INTEGRATED BID PREPARATION WITH EMPHASES ON RISK ASSESSMENT USING NEURAL NETWORKS

Tarek M. Hegazy

A Thesis

in

The Centre for Building Studies

Presented in Partial Fulfilment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montreal, Quebec, Canada

July 1993

National Library of Canada
Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services Branch
Direction des acquisitions et des services bibliographiques

395 Wellington Street
Ottawa, Ontario
K1A 0N4

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

ISBN 0-315-97647-0

Name TAREK HEGAZY

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

Civil Engr — **SUBJECT TERM**

0543 — **SUBJECT CODE**

U·M·I

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS
Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION
General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523
Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS
Language
- General 0679
- Ancient 0289
- Linguistics 0290
- Modern 0291

Literature
- General 0401
- Classical 0294
- Comparative 0295
- Medieval 0297
- Modern 0298
- African 0316
- American 0591
- Asian 0305
- Canadian (English) 0352
- Canadian (French) 0355
- English 0593
- Germanic 0311
- Latin American 0312
- Middle Eastern 0315
- Romance 0313
- Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY
Philosophy 0422
Religion
- General 0318
- Biblical Studies 0321
- Clergy 0319
- History of 0320
- Philosophy of 0322

Theology 0469

SOCIAL SCIENCES
American Studies 0323
Anthropology
- Archaeology 0324
- Cultural 0326
- Physical 0327

Business Administration
- General 0310
- Accounting 0272
- Banking 0770
- Management 0454
- Marketing 0338

Canadian Studies 0385
Economics
- General 0501
- Agricultural 0503
- Commerce-Business 0505
- Finance 0508
- History 0509
- Labor 0510
- Theory 0511

Folklore 0358
Geography 0366
Gerontology 0351
History
- General 0578
- Ancient 0579
- Medieval 0581
- Modern 0582
- Black 0328
- African 0331
- Asia, Australia and Oceania 0332
- Canadian 0334
- European 0335
- Latin American 0336
- Middle Eastern 0333
- United States 0337

History of Science 0585
Law 0398
Political Science
- General 0615
- International Law and Relations 0616
- Public Administration 0617

Recreation 0814
Social Work 0452
Sociology
- General 0626
- Criminology and Penology 0627
- Demography 0938
- Ethnic and Racial Studies 0631
- Individual and Family Studies 0628
- Industrial and Labor Relations 0629
- Public and Social Welfare 0630
- Social Structure and Development 0700
- Theory and Methods 0344

Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES
Agriculture
- General 0473
- Agronomy 0285
- Animal Culture and Nutrition 0475
- Animal Pathology 0476
- Food Science and Technology 0359
- Forestry and Wildlife 0478
- Plant Culture 0479
- Plant Pathology 0480
- Plant Physiology 0817
- Range Management 0777
- Wood Technology 0746

Biology
- General 0306
- Anatomy 0287
- Biostatistics 0308
- Botany 0309
- Cell 0379
- Ecology 0329
- Entomology 0353
- Genetics 0369
- Limnology 0793
- Microbiology 0410
- Molecular 0307
- Neuroscience 0317
- Oceanography 0416
- Physiology 0433
- Radiation 0821
- Veterinary Science 0778
- Zoology 0472

Biophysics
- General 0786
- Medical 0760

EARTH SCIENCES
Biogeochemistry 0425
Geochemistry 0996
Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES
Environmental Sciences 0768
Health Sciences
- General 0566
- Audiology 0300
- Chemotherapy 0992
- Dentistry 0567
- Education 0350
- Hospital Management 0769
- Human Development 0758
- Immunology 0982
- Medicine and Surgery 0564
- Mental Health 0347
- Nursing 0569
- Nutrition 0570
- Obstetrics and Gynecology 0380
- Occupational Health and Therapy 0354
- Ophthalmology 0381
- Pathology 0571
- Pharmacology 0419
- Pharmacy 0572
- Physical Therapy 0382
- Public Health 0573
- Radiology 0574
- Recreation 0575
- Speech Pathology 0460
- Toxicology 0383

Home Economics 0386

PHYSICAL SCIENCES
Pure Sciences
Chemistry
- General 0485
- Agricultural 0749
- Analytical 0486
- Biochemistry 0487
- Inorganic 0488
- Nuclear 0738
- Organic 0490
- Pharmaceutical 0491
- Physical 0494
- Polymer 0495
- Radiation 0754

Mathematics 0405
Physics
- General 0605
- Acoustics 0986
- Astronomy and Astrophysics 0606
- Atmospheric Science 0608
- Atomic 0748
- Electronics and Electricity 0607
- Elementary Particles and High Energy 0798
- Fluid and Plasma 0759
- Molecular 0609
- Nuclear 0610
- Optics 0752
- Radiation 0756
- Solid State 0611

Statistics 0463

Applied Sciences
Applied Mechanics 0346
Computer Science 0984

Engineering
- General 0537
- Aerospace 0538
- Agricultural 0539
- Automotive 0540
- Biomedical 0541
- Chemical 0542
- Civil 0543
- Electronics and Electrical 0544
- Heat and Thermodynamics 0348
- Hydraulic 0545
- Industrial 0546
- Marine 0547
- Materials Science 0794
- Mechanical 0548
- Metallurgy 0743
- Mining 0551
- Nuclear 0552
- Packaging 0549
- Petroleum 0765
- Sanitary and Municipal 0554
- System Science 0790

Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY
General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

ABSTRACT

INTEGRATED BID PREPARATION WITH EMPHASES ON RISK ASSESSMENT USING NEURAL NETWORKS

Tarek M. Hegazy, Ph.D.
Concordia University, 1993

Construction estimating works as the basis for various strategic decisions regarding the preparation of bid proposals, procurement plans, various levels of schedules, and job cost control. Under the highly risky environment of the prevalent competitive bidding practice, preparation of realistic estimates pertaining to those management decisions has been a complex task that is often performed on an ad hoc and piecemeal manner. Conventional procedures and tools have proved inadequate to provide a structured decision aid that, under such environment, maximizes the contractor's chances of winning a job with maximum potential profit, and further generates practical baseline plans needed for job control to maintain this profit. Yet, the situation has been translated into a high percentage of business failures, a high potential for claims, and at best a low profit margin in the industry.

This research presents a methodology for an integrated cost estimation and bid preparation, with emphasis on the assessment of bidding risks and optimum markup estimation. The methodology utilizes available tools (algorithms, database management systems, and AI-based techniques) that can benefit from current industry practice and provide an adequate decision aid during bid preparation. The

methodology facilitates integration among estimating, planning and scheduling, and bid unbalancing. It incorporates enhancements to the various functions that cover the quantitative aspects of an estimate including: direct and indirect cost estimation, planning and scheduling, and resource utilization. This enables detailed estimates of costs and durations to be generated for all the project tasks, with minimal redundancy and in less time. Such estimates also establish the baselines needed for efficient job control.

For practicality, the methodology accounts for the qualitative (risk-related) factors that play a vital role in the preparation of competitive bid proposals (e.g., competition, market conditions, and contractor keenness for the job). The methodology utilizes Neural Networks, an AI-based technique that employs a learning mechanism and emulates the human ability to solve pattern recognition tasks similar to many problems encountered in construction. This technique is introduced as a new tool to the industry, incorporating several potential applications. A neural network model is designed and used to arrive at an optimum markup value that maximizes the contractor's potential profit and predicts the probability of winning the job at such level of profit, in response to the project risk pattern. The methodology then utilizes the data obtained through the detailed estimate to optimally unbalance the final bid, in an effort to improve the contractor's cash flow while maintaining his competitiveness. A PC-based prototype is developed to automate the bid preparation process and an example application is presented in order to demonstrate the effectiveness and practicality of the proposed methodology. The proposed integrated methodology contributes to current automation efforts in construction and its modular architecture allows for further enhancements and expansions. The developments made with respect to the markup estimation problem demonstrates the powerful capabilities of neural networks and the potential benefits of deriving analogy-based solutions to complicated construction problems that are characterized by high uncertainty. This approach could readily be utilized in other domains in construction management where solutions are based primarily on holistic analogy and traditional algorithmic solutions are inadequate.

ACKNOWLEDGEMENTS

Humble thanks are first offered to God.

I wish to express my deepest gratitude and sincere appreciation to my supervisor Dr. O. Moselhi, for his academic as well as financial support (NSERC and FCAR team grant), valuable advice, and constructive guidance throughout all the stages of this study.

I wish also to express my deepest gratitude to my co-supervisor Dr. P. Fazio, for his unbounded support for this research both academic and financial (NSERC and FCAR team grant). His practical comments are deeply appreciated.

My thanks to all those, at the Centre for Building Studies, Directors, Staff and Faculty members, Secretaries, CABD lab operators, and colleagues: Ismail Budaiwi, Adel Abdou, Khalid El-Rayes, Ahmed Moktar, and Pasit Lorterapong, who helped a great deal in carrying out this thesis.

I am deeply indebted to Sam Habib of Constructions Louisbourg Ltée for his encouragement, financial support, and practical advice which was of great asset to this thesis.

