontario, Canada.

21. Federation of Canadian Municipalities (May 2001). "Early warning: will Canadian cities compete?" FCM, Ottawa, Ontario, Canada.

22. Flintsch, G. W. and Chen, C. (2004). "Soft Computing Applications in Infrastructure Management." *Journal of Infrastructure Systems*, 10(4), 157–166.

23. Fukuhara, T., Terada, K., Nagao, M., Kasahara, A., and Ichihashi, S. (1990). "Automatic Pavement-Distress-Survey System." *Journal of Transportation Engineering*, ASCE, May/June, 1990.

24. Fundakowski, R. A., et al. (1991). "Video Image Processing for Evaluating Pavement Surface Distress." Final Report, National Cooperative Highway Research Program 1-27, Washington, D.C., September, 1991.

25. Greimann, L., Stecker, J., Rens, K., McKay, D., and Foltz, S. (1997). "Condition Assessment of Lock and Dam Structures." *Proceedings of Infrastructure condition assessment: art, science, and practice*, ASCE, Boston, Massachusetts, August 25-27, pp.385-394.

26. Grivas, D. A., Schultz, B. C., and Mason, G. L. (1997). "A Framework for Using Satellite and Airborne Remote Sensing Technology in Infrastructure Performance Assessment." *Proceedings of Infrastructure condition assessment: art, science, and practice*, ASCE, Boston, Massachusetts, August 25-27, pp.81-90.

27. Hass, R., Hudson, W.R., and Zaniewski, J.P. (1994). "Modern Pavement Management." Krieger Publishing Company, Malabar, Fla.

28. Hatry, H.P. and Steinthal, B.G. (1984). "Guides to Managing Urban Capital Series." vol. 4: Guide to Selecting Maintenance Strategies for Capital Facilities, The Urban Institute, Washington, D.C.

29. Hedfi, A., and Stephanos, P. (2001). "Pavement performance modeling: An applied approach at the state of Maryland." *5th International Conference on Managing Pavements* (CD-ROM).

30. Hegazy, T. (2002). "Computer-Based Construction Project Management." Prentice Hall, Upper Saddle River, NJ, USA.

31. Hegazy, T. (2004). "Execution Planning: A New Innovative Dimension of Infrastructure Management Systems." *Proceedings of the 83rd Transportation Research Board Annual Meeting*, TRB, Jan. 11-15, Washington, D.C., USA.

32. Hegazy, T., Elbeltagi, E., and Elbehairy, H. (2004). "Bridge Deck Management System with Integrated Life Cycle Cost Optimization." Transportation Research Record: *Journal of the Transportation Research Board*, No. 1866, TRB, National Research Council, Washington, D.C., pp. 44-50.

33. Heiler, M., Motazed, B., and McNeil, S. (1993). "Ground Penetration Radar: Images of Subsurface Characteristics." *Proceedings of the Conference on Digital Image Processing: Techniques and Applications in Civil Engineering*, Kona, Hawaii, February 28- March 5, 1993. ASCE, pp. 173-178.

34. Holland, J. (1975). "Adaptation in Natural and Artificial Systems." The University of Michigan Press, Ann Arbor, MI.

35. Huang, Y., and Moore, R. K. (1997). "Roughness level probability prediction using artificial neural networks." Transportation Research Record: *Journal of the Transportation Research Board*, No.1592, TRB, Washington, D.C., pp. 89-97.

36. Hudson, W.R., Hass, R., and Uddin, W. (1997). "Infrastructure Management." McGraw-Hill, New York, USA.

37. I civil Engineer (2005 a). "http://www.icivilengineer.com/Failure_Watch/Portugal_ Bridge/" Accessed October 31, 2005.

38. I Civil Engineer (2005 b). "http://www.icivilengineer.com/Failure_Watch/2000_ Cleveland/" Accessed October 31, 2005.

39. I Civil Engineer (2005 c). "http://www.icivilengineer.com/Failure_Watch/Russia_ Apartment/" Accessed October 31, 2005.

40. Infrastructure (1992). "A Century of Building America", *Infrastructure*, July 1992.

41. IRC (1994). "Protocols for Building Condition Assessment, Institute for Research in Construction." National Research Council Canada, Ottawa, Canada, pp110.

42. Itoh, Y., Hammad, A., Liu, C., Shintoku, Y., (1997). "Network-Level Bridge Life Cycle Management System." *Journal of Infrastructure Systems*, ASCE, 3 (1) pp. 31-39.

43. Jiang, Y., Saito, M., and Sinha, K. C. (1988). "Bridge Performance Prediction Model Using the Markov Chain." Transportation Research Record: *Journal of the Transportation Research Board*, No.1180, TRB, Washington, D.C., pp. 25-32.

44. Kaiser, H. H. (1996). "A Foundation to Uphold: A Study of Facilities Conditions at U.S. Colleges and Universities." Association of Physical Plant Administrators, Alexandria, Virginia, pp195.

45. Langevine, R., Allouch, M., AbouRizk, S., and Nicoll, J. (2005). "A Building maintenance Decision Support System." *Proceedings of 1st CSCE Specialty Conference on Infrastructure Technologies, Management, and Policies*, CSCE, Toronto, Ontario, Canada, June 2-4, FR-176.