I would like to thank my colleagues at Suez Canal Authority and Research Centre in Ismailia, Egypt, for their support and encouragement.

I wish to express my special thanks to my family; my wife and sons who have stood beside me, encouraged me, shared with me many happy moments, and endured a lot of inconveniences along the course of this study.

Finally, I would like to dedicate this thesis to my father and mother who have been a continuous source of love and encouragement throughout my whole life.

Table of Contents

Chapter Page

Chapter Page

NOMENCLATURE

P.E. = processing element;

X = input to a P.E.;

O = actual output of a P.E.;

D = desired output of a P.E.;

W = weight of a connection incoming to a P.E.;

δ = error between O and D;

θ_i = a bias for the i th P.E.;

η = learning rate coefficient;

P = number of training examples;

E = mean square error of a neural network over all training examples;

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 General

A large percentage of all construction work, particularly government and public sector-financed work, is awarded on the basis of competitive bidding. Under this system, construction contracts are mainly either a lump sum or a unit price form and projects are delivered mainly through the conventional design-bid-build approach. Traditionally, the owner invites a number of prospective contractors to compete for the construction of a project. The award is made on the basis of the proposed bid prices with the project generally awarded to the **"lowest responsible"** bidder. This approach ensures several benefits to the owner including: (1) known final cost before construction starts (provided that no changes are introduced); (2) lowest cost of constructing the facility; and (3) minimum risks since budgets can be arranged before hand. As for the contractor, actual total costs are not known until the end of the project and several risks are anticipated to greatly affect those costs, including: omissions, unforseens, expected competition, owner changes and attitude, regulatory rules at project location, resource availability, and acts of God. Added to that are the fluctuation in market conditions, inflation and escalation rates. For a contractor to be successful in such a business environment, his compiled bid prices should be low enough to win a sufficient number of projects, and, high enough to realize a reasonable net profit.

The difficulty in compiling a bid price stems from the inadequacy of current procedures and tools to achieve the desired balance.

Frequently, bid preparation and cost estimation are referred to as synonyms. Most cost estimation software systems, for instance, are developed to aid in computing the direct, and to a lesser extent the indirect, costs as the major component(s) in the bid proposal, presuming that this is enough and sufficient for bid preparation. In practice, however, contractors generally prepare bid proposals by performing four main functions: 1) breakdown of cost items and related direct cost computations; 2) assignment of indirect costs in view of the project expected schedule and resource use; 3) assessment of the project risks and estimation of optimum markup; and 4) compiling the bid into unit price(s) assigned in a balanced or unbalanced manner to the contract item(s). On one hand, direct and indirect costs may, in traditional constructed facilities, amount to up to 90% of the total bid price. The procedures for their estimation are generally structured, incorporating algorithmic computations based on accurate quantity take-off and good record keeping about productivity and unit costs. On the other hand, the other aspects of bid preparation, particularly number 3 above, generally contribute less to the total bid amount, although may in situations be a predominant bid item. These aspects require assessment of the project risks through experienced qualitative judgement and sometimes a difficult to explain gut feeling, with less computations involved. The inadequacy of available tools to consider these risk related aspects which are severe in competitive bidding have contributed to a lack of structured bid preparation methodology and have resulted in a large percentage of failures in the industry (Kangari 1988). Yet, most available estimation tools have been powerful in certain aspects mainly related to accounting (Rabchuk 1991), however, insufficient to address all the elements of the bid preparation process (Moselhi et al. 1993).

Due to the severity of the competitive bidding approach, different types of negotiated contractual agreements have evolved (i.e., cost plus with guaranteed maximum price) to allow for better risk-sharing between owners and contractors. However, these types of contracts apply mostly to highly risky R & D and fast-track delivered projects. For these projects, either the availability of reliable technology is limited and owners can not benefit from the competition, or construction has to proceed although design is not yet complete. In both cases, costs cannot be realistically pre-estimated and the amount of risk involved is relatively high to be taken by one party. Also, cost plus contracts often lead to higher final costs known only at the end of the project and thus, they do not suit well the planning and fiscal requirements of government agencies which are the major spender on construction projects. Despite the disadvantages of the competitive bidding approach, it is still the prevailing contractual practice, and contractors must deal practically with its realities.

1.2 Construction Bid Preparation

Construction estimating is one of the most crucial functions in project management. On one hand, accurate bid proposals maintain contractor's success and establish his potential profits. Inaccurate estimates could result in either significant monetary losses, if the estimates are too low, or no jobs at all if too high. On the other hand, estimating practical cost and schedule baselines is certainly most important for efficient job control to track the planned profit.

Construction estimating is a complex and a crude process. This is due to the absence of standardization of conditions from one job to the other. In addition, the inherently complex factors of weather, labour, material and locality make the computation of exact construction costs a matter of accident than of design. These estimating difficulties are further exacerbated in the competitive environment, that is so prevalent in the construction industry. In order to win a job and maintain a fair profit under such an environment, the contractor needs not only to consider the quantitative aspects of an estimate but also several qualitative (risk-related) aspects including: market activity and volatility, competition, project anticipated risks, and organization inherent factors. These qualitative factors strongly influence markup and bid unbalancing decisions. The importance of estimating an optimum markup to the development of practical bid proposals stems from the following reasons:

- Markup is a major risk assessment measure: when developing a bid, contractors appear to have three ways to compensate for risk (de Neufville and King 1991): 1) estimate optimum markup depending on the risk; 2) add contingencies based on the risk of each item being estimated; and 3) a combination of both. To estimate contingency, formal decision and risk analysis techniques (e.g., Monte Carlo simulation; decision trees; and utility theory) have been used by researchers to adjust cost items for risks related to estimating inaccuracies, schedule slippages, construction problems, and miscellaneous unforseens (Birnie and Yates 1991; Shafer 1991; Isaac 1990; Yeo 1990; and Kerridge 1986). The present study adopts the first strategy to assess the project risks and estimate an optimum markup accordingly. This is coupled with the development of practical cost and schedule estimates, thus, reducing the risks of contingency-related items. This strategy is advantageous when the time permitted for bid preparation is limited. Contingency estimation could, however, be used for managing contingency during construction rather than in the bidding stage. Such a contingency management policy, in addition to other corporate risk management policies such as insurance and risk prevention (e.g., Al-Bahar and Crandall 1991; Al-bahar and Gerwick 1990; Frantz 1990) as well as a suitable claim management policy (e.g., Manzanera and Bubshait 1991; Younker 1990), enable contractors to reduce their cost of risk and, in some cases, have an edge over other contractors by reducing their markups.

- Markup plays a vital role in winning contracts: after the decision to bid is made, contractors generally determine their bid price by performing two distinct functions: (1) estimation of the project costs (direct and indirect); and (2) determination of the optimum markup. Cost estimation is mainly a quantitative process that lends itself to algorithmic solutions. Despite the difficulties generally associated with cost estimation for construction projects and the short time permitted to prepare bids, most of the tools and methodologies used, although lack integration, are highly structured and have considerably matured over the last two decades. In addition, there has been ample published cost data and computer software systems with comprehensive databases tied with peripherals including digitizers, that aid in the cost estimation function. It has been observed, therefore, that bidders' cost estimates may well be very similar due to common factors, including: they all have access to the same labour supply, use the same types of equipments, obtain supplies and materials from the same sources, and have somewhat comparable supervisory capabilities (Park 1979). As such, the variations in competitors' bids are mainly due to their selected markups which reflect the bidding policies chosen to achieve their own objectives (Skitmore 1989). The estimated markup, therefore, plays a vital role in striking the optimum balance between a bid price that is as "practically low" as possible to win the job, and yet as "practically high" as possible to maximize profit.

- Contractors have no adequate tools for estimating markup: optimum markup estimation is a decision problem that is so highly unstructured that it is very difficult to analyze and formulate an adequate solution mechanism. It is both time consuming and complicated to identify all the related factors that form a rational basis for such decisions, analyze their individual strengths, and then quantify their combined impact on the decision. This is particularly so since the bidding environment is often characterised by high uncertainty, intense competition, and difficult to quantify risks. The usual practice, however, is to make bid decisions on the basis of intuition, derived from a mixture of gut feeling, experience, and guesses (Ahmad 1990). This implies some form of pattern recognition task to derive a solution based on analogy with previous encounters, rather than computation or deep reasoning about the problem elements. Such a decision making process is normally implicit and highly dependent on the experience of the decision maker who can hardly explain and describe his decision-making rationale. The complexity of the problem and the inability of traditional algorithmic tools to adequately model the decision-making process and the uncertainty inherent in the problem environment have left contractors with little guidance in formulating successful bidding strategies.

With the advent of affordable, reliable, and more efficient computer systems in the 1980s, intensive research work has been conducted in the area of cost estimation, aiming at the development of effective tools for contractors to compile competitive bids in a timely and practical manner. On one hand, a large number of commercial computer software systems (Arditi and Riad 1988), have been developed to assist contractors organize their direct and indirect cost estimates. An equally large number of books and publications have also provided ample cost data and/or demonstrated methodologies and custom-designed computerized systems, that aid contractors in cost estimation and/or job cost control (e.g., Means 1990, Neil 1982, Suckarieh 1984, Riggs 1988, Rayburn 1989, and Eldin 1989). On the other hand, various research efforts have been conducted to develop decision aids regarding the estimation of an optimum markup value and the determination of a bid unbalancing method. Several probability-based bidding strategy models have been proposed since the mid 1950s, providing different assessments to the optimum markup problem, mainly in account of job profitability and expected competition (e.g., the bidding strategy models of Friedman 1956, Gates 1967, and Carr 1987). Other research efforts have been directed towards the development of optimum bid unbalancing methods that improve the contractor's cash flow, using mathematical programming (e.g., Stark and Mayer 1983).

Despite the proliferation of available estimating tools and their underlying methodologies, estimators develop procedures of their own to compile the cost of construction, based mainly on their experience and in an intuitive manner that suits their work environment (Peurifoy and Oberlender 1989). This has been attributed to a lack of generally accepted estimating guidelines (Carr 1990) and a lack of recognition of the steps essential to a successful bid preparation (Hicks 1992). This in addition to the inability of existing tools to provide integration among the different efforts necessary for an overall methodology. The main deficiencies of available tools and procedures could be summarized as follows (Moselhi et al. 1991d):

- Lack of integration: the cost and time of construction are highly dependent on the quantity of work, cost of resources, date of execution, and productivity of working crews, pertaining to the method of construction adopted. Thus, cost estimation, planning and scheduling and control operations are highly interrelated management functions, although usually treated as three distinct and isolated ones (Stephenson 1990). Most of available tools are designed to perform one of the three functions, leading to high redundancy, time consuming processing operations, and higher probability of introducing errors.

- Lack of structured methodology: Although the use of the WBS has been strongly supported by researchers seeking integration among estimating, planning and scheduling, and control (Neil 1982, Meuller 1986, Adrian 1987, Keisk and Selby 1990, Riggs 1990, Al-Tabtabai and Diekman 1990), Nevertheless, the construction industry in general and the building sector in particular have not yet embraced WBS and integrated project control.

- Lack of practicality: available systems do not incorporate procedures for optimum markup estimation and optimum bid unbalancing, into the cost estimation methodology, leaving them to the intuitive judgement of the contractor. This may be due to the subjective nature of these decisions and also the lack of simplified procedures pertaining to their resolution. The optimum markup estimation has been a subject of large controversy and debate regarding the validity and practicality of the models developed and their associated results. Also, the bid unbalancing procedure developed by Stark and Mayer requires large input data that relates to both cost and schedule, and as such is limited to integrated cost and schedule environments which, if yet existed, are unlikely to be employed in the bidding stage.

- Inefficient utilization of available information: contractors often have information regarding their own organization, economic conditions, market volatility, project risks, and the bid competitors. However, available systems are not designed to store, process, and benefit from such valuable experience-based information.

The inconsistency between the highly detailed and accurate performance of most cost estimation tools and the inadequate decision aids regarding markup estimation, in addition to the complexity of the process itself, have contributed to a lack of reliability of the final outcomes of existing estimating tools. No matter what tools and methods are used, the high cost and large time associated with preparing detailed estimates (0.05% to 2% of the total project cost - Ahuja and Campbell 1988), yet, have not been recoverable by increase in the contractor's competitiveness and profitability. This has discouraged contractors to perform pre-bid detailed estimates despite the relevance of those details to a proper bidding decision. Consequently, estimates for bid proposals are more often based on less accurate methods, historical records, and/or unadjusted cost data without proper consideration of available resources, cash flow constraints and the project conditions. The result of a recent survey seems to support this argument (Tavakoli and Riachi 1990). The survey investigated the use of the Critical Path Method (CPM) of planning and scheduling among the top 400 Contractors rated by the Engineering News Record in 1987. The survey revealed that only 28.6% of the respondents use CPM for estimating and bidding, as compared to 94.6% using CPM in detailed planning of work prior to start of construction. Thus, estimates for cost and schedule base plans are most often performed after a contractor wins a job and is committed to a final cost and a total duration irrespective of his actual expenditures and execution time. The situation has been translated into a poorly structured estimating procedure, high percentage of business failures, high potential for claims, and at best a low profit margin in the industry (Kangari 1988).

1.3 Scope and Objectives

This research investigates the development of a methodology for an integrated cost estimation and bid preparation, with emphasis on the assessment of bidding risks and markup estimation. Such a methodology would utilize available tools (algorithms, database management systems, and AI-based techniques) that can benefit from current industry practice and provide adequate decision aid during bid preparation. The research primary objectives are stated as follows:

1. Develop a methodology that efficiently integrates the management functions of estimating, planning, and scheduling. The integrated methodology should minimize redundancy and reduce time and cost needed to prepare detailed pre-bid estimates of cost and schedule and compile practical, competitive, and potentially profitable bid proposals.

2. Identify the qualitative factors, inherent in the prevalent competitive bidding environment, which need to be considered in the assessment of bid preparation risks and the estimation of optimum markups.

3. Study the applicability of Neural Networks, an AI-based technique, for analogy-based solutions to construction problems. Accordingly, develop a risk assessment model that recognizes a project environment and provide a practical assessment to optimum markup estimation.

4. Develop a prototype based on the proposed methodology and demonstrate its use on an example application.

This study focuses on the competitive bidding process in which contractors compete against others for the right to construct a project. The study investigates integration among the three essential management function of estimating, planning and scheduling, and control performed by **"general contractors"** under the conventional "design then build" approach of project delivery. This approach is commonly used in the competitive bidding environment where, design is performed by an independent A/E organization and completed prior to bidding and construction stages. Such an approach exhibits lack of integration and coordination between the design and construction phases of projects, as compared to other project delivery approaches (e.g., turn-key; fast-track). The proposed system, therefore, adopts a modular structure to facilitate the required integration. It should be noted that the study deals mainly with medium size high-rise building projects (10-100) $ millions. The proposed integrated system, however, can be applied to other types including heavy civil projects. The emphasis on building projects is because of the large number of trades and subcontractors involved in constructing building projects and also the little perspective on how all these pieces fit together.

1.4 Approach

The study seeks the development an optimum problem-solving strategy for the bid preparation problem. This is achieved by selecting an optimum combination of problem-solving techniques (i.e., algorithm-based; knowledge-based; and analogy-based), tools, and development environments that: 1) suit the nature of all sub-problem components; 2) meet the accuracy required; 3) facilitate integration and effective flow of information among all system components; 4) meet user needs and level of experience; 5) utilize current industry practice and domain knowledge; 6) maximize the benefits and minimize the limitations of the individual techniques; and 7) are practical and cost effective within the system development constraints. The research approach consists of the following:

1. Design the bid preparation framework in a modular architecture incorporating four manageable components (modules), each represents one of the management functions needed for integration. These modules are: a cost estimation module, a planning and scheduling module, a risk assessment module, and a bid unbalancing module. These four modules are tied together by a core of databases, forming a comprehensive MIS.

2. Review of the theory and current developments in the different management functions included in the system modules. This helps identify, for each module, the most appropriate procedure amendable to the system, from the many practices described in the literature. Also it works as a system requirement analyses necessary to formulate practical specifications for the

integrated system.

3. Examine the literature to see how current techniques could be used or adapted to satisfy the specifications developed. Accordingly establish a practical and comprehensive methodology for bid preparation with adequate assessment for the project risks.

4. Select suitable software and hardware necessary for preparing a computerized model, and then proceed with the development and the validation of the system.

1.5 Thesis Organization

Chapter 2 presents a literature review of the state-of-the-art efforts that are related to the process of bid preparation: MIS and databases, cost estimation, planning and scheduling, risk assessment, and bid unbalancing. Neural networks are introduced as a tool for risk assessment at the markup level. Historical evolution of neural networks, their components and characteristics, and the different neural network paradigms are described.

Chapter 3 exhibits the establishment of the proposed bid preparation framework and methodology. System requirements and necessary considerations are identified and outlined. A general procedure of formulating an optimum problem-solving strategy is presented and applied to the bid preparation problem. The system architecture and components are then described along with the proposed bid preparation methodology.

Chapter 4 sets up the environment of the backpropagation paradigm for modelling the markup estimation problem. Basic mathematics of neural networks that led to the formulation of the backpropagation paradigm are described. The problems that face the development of practical backpropagation models are identified along with several techniques and heuristics used in the literature to overcome such problems. An application development procedure is then presented to structure the process of developing a neural network application.

Chapter 5 describes the development of the markup estimation model. The model utilizes current industry practice and accounts for the quantitative as well as qualitative factors that affect bidding decisions. Details of the design, knowledge acquisition and validation, implementation, and testing of the neural network model are described in this chapter.

Chapter 6 describes the adaptation and use of the markup estimation model as a decision support system for bidding in construction. The model is coded in a user friendly software that facilitate storage of previous bid encounters, data input, model adaptation to the user's environment, computations, sensitivity analyses, assessment of the probability of winning, and further integration with other modules for planning and scheduling and cost estimation.

Chapter 7 describes the development of a prototype for the integrated bid preparation system and a demonstration of its performance through a hypothetical example. The limitations of the prototype are described with respect to all its different modules and future development work is suggested. The example project is described and the prototype performance and results are discussed.