46. Lasdon, L.S., Waren, A.D., Jain, A., and Ratner, M. (1978). "Design and Testing of a Generalized Reduced Gradient Code for Nonlinear Programming." *ACM Transactions on Mathematical Software*, ACM, 4 (1), pp. 34-50.

47. Lee, H. and Chou, E. (1993). "Survey of Image Processing Applications in Civil Engineering." *Proceedings of the Conference on Digital Image Processing: Techniques and Applications in Civil Engineering*, Kona, Hawaii, February 28- March 5, 1993. ASCE, pp. 203-210.

48. Lee, J. H. and Aktan, H. M. (1997). "A Study of Building Deterioration." *Proceedings of Infrastructure condition assessment: art, science, and practice*, ASCE, Boston, Massachusetts, August 25-27, pp.1-10.

49. Liu, C., Hammad, A., Itoh,Y. (1997). "Multi-Objective Optimization of Bridge Deck Rehabilitation using Genetic Algorithm." Microcomputers in Civil Engineering, *Journal of Computer-Aided Civil and Infrastructure Engineering*, 12 (6), pp. 431-443.

50. Lou, Z., Gunaratne, M., Lu, J. J., and Dietrich, B. (2001). "Application of neural network model to forecast short-term pavement crack condition: Florida case study." *Journal of Infrastructure Systems*, 7(4), 166–171.

51. Lounis, Z., Vanier, D. J., and Lacasse, M. A. (1998). "A Discrete Stochastic Model for Performance Prediction of Roofing Systems." *Proceedings CIB World Building Congress*, Gävle, Sweden, June 7-12, (1)1, pp. 305-313.

52. Luenberger, D.G. (1984). "Linear and Nonlinear Programming." Addison-Wesley Publishing Company, MA.

53. Madanat, S. (1993). "Optimal Infrastructure Management Decisions under Uncertainty." Transp. Res. – C, 1(1), 77-88.

54. Madanat, S., Karlaftis, M. G., and McCarthy, P. S. (1997). "Probabilistic Infrastructure Deterioration Models with Panel Data." *Journal of Infrastructure Systems*, 1(3), pp. 4-9.

55. Madanat, S., Mishalani, R., and Ibrahim, W. H. W. (1995) "Estimation of Infrastructure Transition Probabilities from Condition Rating Data." *Journal of Infrastructure Systems*, 1(2), pp. 120–125.

56. Maiyamoto, A. (2001). "Development of a Bridge Management System (J-BMS) in Japan." *Conference proceeding, US-Japan Workshop on Life-Cycle Cost Analysis and Design of Civil Infrastructure Systems*, ASCE, Honolulu, Hawaii.

57. Maser, K. (1995). "Evaluation of Bridge Decks and Pavements at Highway Speed Using Ground Penetration Radar." *Proceedings of the SPIE Conference on the Non-destructive Evaluation of Aging Infrastructure*, Oakland, CA, June 6-7 1995. SPIE Volume 2456, pp. 237-248.

58. Maser, K. R. and Zarghamee, M.S. (1997). "Leakage Evaluation of a Buried Aqueduct." *Proceedings of Infrastructure condition assessment: art, science, and practice*, ASCE, Boston, Massachusetts, August 25-27, pp.218-226.

59. Maser, K. R., Egri, R., Lichtenstein, A., and Chase Steven (1997). "Wireless Global Bridge Evaluation & Monitoring System (WGBEMA)." *Proceedings of Infrastructure condition assessment: art, science, and practice*, ASCE, Boston, Massachusetts, August 25-27, pp.91-100.

60. Mason, G. L., Embrechts, M. J., and Grivas, D. A. (1993). "Roadway Identification on Landsat Imagery Using Artificial Neural Networks." *Proceedings of the Artificial Neural Networks in Engineering (ANNIE'93) Conference*, Volume 3, St. Louis, Mo, November 14-17, 1993. ASME, pp.423-428.

61. Mathworld (2005). "http://mathworld.wolfram.com/BranchandBoundAlgorithm.html/" Accessed December 23rd, 2005.

62. Melvin, E. (1992). "Plan, Predict, Prevent: How to Reinvest in Public Buildings." Special Report #62, American Public Works Association Research Foundation, Chicago, pp83.

63. Moore, J.H. and Weatherford, L.R. (2001). "Decision Modeling with Microsoft® Excel." Sixth Edition, Prentice Hall, Upper Saddle River, NJ, USA.

64. Morcous, G., Rivard, H., and Hanna, A. M. (2002 a). "Case-Based Reasoning System for Modeling Infrastructure Deterioration." *Journal of Computing in Civil Engineering*, ASCE, 16(2), pp. 104–114.

65. Morcous, G., Rivard, H., and Hanna, A. M. (2002 b). "Modeling Bridge Deterioration Using Case-Based Reasoning." *Journal of Infrastructure Systems*, 8(3), pp. 86-95.

66. National Education Association (NEA) (April 2000). "Modernizing Our Schools: What will it cost.".

67. People's Daily (2005). "http://english.peopledaily.com.cn/200209/24/eng20020924_103783.shtml" Accessed October 31, 2005.