Chapter 8 is the thesis conclusion and a description of future extensions to current research.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Basically, the cost of construction is dependent on the quantity of work, cost of resources, date of execution, and productivity of working crews, pertaining to the method of construction adopted. Thus, cost estimation, planning, and scheduling are highly interrelated management functions that are highly relevant to the process of bid preparation.

This chapter presents a review of the theory and the state-of-art efforts in areas that are related to the process of bid preparation: MIS and databases, cost estimation, planning and scheduling, risk assessment, and bid unbalancing. This is for the purpose of identifying the procedures, among the many practices described in the literature, which are amendable to the integrated bid preparation system. Neural networks literature is also reviewed in order to identify their potential benefits as a tool for risk assessment at the markup level.

2.2 MIS and Databases

A Management Information System (MIS) is defined by Reynolds (1988) as a computer system capable of integrating data from many sources to provide data and information useful to support operations and decision making. When a MIS is successfully implemented, redundancy, time, and cost are reduced and automation is increased. The importance and the need for such information technology to the construction industry has been recognized (Barton 1985, and Meuller 1986) and the steps essential for proper design of a MIS reported as (Barton 1985):

1. Determine information needs.
2. Integrate information needs.
3. Produce the specifications.
4. Select relevant software and hardware.

In the literature, several researchers have carried out studies that provide insights for designing the proposed bid preparation system (e.g., Ndekugri and McCaffer 1988, Baxendale 1990, Paek and Lavata 1990). Ndekugri and McCaffer 1988 have analyzed the flow of information as well as the inputs and outputs of several management functions performed by contractors, including: planning, estimating, cash flow forecasting, cost control and accounting. Baxendale (1990) has described the development of a MIS using data flow diagrams (DFD), to assist design-and-build organizations for production control. Paek and Lavata (1990) have also proposed a system integrating knowledge-based expert systems into the current information systems to enhance the quality and efficiency of the computerized systems.

Adrian (1987) identified that a major component of an integrated management system is a common database. Ideally, the MIS should utilize common database(s) and provide timely, relevant, accurate, complete, and formatted data for the different management functions. The main function of a common database is then to store raw data relevant to the integrated management functions. These data are collected in a reliable manner from ongoing and past projects and stored by suitable means in the records of a database. The efficiency of capturing, pre-processing, and managing the raw data determine the quality of information stored and used by the different applications (functions). This has motivated some researchers to propose site data capturing systems that ensure more complete and accurate data contained in the company database (McCaffer et al. 1990, and Rasdorf and Herbert 1988).

Although many cost estimating systems have been developed incorporating resource databases, current literature is limited in describing a structure of contractor databases for estimation purposes. In general, however, researchers have reported that databases containing historical productivity data, in terms of manhours needed to perform a certain work, are more practical than those containing unit costs (Adrian 1987, and Rayburn 1989). This is because historical productivity data are not as sensitive to change over time as unit cost data are. Currently in construction, several published cost data describe combination of crews, average productivity data, different construction activities, and work packages (e.g., Means 1990). These data could readily be utilized, as a database core, in developing the integrated bid preparation methodology.

2.3 Cost Estimation

The subject of cost estimation for bid preparation and job control has been of major interest to researchers and practitioners. During the 1980s decade, a large number of commercial software systems, have been developed to assist estimators organize their construction cost estimates. Arditi and Riad (1988) have outlined several of these systems. Equally large number of custom-built computerized systems have been proposed by researchers for estimation and/or control in a variety of formats: programs, spreadsheets, or data bases (e.g., Suckarieh 1984, Riggs 1988, Rayburn 1989, and Eldin 1989). These systems are based on two basic approaches used to organize work items for estimating. One approach is to identify work categories contained in the project's written specifications, such as those of the Construction Specifications Institute (CSI) for building construction projects, and are commonly utilized by commercial software systems (Moselhi et al. 1991d; Peurifoy and Oberlender 1989). The other approach uses a work breakdown structure (WBS - US Department of Energy 1986) to identify work items by their location on the project. The use of the WBS has been supported by researchers seeking integration among estimating, planning and scheduling, and control (Al-Tabtabai and Diekman 1990; Ho 1990; Keisk and Selby 1990; Riggs 1990; Adrian 1987; Meuller 1986; Neil 1982). The WBS technique has been introduced by the United States Department of Defence (DOD) in 1963 and applied widely as an integrated Cost and Schedule Control System Criteria (C/SCSC). The concept has been described clearly in the literature (US Department of Energy 1986). Nguyen (1990) has also provided an overview of the C/SCSC concept, typical cases impairing its implementation as well as possible approaches recommended for its compliance. However, despite its great advantages, the construction industry in general and the building sector in particular, has not yet embraced WBS and integrated project control, probably due to perceived high implementation cost and a lack of leadership in this area (Ho 1990). The survey performed be Tavakoli and Riachi (1990) shows that only 43.6% of the top 400 U.S. contractors use WBS. A few major contractors in construction, however, have recently developed different versions of integrated control systems using WBS as the integrated methodology, examples are the Synergy Integrated System (Bechtel 1989) and the Stone and Webster Integrated Management System (Badger 1987).

Despite the limited use of the WBS particularly in the building sector, it forms a sound basis for the integrated methodology, in addition to a proper design of a contractor code of accounts (HO 1990; Neil 1982). According to Neil, a well-designed code of accounts will satisfy the following requirements:

- Must catalog all needed information.
- Must facilitate consolidation of information at levels of detail from very broad to very detailed.

- Must allow for extraction, sorting, and summary of selected information from within the total.
- Must be compatible with accounting systems used in the company.
- Must be suitable for computerized processing.
- Must be usable by typical operating personnel.
- Must support both current and historical needs.

In the process of performing detailed cost estimates for bidding and control purposes, contractors need to come up with the direct costs attributed to the different work packages of the project. This is done by quantity take-off from complete drawings, specifications and subcontractors' quotations in consideration of the method of construction used. Contractor databases should provide data regarding crew combinations, productivity and pay rates to be used for direct cost calculations. Most of the estimating systems proposed for the industry (Arditi and Riad 1988) perform these calculations very efficiently. Also, some of these systems provide an integrated environment linking cost estimation to CAD and planning and scheduling software (e.g. Precision Estimating). However, these systems may not adopt the WBS concept and do not address many issues that need to be considered in compiling practical cost estimates, including:

- Systems do not provide decision aid regarding proper utilization of owned resources and the need for externally acquired ones. A practical system should facilitate estimation of the work load of owned equipment on ongoing projects. Proper resource rates could then be decided pertaining to a new project depending on expected use of owned versus external resources. The database should contain rates for external resources. This analyses also help in procuring the needed external resources.

- Systems that employ a WBS usually link the estimated direct costs of work packages to an appropriate reporting and responsibility level in the contractor organization, but, not commonly directly linked to the appropriate contract article for bid compilation purposes.
- Systems do not readily facilitate the transfer of data regarding work packages, direct costs, and resources to a planning and scheduling modules.
- Even if data is successfully transferred, usually no feed back cycle is implemented to modify the estimated costs based on the schedule impact. Realistic productivity factors and resource constraints need to considered in this feed back cycle to modify activity durations, and consequently the direct and indirect costs.

Indirect costs need also to be estimated. These costs include project overheads and a part of the firm general overhead. Contractors usually have a list of possible items and estimated costs are assigned to those applicable to the project. The total of the indirect costs cannot readily be attributed to a certain work package or a certain contract item. This issue is covered in the bid unbalancing section.

2.4 Planning and Scheduling

Planning and scheduling are processes highly relevant to realistic cost estimation. They establish estimates regarding the times of performing individual operations, total project duration, and use of resources. Thus, planning and scheduling reduce the uncertainty in direct cost estimation and also work as basis for practical indirect cost estimation. Construction planning involves the breakdown of the project into definable (time consuming), measurable and identifiable work tasks and then establish the logical interdependencies among them. Scheduling = planning + time (Moselhi 1989). Based on prevailing productivity levels and available resources, schedules are produced in conformity with the planning phase. Scheduling establishes activity durations, project completion time, critical activities, floats, etc.

In an integrated system, work activities have been identified by adopting the WBS concept and the use of the project work packages, as described earlier in the cost estimation module. However, project activities should be made to include not only cost consuming tasks (production tasks), but also time consuming tasks (e.g., concrete curing and procurement) as well. The system should facilitate the link among the WBS performed in the cost estimation module, the system databases, and a planning and scheduling algorithm. A proper planning and scheduling algorithm should also be made to incorporate both calendar and working days in order to consider for pre-specified management decision activities (vacations, delays, etc) or pre-specified time constraints. Complete and relevant information regarding the tasks and their interdependencies within an activity should be stored in the activities database.