68. Pontis Release 3.0 (1995). "User's Manual", AASHTO, Washington DC.

69. RECAPP (2002). "RECAPP® 1.0 User Guide." Physical Planning Technologies Inc. PPTI, Toronto, Canada.

70. RECAPP (2003). "RECAPP® 1.0 Training Workshop Manual." Physical Planning Technologies Inc. PPTI, Toronto, Canada, 2003.

71. RS Means Company. (2000). "Means Building Construction Cost Data", R. S. Means Company, Kingston, Mass.

72. Rugless, J. (1993). "Condition Assessment Surveys." *Facilities Engineering Journal,* 21(3), pp. 11-13.

73. Seo, J. H. (1994). "A Lagrangean Relaxation and Network Approach to Large-Scale Optimization for Bridge Management Systems." Ph.D. thesis Texas A&M University.

74. Shekharan, A. R. (2000). "Solution of pavement deterioration equations by genetic algorithms." Transportation Research Record: *Journal of the Transportation Research Board*, No.1669, TRB, Washington, D.C., pp. 101-106.

75. Shohet, I. M. and Perelstein, E. (2004). "Decision Support Model for the Allocation of Resources in Rehabilitation Projects." *Journal of Construction Engineering and Management.* ASCE, 130(2), 249 – 257.

76. Sianipar, P. M. M., and Adams, T. M. (1997). "Fault-tree model of bridge element deterioration due to interaction." *Journal of Infrastructure Systems*, 3(3), 103–110.

77. Statistics Canada (1995). "Capital expenditures on construction by type of asset."

78. The Canadian Urban Transit Association (Sept. 1999). "The need for an Infrastructure Program for Transit: A presentation to Transport Canada." Urban Transit Association.

79. The National Research Council of Canada (1999). "Municipal Infrastructure in Canada: Inventory, Replacement Cost of Assets and Needs." Guy Felio, NRC, Canada.

80. Transportation Association of Canada TAC (1999). "Primer on Highway Asset Management Systems." Ottawa, March 1999.

81. U.S. Census Bureau (1999). “Statistical abstract of the Unites States, construction and housing.”

82. U.S. Department of Education (1999). “Digest of Education Statistics.”

83. UNIFORMAT II (2005). “UNIFORMAT II Element Classification for Building Specifications, Cost Estimating, and Cost Analysis.” http://www.bfrl.nist.gov /oae/publications/nistirs/6389.pdf, Accessed October 31, 2005.

84. Uzarski, D. R. (2002). ”Condition assessment Manual for Building Components for use with BUILDER Version 2.1.” U.S. Army, *Engineering Research and Development Centre - Construction Engineering Research Laboratory (ERDC-CERL)*, Champaign, IL, USA, March 2002.

85. Uzarski, D. R., Burley, L.A. (1997). ”Assessing Building Condition by the Use of Condition Indexes.” *Proceedings of Infrastructure condition assessment: art, science, and practice*, ASCE, Boston, Massachusetts, August 25-27, pp. 365-374.

86. Vanderbei, R.J. (2001). “Linear Programming: Foundation and Extension.” Second Edition, Graduate Textbook, Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544, “www.princeton.edu/~rvdb/LPbook/onlinebook.pdf”, Accessed December 23rd, 2005.

87. Vanier, D.J. (1999). “Why Industry Needs Asset Management Tools.” NRCC Seminar Series, Innovations in Urban Infrastructure 1999, APWA International Public Works Congress, Denver, USA, pp 11-25.

88. Vanier, D.J. (2001). ”Asset Management: “A” to “Z”.” Seminar Series, Innovations in Urban Infrastructure, APWA International Public Works Congress, Philadelphia, USA.

89. Warhus, J. P., Mast, J. E., and Nelson, S. D. (1995). “Imaging Radar for Bridge deck Inspection.” *Proceedings of the SPIE Conference on the Non-destructive Evaluation of Aging Infrastructure*, Oakland, CA, June 6-7 1995. SPIE Volume 2456, pp. 296-305.

90. Yang, J., Lu, J. J., Gunaratne, M., and Xiang. Q. (2003). “Overall pavement condition forecasting using neural networks – An application to Florida highway network.” *Proceedings of the 82nd Transportation Research Board Annual Meeting (CD-ROM),* Washington, D.C.

APPENDIX A

Solving IP optimization problems using Branch-and-Bound

Branch and bound algorithms are a variety of adaptive partition strategies which have been proposed to solve global optimization models. These algorithms are based on partition, sampling, and subsequent lower and upper bounding procedures. These operations are applied iteratively to the collection of active ('candidate') subsets within the feasible set of solution. Their exhaustive search feature is guaranteed in similar spirit to the analogous integer linear programming methodology (Mathworld 2005). This approach is used in Excel Solver, as a procedure for IP problems. The branch-and-bound procedure can be summarized in the following four steps (exemplified for a maximization problem), as explained in More and Weatherford 2001:

1. Solve the original ILP formulation as a relaxed LP (without the integer constraint). The objective function for the relaxation is the value of the ILP's upper bound. If the optimal solution is all-integer, it is optimal for the ILP;

2. If the LP relaxation has some integer variable at a fractional value, form two sub-models from this parent branch, so create two new unsolved LP sub-models (the successors) with the property that the optimal solution to one of the successor ILPs will be the optimal solution to the parent ILP. The branching may be accomplished by taking any fractional variable, say x_i*, if the optimal solution to the parent's relaxation. Let [x_i*] be the truncation of x_i* to its integer part. Then [x_i*] + 1 is the next integer larger than x_i*. One successor sub-model will be the parent's LP model augmented by the constraint in $x_i \leq [x_i*]$. The other successor sub-model is formed by augmented the parent's LP model with $x_i \geq [x_i*] + 1$.

3. Commence with any unsolved sub-model in step 2 and optimize it as a relaxed LP sub-model. If the optimal solution is all-integer, evaluate its objective function at this point. Compare the objective function of the best ILP model's solution found so far with this relaxed sub-model's objective function. If the relaxed sub-model's objective function is worse than the best ILP solution found so far, don't continue to eliminate any remaining fractional variables in the sub-model, as it is already an inferior candidate; instead, throw that sub-model away and continue with another one. If the relaxed sub-model is better than the best ILP so far, then proceed to eliminate any other fractional variables it may have using the constraint

augmentation procedure by going back to step 2. If all remaining relaxed sub-models have integer solutions, go to step 4; otherwise go back to step 2.

4. The optimal solution to the original ILP is the all-integer solution of some sub-model that produced the best value of the objective function found so far.

It should be clear from this cryptic summary that the original ILP is decomposed into a growing sequence of LP sub-models, each augmented with additional constraints on any fractional decision variables, which are then in turn re-optimized.

The case of binary variables is a special case of the integer variables (only 0 or 1); branch-and-bound technique is still the suitable approach to solve this problem. In this case, with the assumption that the branching on the binary variable is y_1, then one successor will have $y_1 = 0$ and the other will have $y_1 = 1$.

An example of the branch-and-bound calculation (taken from Vanderbei R.J. 2000) is as follows:

The following materials are from various sources including web resources and two main books: (1) "Decision Modeling with Microsoft Excel" by Moore and Weatherford; and (2) "Linear Programming: Foundations and Extensions" by Robert J. Vanderbei (2001).

$$\mathit{Max}\ 17\,x_1 + 12\,x_2$$

Subject to:

$$10\,x_1 + 7\,x_2 \leq 40$$

$$x_1 + x_2 \leq 5$$

$$x_1, x_2 \geq 0$$

$$x_1, x_2\ \textit{integers.}$$

Figure 1 shows the graphical representation for the problem. The linear programming problem obtained by ignoring the integrality constraint is called the LP-relaxation. Since it has fewer constraints, its optimal solution provides an upper bound ζ^0 on the optimal solution ζ^* to the integer programming problem. Figure 1 shows the feasible points for the integer programming problem as well as the feasible polytope for its LP-relaxation.

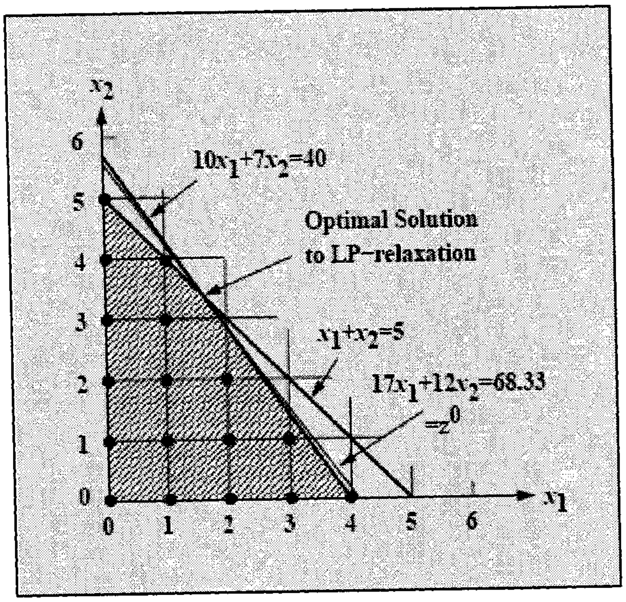


Figure 1: An integer programming problem. The dots represent the feasible integer points, and the shaded region shows the feasible region for the LP- relaxation.

The solution to the LP-relaxation is at (x_1, x_2) = (5/3, 10/3), and the optimal objective value is 205/3 = 68.33. Rounding each component of this solution to the nearest integer, the solution will be (2, 3), which is not feasible. The feasible integer solution that is closest to the LP-optimal solution is (1, 3), but it can be seen from Figure 1 that this solution is not the optimal solution to the integer programming problem. In fact, it is easy to see from the figure that the optimal integer solution is either (1, 4) or (4, 0). The solution is as follows:

Consider variable x_1 in the optimal solution to the LP- relaxation. Its value is 5/3 (i.e., 1.667). In the optimal solution to the integer programming problem, it will be an integer. Hence, it will satisfy either $x_1 \leq 1$ or $x_1 \geq 2$. Let's consider these two cases separately. Let P1denote the linear programming problem obtained by adding the constraint $x_1 \leq 1$ to the LP- relaxation, and let P2 denote the problem obtained by including the other possibility, $x_1 \geq 2$. The feasible regions for P1 and P2 are shown in Figure 2. Let us study P1 first. It is clear from Figure 2 that the optimal solution is at (x_1, x_2) = (1, 4) with an objective value of 65. This is the first feasible solution to the integer programming problem (best-so-far).