Until recently, mainly two scheduling techniques have been used in construction; bar charts and network based schedules. Bar charts are simple and easy to understand by all levels of management and supervision. However, their major deficiency is that they do not show enough technological and organizational interdependencies among activities. Thus, bar charts are not likely to be used as a means to arrive at a schedule, but rather as a tool for output purposes of data calculated using a more comprehensive model of the construction process (Laramee 1983). Network techniques: Critical Path Method (CPM) and Precedence Diagram Method (PDM), introduced in the 1950s and 1960s respectively, have widely been used as a scheduling tool in construction. Both methods require careful study of the work activities and their resources, durations and logical relationships. Since these tasks are often performed in an unstructured fashion, network techniques required huge effort, cost and time to prepare. In addition, network techniques are based on the assumption of unlimited resources and utterly fail to model construction projects comprising repetitive activities such as high rise buildings (Arditi and Albulak 1979, Birrell 1980, Davis 1974 and Jaafari 1984). Commercially available software tools that adopt the prescribed bar chart and/or network techniques, require massive inputs and outputs, demand frequent updating, and fail to provide the required experiential knowledge for the planning and scheduling process (Passanisi 1985, and Kidd 1990).

In view of these deficiencies, two lines of research are currently being carried out in parallel, the first has focused on automating the generation of construction plans and schedules. This line of research has mainly utilized new and promising computer technologies derived from artificial intelligence (AI) research, such as Knowledge-based systems (KBS). These techniques have been reported as better tools for solving the industry ill-structured problems (Levitt 1987, Mohan 1990 and Kim 1990, Moselhi and Nicholas 1990, Nicholas 1989). According to many of the studies proposed in this line of research (e.g., Kartam and Levitt 1990, Zozaya-Gorostiza 1988), the absolute definition of a plan is a sequence of actions needed to transform the orientation of a group of objects from an initial status to a goal status. The computer planner determines which object to handle first by searching for an object which has its preconditions met, given the current orientation. Once a candidate object is found, an appropriate action is selected to be performed, based on the object type. After this action is performed, the effect of this action is a new orientation which might meet the preconditions of another object. The search mechanism continues to look for another objects to be selected and proper actions performed leading to continuous change in the orientation until the goal status is met. In real project notations, the objects are the physical components of a project (e.g., beam, column, etc.), the goal status is the complete project built according to the drawings and specifications, and actions could be "build-beam".

The object "beam" could have preconditions as a result of two general principles: (1) need for gravity support (the column below the beam); and (2) safety requirement (the deck below the beam). As such, the major components of KBS for automated planning are domain-specific facts and knowledge about the necessary actions and preconditions of each object, as well as an appropriate reasoning (search and ordering) mechanism. The several prototypes recently developed for the construction industry differ mainly in their assessment of these components. Major research efforts in this line of research include the work at Carnegie Mellon University (CMU) and Massachusetts Institute of Technology (MIT). The CMU research effort that focuses on the development of KBES for construction planning was reported by Hendrickson et al (1987). Their prototype, CONSTRUCTION PLANEX is initially being developed to generate project activity networks, cost estimates, and schedules of modular high rise buildings. The system uses the MASTERFORMAT codes to aggregate detailed activity elements into project activities (bottom-up) and produces plans using lookup of successor data that are provided in advance rather than deduced. At MIT, another KBES prototype related to construction planning is reported by Navinchandra et al (1988). Their system, GHOST, is part of a larger integrated environment called CONPLAN. GHOST requires list of activities as input and produces as output the precedence relationships among activities and a project network diagram. GHOST planning technique is based on domain-specific critics. for example, a critic called "supported-by" will lock at various activities that might be executed in parallel and determine if an object is supported by other object, and, if that is the case, it will assert that constructing the support comes before the supported activity. Following these major efforts, a large number of knowledge-based systems have been suggested for planning and scheduling including: Kartam and Levitt (1990), Echeverry et al. (1989), Benjamin et al. (1990), Kano (1990), and Stephenson (1990). Moselhi and Nicholas (1990) have also developed a hybrid expert system that integrates an expert system building tool (ESBT), a knowledge base of activities and their relationships in rules, databases, a planning and scheduling software, and external programs in Fortran. Their expert system "ESCHEDULER" produces an "As planned" schedule and modifies it according to several factors including weather, overtime and congestion of trades. After these modifications, ESCHEDULER produces an "As possible" schedule. Although the system does not address cost estimation issues, the modification cycle is of interest to this study.

Several of knowledge-based systems, as those discussed in the last paragraph, are reported by many researchers as only non-operational prototypes (Mohan 1990, Brandon 1990, Fung Fai 1989, and Adeli 1988). This line of research faces development problems with respect to knowledge acquisition and representation. Several of the operational prototypes (e.g. GHOST and PLANEX) require special hardware and take long computer time to produce a plan, despite the limited number of objects included (as opposed to the large number expected to be involved in a high rise construction). These problems have resulted from the inadequacy of conventional AI architectures that adopt only reasoning aspects to provide a precise model of the human intelligence (Gallant 1988 and Castelaz et al. 1987). Most of the systems have not addressed the practicality of those automated planners from a contractual point of view. For example, most of the those planners obtain their object descriptions automatically from CAD files used in the design stage, thus, these systems could be restricted to the design-build environment. Continuous work under this line of research integrating several AI-based techniques could produce more practical systems in the near future.

Parallel to the research in AI planning, the second line of research has embarked on the traditional path of using conventional algorithmic tools and procedures. Researchers have focused on the development of more suitable planning and scheduling models of projects comprising both repetitive and non-repetitive activities (e.g., high-rise buildings) as well as more integrated efforts to improve the schedule practicality and development productivity. The approach that has been adopted is to integrate usual network techniques suited for projects comprising non-repetitive activities, and the Line of Balance technique (LOB) used mainly for planning, scheduling and control of linear projects comprising sequential and repetitive activities (e.g., pipelines, highways).

The LOB method was originally developed by the U.S. Navy Department in 1951 for programming and control of repetitive projects. Although variation of the original method have been proposed, all are practically similar and provide compatible graphical presentations. These variations include "Linear Scheduling Method" by Johnston (1981), "Time Space Method" by Stradal and Cacha (1982), "Line of Balance" by Lumsden (1968), and "Time Chanage Charts" by Mawdesley et al. (1989). The LOB, however, faded into complete obscurity in the early 1960's with the development of the network analyses techniques. In 1966 the National Building Agency was able to combine the advantages offered by networks and the principles of the original (LOB) method, hence retaining the power of networks and making full use of the advantages offered by the LOB. The method was used for planning and control of purely repetitive projects.

In all the graphical approaches, repetitive activities are plotted as lines with constant or varying slopes (where slope represents the production rate), the axes being units versus time (Fig. 2.1). The advantages of these charts are: its simple presentation and clarity, and its consideration of work continuity and the learning phenomena. Irrespective of these advantages, the sole use of the LOB has several limitations (Laramee 1983): (1) does not indicate the schedule of work of a particular trade (or activity) from unit to unit; (2) cannot allow for the presence of non-repetitive activities; and (3) does not show the relation among activities (e.g., critical and non-critical activities). Recent surveys in the North American market indicate that the LOB method is not commonly used and many practitioners in construction are not aware of this method (Mansur 1990 and Riggs 1990).

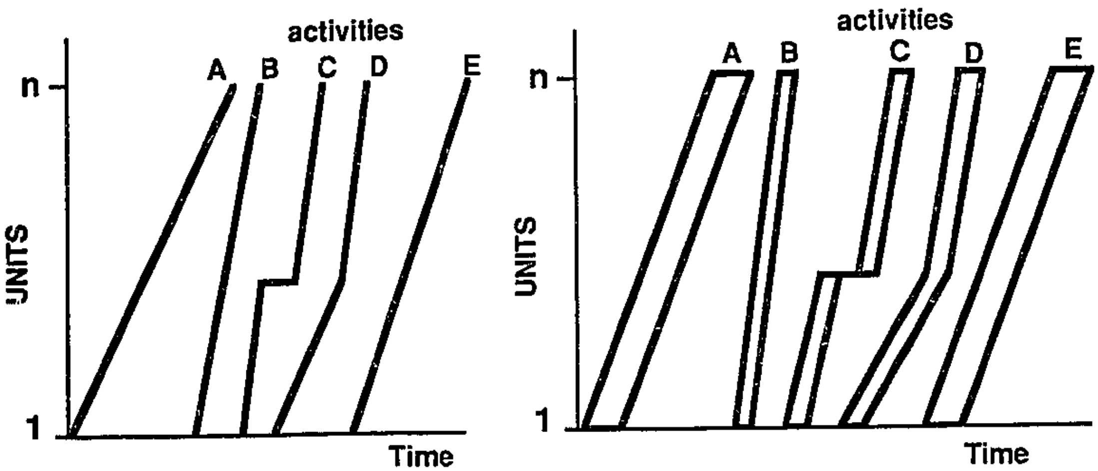


Fig. 2.1: A typical line of balance chart.

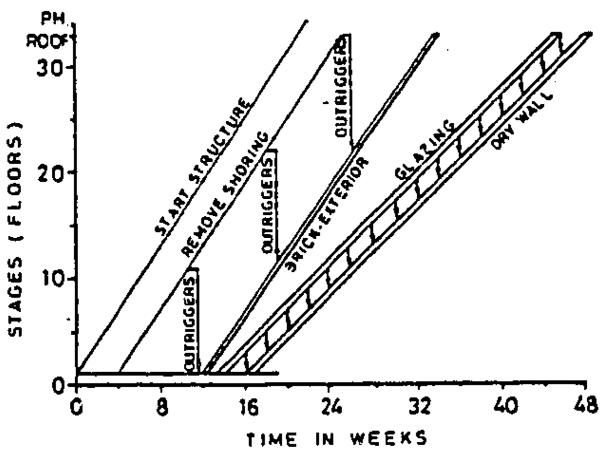


Fig. 2.2: Graphical plan for a high-rise building critical activities.