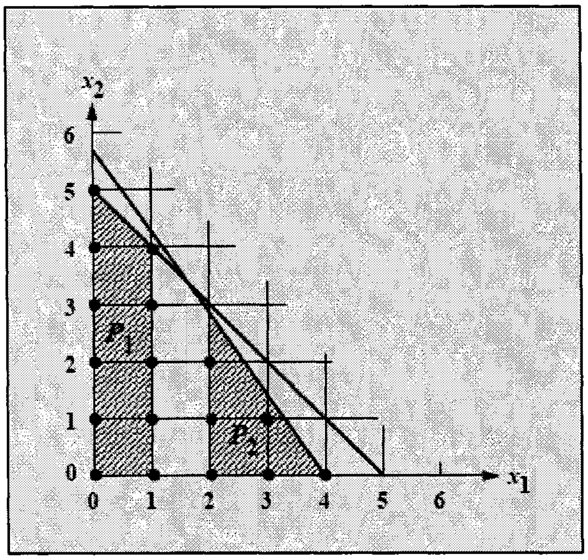


Figure 2: The feasible sub-regions formed by the first branch

Now let's consider P2. From Figure 2, the optimal solution is at $(x_1, x_2) = (2, 20/7)$. In this case, the objective function value is $478/7 = 68.29$. If this value had turned out to be less than the best-so-far value, then the best-so-far is the solution, since any integer solution that lies within the feasible region for P2 would have a smaller value yet. But this is not the case, and so the investigation must continue as a systematic search. Since $x_2 = 20/7 = 2.86$, P2 will be divided into two sub-problems, one in which the constraint $x_2 \leq 2$ is added and one with $x_2 \geq 3$ added.

Before considering these two new cases, it is noted that a tree of linear programming sub-problems started to be developed. This tree is called the enumeration tree. The tree as far as the solution reached is shown in Figure 3. The double box around P1 indicates that that part of the tree is done: i.e., there are no branches emanating from P1, it is a leaf node. The two empty boxes below P2 indicate two sub-problems that have yet to be studied. Let's proceed by looking at the left branch, which corresponds to adding the constraint $x_2 \leq 2$ to what we had before. Let's denote this sub-problem by P3. Its feasible region is shown in Figure 4, from which the optimal solution is at (2.6, 2). The associated optimal objective value is 68.2. Again, the solution is fractional. Hence, the process of subdividing must continue. This time consider the two cases: either $x_1 \leq 2$ or $x_1 \geq 3$.

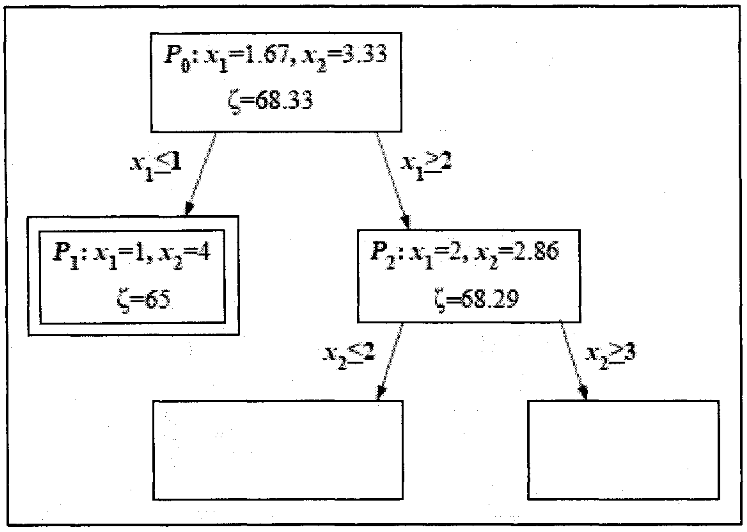


Figure 3: The beginnings of the enumeration tree

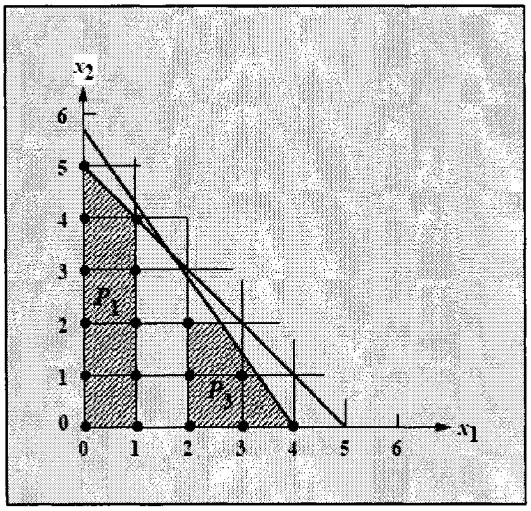


Figure 4: The refinement of P1 to P3

Figure 5 shows the enumeration tree as it now stands. At this juncture, there are three directions in which the search can be preceded, either to study the other branch under P2 or work on one of the two branches sitting under P3. If all the problems on a given level of the tree were systematically solved before going deeper, what is referred to "breadth-first search" had been performed.

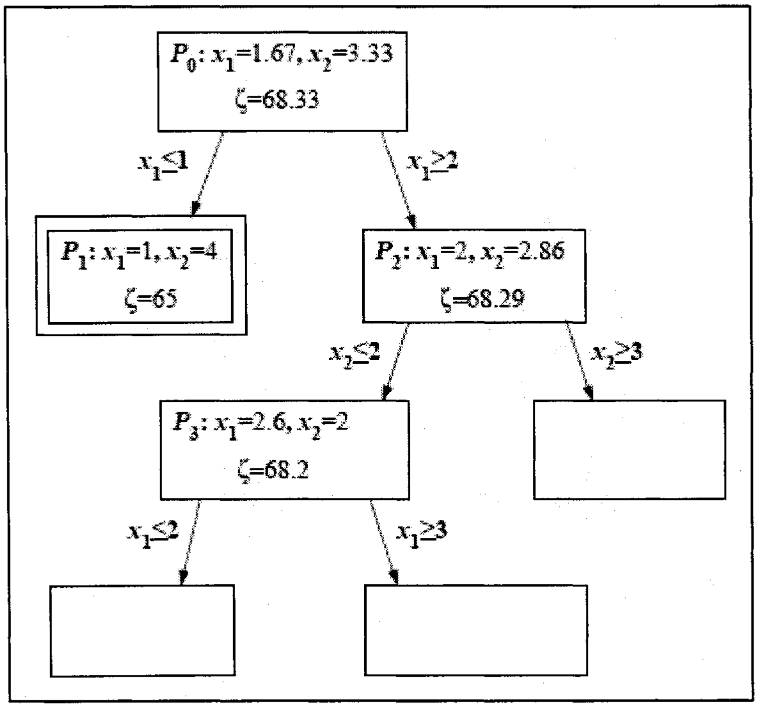


Figure 5: The enumeration tree after solving P3

On the other hand, going deep before going wide is called a depth-first search. A depth-first search is better, and to be specific, let us always choose the left branch before the right branch (in practice, there are much better rules that one can employ here). So the next linear programming problem is the one that we get by adding the constraint that x_1 ≤2 to the constraints that defined P3. Let us call this new problem P4. Its feasible region is shown in Figure 6. It is easy to see that the optimal solution to this problem is (2, 2), with an objective value of 58. This solution is an integer solution, so it is feasible for the integer programming problem. But it is not better than the best-so-far, as such this branch can be neglected (no deeper search are required on this node).

Since problem P4 is a leaf in the enumeration tree, it is important to work back up the tree looking for the first node that has an unsolved problem sitting under it. For the case at hand, the unsolved problem is on the right branch underneath P3. Let us call this problem P5. It too is depicted in Figure 6. The optimal solution is (3, 1.43), with an optimal objective function value of 68.14. Since this objective function value is larger than the value of the best-so-far integer solution, it needs further investigations by dividing into two possibilities, either x_2 ≤1or x_2 ≥2. At this point, the enumeration tree looks like that shown in Figure 7.