(Ref. O'Brien 1975)

Since the research in hand focuses on the design of a practical and generic estimating system, the research is directed toward the development of a more practical planning and scheduling model of high-rise buildings. Deep refinement of the planning and scheduling model using AI-based methods could be the focus of future work. In the literature, three major studies have attempted to develop such a model by integrating the LOB method and network technique (O'Brien 1975, Laramee 1983, and Russell 1990).

O'Brien (1975) used a network to model non-repetitive activities which are performed from the start of the project until the first repetitive floor (i.e. site work, excavation, substructure and basement). Another network is then constructed to represent a typical floor of the high-rise building. When this network was analyzed, critical activities (which must be performed sequentially) were identified. The procedure then uses a LOB chart to show the coordination between the "critical" trades along the repetitive floors. The final chart showed only 7 activities out of 24 (see Fig. 2.2) and did not show the non-critical activities. However, the study suggested that a repetitive activity is possible to have other kind of dependencies than just the preceding repetitive activity. The LOB chart could show such relationships graphically, however, no general mathematical formulations were included in the model proposed. Also activities aggregation using the LOB charts was not addressed. Therefore the model of O'Brien was not explicit enough in dealing with these limitations.

At the Centre for Building Studies, Concordia University, Laramee (1983) examined the construction process of two high-rise buildings constructed at down town Montreal in order to formulate the specifications for the planning and scheduling model that he developed. The study identified that repetitive and non-repetitive activities are often dependent one to the other and that the scheduling model should allow precedence relationships between the two types of activities. Unlike O'Brien's, the model developed by Laramee is a merge algorithm between the network technique and the LOB method. Laramee's algorithm has other interesting features including: allow for continuity requirement and learning phenomena in the mathematical relationships, allow dependency relationships between repetitive and non-repetitive activities, and uses calendar days and adjusts the schedule based on "weather factors" that affect the productivity in the different months of the year. However, despite these advantages and the well-structure of the algorithm, it has some limitations including: (1) it does not allow activity relations other than finish to start; (2) does not permit activity splitting in order to consider for space or technological constrains; (3) does not show the schedule of work of a particular trade or activity from unit to unit; (4) does not adopt a WBS and therefore the problem of activities aggregation is not addressed; (5) does not investigate the methods and processes used at the detailed activity level; and (6) does not improve the presentation of the LOB charts to include non-critical activities.

Russell (1990) has developed a highly structured planning and scheduling system for projects which contained significant repetition of work. The system is called REPCON and is currently being commercialized in the United States and Canada. It has been developed as part of a research program directed at identifying and satisfying the needs of medium-sized general building and civil engineering contractors (a construction volume of $10 to $100 million per year). The system addresses activities and logic in a generalized sense and uses an object oriented methodology to link an activity either globally to another activity or locally to a certain entity of another activity. Although the REPCON system is conceptually sound and solves many of the limitations that faces Laramee's algorithm, it optimizes only the time, without incorporating cost, in schedule development and does not improve the representation of repetitive activities. In addition, the non-transparency of the code disables modifications to be made on it as required by the system in hand.

The development of the practical estimating system in hand has to address the above mentioned limitations. As the WBS is the proposed technique of identifying the project activities and their levels of breakdown, the problem of activity aggregation and detailing comes to the surface. One way of tackling this problem has been proposed by Kano (1990). Kano has developed a planning and scheduling prototype based on the Down-from-the-Top methodology of schedule development. The start and finish dates calculated in the upper level become constraints to the schedule of the lower level. Further analyses of the lower level may also mandate updating to the upper level. This updating is done in an Up-from-the-Bottom method. In this case, the schedule of the lower level activities are the elements which compose the upper level schedule. The schedule is thus refined by back and forth directions (cycles) in the breakdown structure. This methodology that Kano presented has to be adapted to the estimation system in hand, in view of the different types of interdependence relationships introduced between repetitive and non-repetitive activities included in the different work packages. This is beside presentation enhancements such as those proposed by Mansur (1990), see Fig. 2.3.

Once an initial schedule is successfully produced, activities are assigned planned start and finish times, and critical activities are determined. Cost implications of such a schedule should be addressed. Also, practical productivity factors should be assigned to the activities to account for expected weather conditions at the scheduled times, trade congestion, over time, and other productivity-related factors. Work load of owned resources should also be considered to adjust the estimated costs of resources. Based on these considerations, a cycle of schedule updates may become necessary and resources may have to be reallocated, impacting the project costs and the schedule. Accordingly, reports pertaining to the project activities, direct costs, schedule, and resource use can be generated at the different management levels. Some of these reports, particularly those pertaining to accumulated resource use, provide a guide for indirect cost estimation.

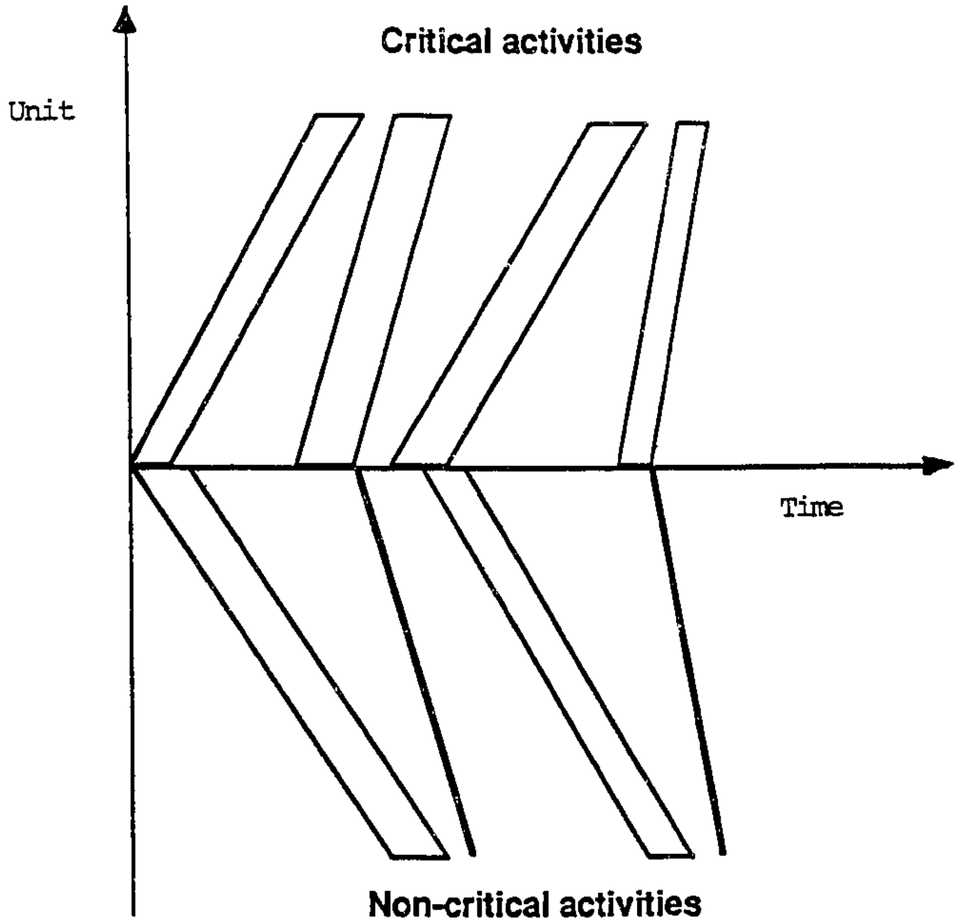


Fig. 2.3: Representation enhancement to the LOB Chart.

(Ref. Mansur 1990)

2.5 Risk Assessment and Markup Estimation

The three subsections regarding MIS and databases, cost estimation, and planning and scheduling have discussed the functions and procedures used to consider the quantitative aspects of a bid price estimation. Added to all these estimated costs, contractors should decide on a markup value as a percent of these costs, to cover mainly their profit. Markup decisions are crucial to the success of the business organization and its long-term performance since they determine the contractor's probability of winning the job in addition to his potential profit. These decisions require consideration of many qualitative factors that represent the risk environment which the contractor has to practically assess, if low enough and high enough markups are to be estimated.