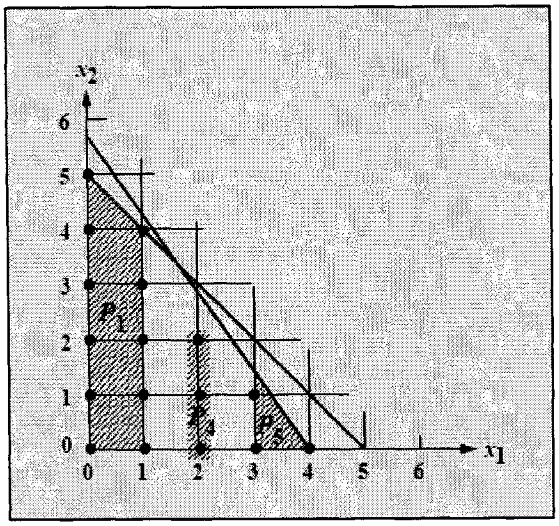


Figure 6: The refinement of P3 to P4

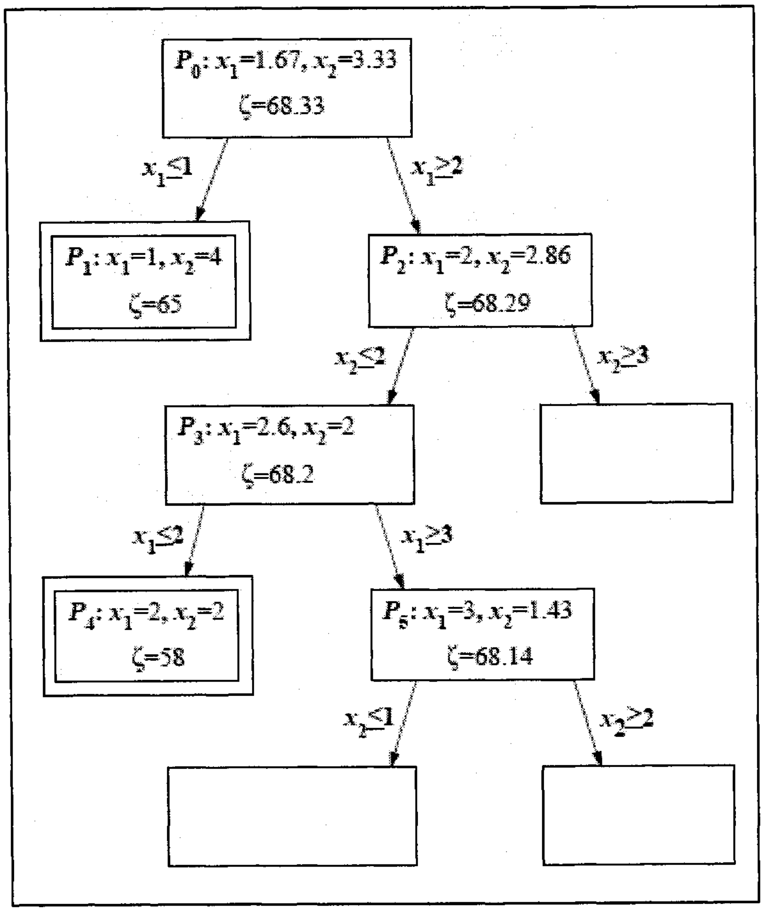


Figure 7: The enumeration tree after solving P5. The double box around P4 indicates that it is a leaf in the tree

Let P6 denote the linear programming problem under P5. Its feasible region is shown in Figure 8. The optimal solution is (3.3, 1), and the associated objective value is 68.1. Again, the solution is fractional and has a higher objective value than the best-so-far integer solution. Hence, it must be subdivided based on $x_1 \leq 3$ as opposed to $x_1 \geq 4$. Denoting these two problems by P7 and P8, their feasible regions are as depicted in Figure 9. The solution to P7 is (3, 1), and the objective value is 63. This is an integer solution, but it is not better than the best-so-far. The node becomes a leaf, since the solution is integral. Hence, move on to P8. The solution to this problem is also integer, (4, 0). Also, the objective value associated with this solution is 68, which is a new record for feasible integer solutions. Hence, this solution becomes the best-so-far. The enumeration tree at this point is shown in Figure 10.