Research efforts described in the literature to estimate optimum markup can be generally classified, as shown in Fig. 2.4, into three categories: 1) models based on probability theory; 2) models based on decision analysis techniques (analytical hierarchy process and multi-attribute utility theory); and 3) models based on artificial intelligence (AI) and knowledge based expert systems (KBES). The first category of models were developed as early as the mid 1950s when Friedman (1956) developed the first competitive-bidding-strategy model. Friedman's model was further elaborated by a series of researchers (e.g., Rosenshine 1972; Ioannou 1988). A parallel family of probability-based models is based on the concepts set forth by Gates (1967) and subsequently elaborated by others (Rosenshine 1972;

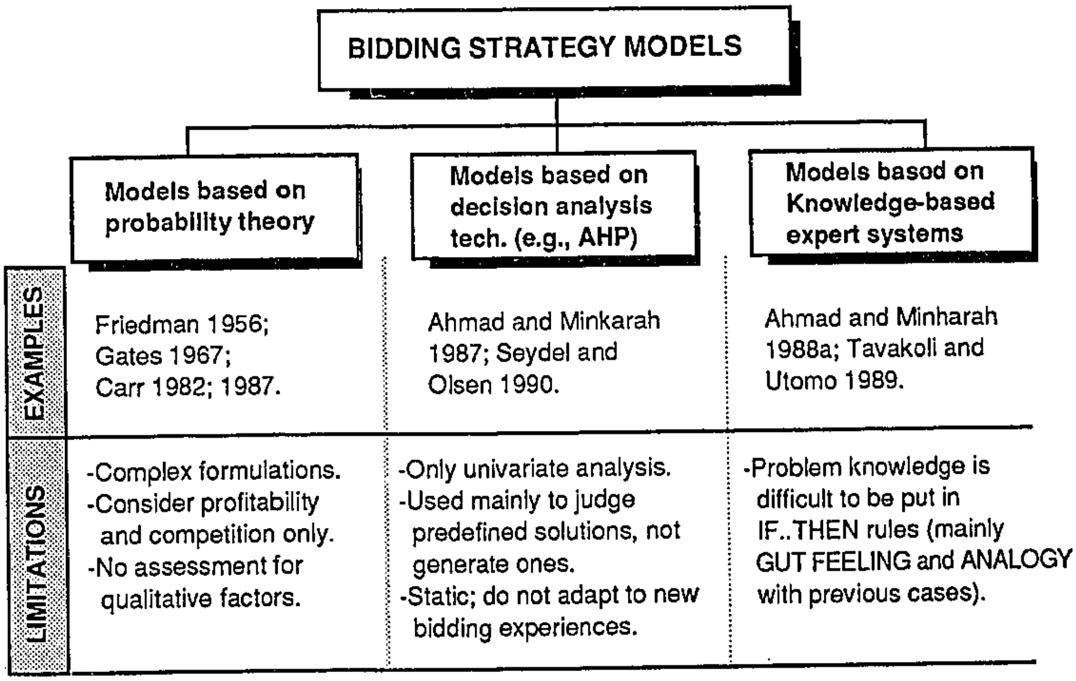


Fig. 2.4: Bidding Strategy Models.

Dixie 1974). The latter family of models differs from the first in that it uses a different way to calculate the probability of winning over a group of competitors (see Benjamin and Meador 1979). A third variant was proposed by Carr (1982, 1987). His differs from the preceding two in that it treats cost, rather than profit, as the random variable. In general, however, the three families of models have commonalities in structure and characteristics (de Neufville and King 1991). They are based solely on the probability theory, considering only two factors: profitability and expected competition. These models are, basically, methodologies designed to optimize one decision variable of the bidding strategy problem, namely, expected profit (Ibbs and Crandall 1982). The expected profit is, for a given bid amount, the product of the profit that would be realized from the bid and the probability of winning the job at that bid. These models utilize the previous performance of other competitors (mean and standard deviation of the distribution of a competitor-bid to contractor-cost ratio). Often, models of this category exhibit complex mathematical formulations despite the simplifying assumptions made for modelling convenience.

Despite the differences in assumptions and basic formulations among probability-based models, they generally provide answers to three questions 1) what is the probability of winning at a desired markup?; 2) what is the optimum markup value?; and 3) what is the probability of winning at optimum markup?. However, the answers these models provide to these questions are inherently different. This has triggered a growing debate among researchers and consequently contributed to a lack of confidence and enthusiasm in the use of bidding models among contractors. Recent developments in bidding strategies have also indicated the inadequacy of the sole use of probabilistic methods since they do not account for a number of important factors such as contractor keenness for the job, prevailing market conditions and owner attitude, that govern the determination of optimum markup in current practice (Runeson 1990; Flanagan and Norman 1985). Those researchers, among others, have viewed probability-based models as fundamentally weak since optimum markup decision depends more on the state of economy and other subjectively evaluated factors, than on the relative margins between bids on previous encounters. This argument stems from the observation that some contractors, in a highly competitive environment, bid below their estimated costs to win a job. Hence, such models provide no help to contractors in such a situation. The results of two recent surveys (Ahmad and Minkarah 1988 b, and Sey and Dikbas 1990) seem to support this argument. Ahmad and Minkarah (1988 b) have conducted a questionnaire survey among the top 400 general contractors in the United States to identify the factors that characterize the bidding decision-making process. Two sets of factors, with different rank orders, were identified; one affects bid/no-bid decisions and the other affects percent-markup decisions. Table 2.1 lists the 10 top-ranked factors, out of an exhaustive 31-item list included in the survey, affecting percent-markup decisions. It was noted that **competition** and **profitability**, which are the only two factors considered in the formulation of probability-based models, were not among the top-ranked factors. In the second survey (Sey and Dikbas 1990), the most decisive factors were ranked in the following order: the workload of the firm, desirability of the job, top management pressures on estimators, and the **job risk.** Although the survey was conducted in another country, the findings are in good agreement with those of the first survey despite the country of origin.

Table 2.1: Factors affecting percent-markup decisions

RANK	FACTOR
1	Degree of hazard
2	Degree of difficulty
3	Type of job
4	Uncertainty in estimate
5	Historical profit
6	Current work load
7	Risk of investment
8	Rate of return
9	Owner
10	Location

Ref.: Ahmad and Minkarah (1988 b).

Despite the limited use of bidding models by contractors, the argument of their total inapplicability is not true since probabilistic analyses to previous encounters could disclose the essence of those unquantifiable factors implicitly included in the optimum markup value that a contractor decides. Comparative studies have been conducted among the modes' results to extract useful correlations (Benjamin and Meador 1979; Tarranza and Carmichael 1986). Interesting results have been found out by a recent work (Moselhi and Hegazy 1990 a,b), conducting a parametric study and sensitivity analysis among the three commonly used models of Friedman, Gates, and Carr. The purpose was to extract the essence of existing probabilistic models through identification of general trends that could then be utilized in a simple manner to estimate optimum markup. A computer program (**BID**) was developed (see Fig. 2.5) and used to generate optimum markup trends over a wide range of statistical parameters that characterize bid situations (mean and standard deviation of bid/cost ratio and number of typical competitors). The study indicated that Carr's direct solution model produces results that are between those of Friedman and Gates, for the most practical range of a typical competitor behaviour. In this sense, Carr's direct solution could be viewed as a most likely estimator between Gates optimistic and Friedman pessimistic models. It was also shown that the main factor in the determination of an initial optimum markup (based on the competition) is the mean of the bid/cost ratio. Simple charts having a **"Hockey Stick"** shape were developed for the direct determination of optimum markup. Although the study did not explicitly address how the qualitative factors could be incorporated into the optimum markup decision, it provides insights on the impact of changes in the competition on the optimum markup. This greatly helps contractors to make last minute decisions at the negotiation table.

Several research work was also pursued trying to modify probability-based models to account for the qualitative factors described earlier. Some of these efforts have

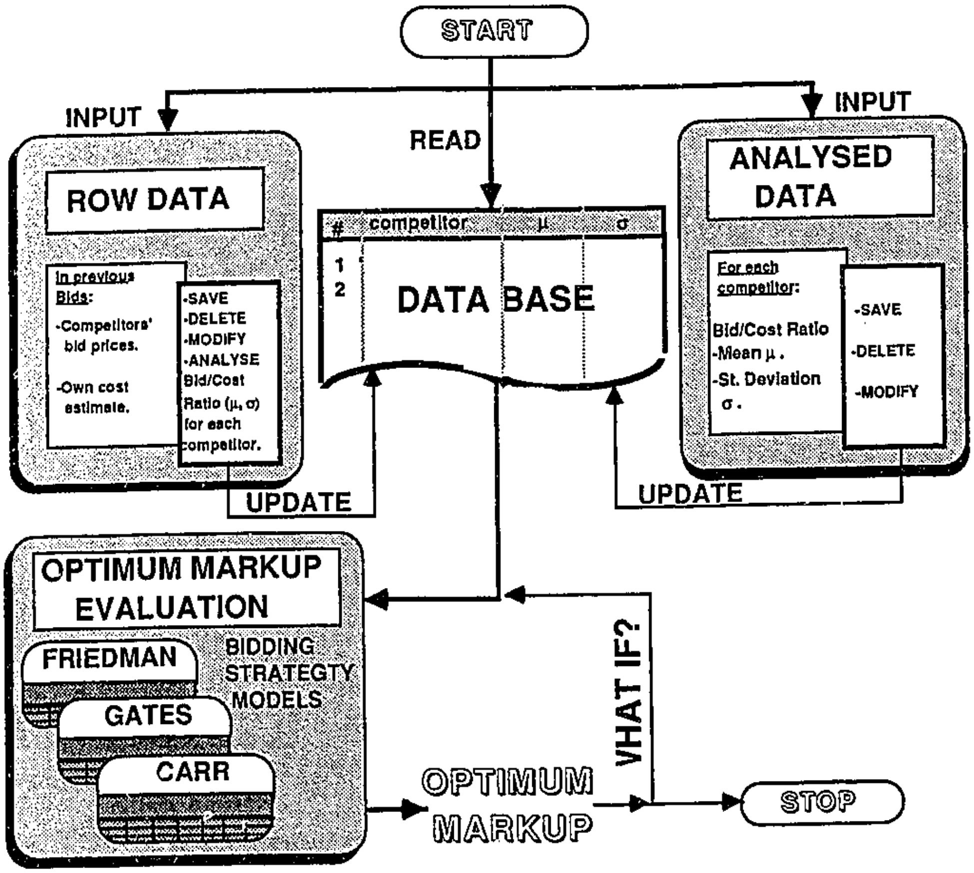


Fig. 2.5: Computer program (BID) schematic diagram.
(Ref. Moselhi and Hegazy 1990b)