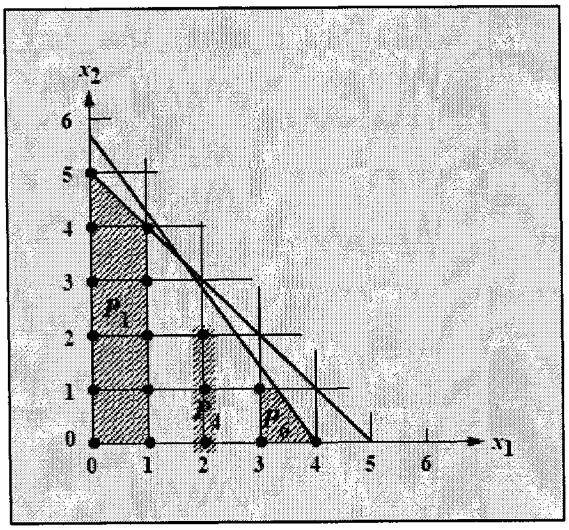


Figure 8: The refinement of P5 to P6

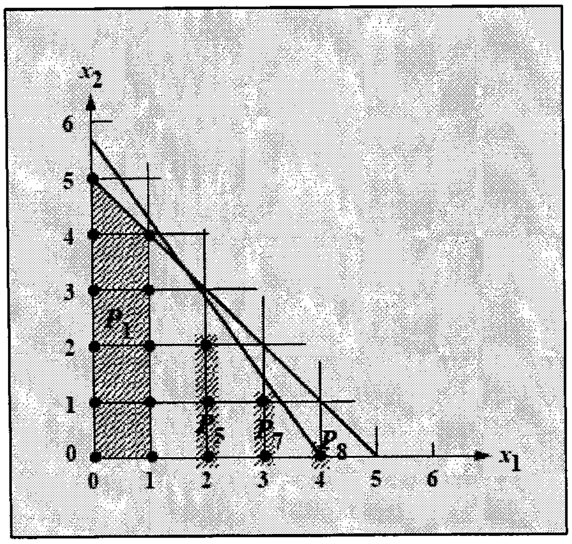


Figure 9: The refinement of P6 to P8

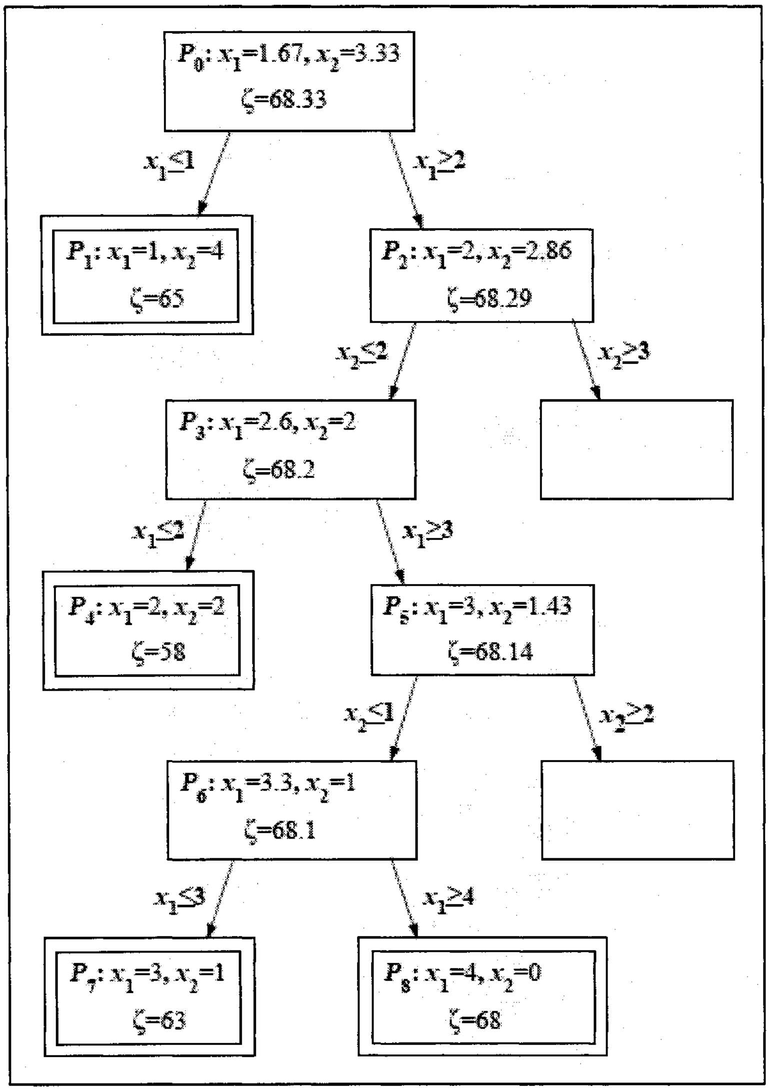


Figure 10: The enumeration tree after solving P6, P7, and P8

Now go back and solve the problems under P5 and P2 (and any sub-problems thereof). It turns out that both these sub-problems are infeasible, and so no more subdivisions are needed. The enumeration tree is now completely fathomed and is shown in Figure 11. The optimal solution to the original integer programming problem, as such, was found in problem P8. The solution is (x_1, x_2) = (4, 0), and the associated objective function value is 68.

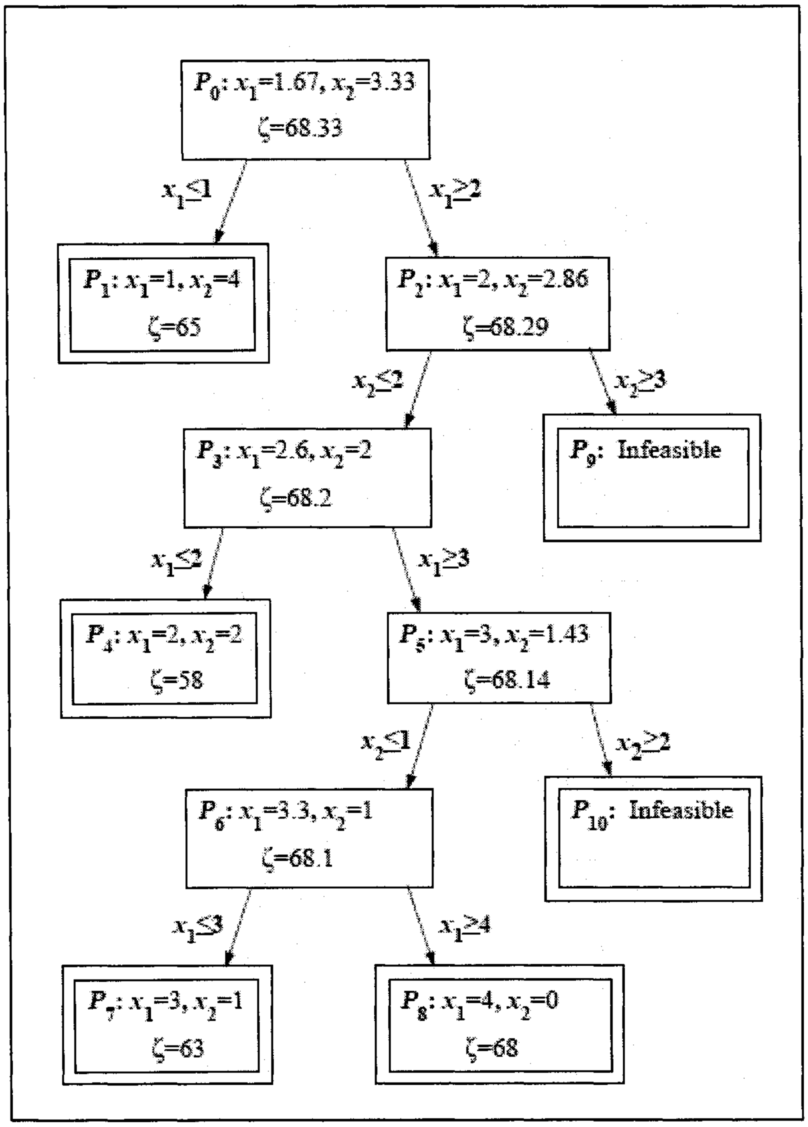


Figure 11: The complete enumeration tree