not been successful since they included some of the qualitative factors into the model formulation, and thus, added more complexity to such models. Some of those non-successful efforts include the early works of Goodman and Burmeister (1976), Knode and Swanson (1978), and Grinyer and Whittaker (1973) where factors such as contractor's managerial judgement and limited capacity were introduced into the models. A recent work (Griffis 1992) also has attempted to fine tune the probability of beating a single competitor by considering his expected work load. This effort, however, have resulted in more complicated formulations that is expected merely to fine tune a local optimum, rather than a global one (Moselhi and Hegazy 1993b).

In order to consider for the multi-attributed and subjective nature of bidding decisions, analytical decision analysis tools have been used (second category of markup models-see Fig. 2.4). Multi-attribute utility theory was utilized by Ahmad and Minkarah (1987) in developing an additive utility model to estimate markup. The analytical hierarchy process (AHP) was also utilized by Seydel and Olson (1990) in developing their markup estimation model. Both the utility theory and the AHP are techniques that are able to consider multiple criteria and incorporate the contractor's preference structure into the model formulation. These techniques are normally used to evaluate a set of pre-defined decision alternatives (e.g., bid or no bid), in terms of their relative fulfilment of a set of pre-defined decision objectives. A representative example is the deterministic model developed by Ahmad (1990) for the bid/no-bid problem. Interesting findings have also been presented by de Neufville and King (1991), establishing utility functions for evaluating the effect of risks and need for work on contractors premiums. Although their empirical study was not intended as a markup estimation model, it demonstrated that contractors systematically add significant premiums to their bids, on the order of 3%, when their need for work is low or when the risks associated with the bid projects are high.

Despite the ability of these techniques to structure the decision-making process, they exhibit several limitations when applied to the percent markup estimation problem, including (Moselhi et al. 1991b):

- Markup estimation problem is one of generating a finite set of feasible alternatives under uncertainty, rather than of evaluating pre-defined decision alternatives with respect to given objectives. However, the latter could be applied to further optimize the decision and arrive at a most suitable alternative of those generated.
- Markup is normally a continuous rather than a discrete variable, which enlarges the problem solution space.
- The techniques can only be used for univariate analysis. Thus, based on the qualitative factors, a markup value could solely be determined to presumably enable the contractor to win the contract. However, the implications of those factors, and the decided markup, on other issues that

are important for proper bidding decisions (e.g., actual project profitability, project potential for claims and change orders) cannot be jointly evaluated.

- The techniques are static; not able to adapt their performance based on new bidding experiences.

To overcome some of these limitations, Seydel and Olson have assumed that there exists a finite set of discrete markup ratios that can be identified and correspondingly evaluated with respect to a set of objectives, including: profitability, risk of monetary loss after winning, and work continuity. The model developed by Ahmad and Minkarah, on the other hand, generates optimum markup by maximizing an expected utility function using an optimization routine. The expected utility function is an additive form, combining three utility functions for loss, general overhead, and profit. It also incorporates probabilistic parameters representing competition, and uncertainty. The model is complicated and, similar to probability-based models, requires analysis of large historical data. Also, both this model and the AHP-based model of Seydel and Olson fall short in considering many of the qualitative factors described earlier, in addition to their dependency on a contractor's particular experience not the domain general knowledge, making them not able to provide experienced guidance that may help contractors modify their own practices.

As opposed to the algorithmic and mathematically complicated nature of traditional decision making tools, research in artificial intelligence (AI) have been concerned with encoding the domain experts' knowledge and the problem solving process into more "intelligent" and automated computerised systems. KBES have been the fastest product of AI to come to a wide use due to their ability to capture the experts' rules of thumb and experience and to model ill-structured problems, permitting reasoning and explanation facilities. Construction engineering and management has been considered a fertile field for many expert system applications (Mohan 1990). This is due to the fact that expert knowledge, judgement and experience are key requirements for the resolution of construction problems as well as for performing most of construction engineering and management tasks (Levitt 1987).

KBES applications have been proposed for the markup estimation problem (e.g., Ahmad and Minkarah 1988a and Tavakoli and Utomo 1989). These models, however, utilize the expert system environment mainly to provide a user friendly interface, explanation capabilities, and automate the calculations involved. In general, however, expert system applications for several construction problems including markup, may prove inadequate. A number of expert systems in the construction domain have recently been evaluated (e.g., Adeli 1988, Allwood 1989, Bowen and Erwin 1990, Brandon 1990, Fung Fai 1990, Minkarah and Ahmad 1989, among others) to reveal that only a few of these systems perform intelligently as originally claimed. Knowledge acquisition has been reported as the major bottleneck in the development of expert systems (Bowen and Erwin 1990, and Gallant 1988). This is due to the fact that human domain experts have significant implicit knowledge which they cannot articulate no matter how hard they try. This implicit knowledge can amount to 100% if an expert is confronted with routine problems, such as the markup problem (Musen and Van Der Lei 1988). Brandon 1990 and Fung Fai 1990 have also reported that knowledge structuring and domain limitation represent major problems in conventional expert system architectures (serial knowledge-based architectures that lack deeper representations of the domain in terms of spatial, causal or temporal models). In spite of the initial stereotyped expectations regarding expert systems intelligence, their reasoning capabilities have been reported to be only a fraction of what is needed to achieve intelligence (Gallant 1988 and Castelaz et al 1987). Expert systems lack the ability to learn by themselves, generalize solutions, and adequately respond to highly correlated, noisy, incomplete or previously unseen data (Wassermann 1989, and Pao 1989). Moreover, the serial architecture of conventional expert systems adds severe restrictions on their practicality because it makes their complex applications considerably slow and costly to process as opposed to the new generation of parallel processing architectures (Gallant 1988 and Castelaz et al 1987). Expert systems applications further require intensive software development and maintenance beside large allocated memory. These problems collectively, limit conventional expert systems to areas where a substantial body of knowledge connecting situations to actions exists, providing that sufficient amount of time is allowed for reasoning (Jackson 1986), which is not the case in markup estimation.

The ability of construction experts to produce reliable decisions based on partial cues, as in the process of estimating optimum markup, has two main implications. First, construction experts in reality do not reason to come up with a decision, but rather use these rules to try to explain why they arrived at those decisions (Gallant 1988). Therefore, modelling a certain situation in the form of IF..THEN rules as those used in expert systems is not sufficient to provide performance similar to the human expert since any lack of required data will stop the reasoning session. Second, situations that consist of a large number of interrelated attributes that have to be considered in parallel are very difficult to model since construction experts might fail to explain why or how they arrive at decisions. These implications confirm that the principal act in solving a number of construction problems is of pattern recognition rather than deep reasoning about the problem elements.

Therefore, the limitations described confirm the unsuitability of using KBES as a tool for modelling the markup estimation problem. The markup problem lends itself more to an analogy-based solution which expert systems are ill-equipped to model. The problem usually consists of a large number of attributes (qualitative factors) that have to be considered in parallel and in a holistic manner (as a pattern) rather

Table 2.2: Criteria for a potential Neural Network application.

Criteria
- Conventional algorithmic tools are inadequate.
- Problem requires qualitative or complex quantitative reasoning.
- Problem is routine and knowledge is mainly implicit and can not be modelled in the form of IF..THEN rules.
- Solution is derived from a large number of highly interdependent parameters that have no precise quantification.
- Solution is derived quickly, based on "GUT FEELING" and no step-wise logic or computations are involved.
- Problem area is rich in historical examples but data set is incomplete, contains errors, and describes specific examples.
- Development time is short, but sufficient training time is available.

than a sequential and segmented manner, often with no computations or deep reasoning involved. Also, the knowledge needed for the problem resolution is mainly implicit and cannot be adequately modelled in the form of IF..THEN rules as utilized in most expert systems. Recently, another research in AI, Neural Networks (NN), has provided powerful tools, performing complicated pattern recognition tasks and thus could provide a more suitable and practical solution to the markup estimation problem, in consistency with the criteria outlined in Table 2.2 (Moselhi et al. 1992a). Since the problem in hand meets these criteria for a potential NN application, neural networks literature is studied (section 2.7) to investigate their potential benefits in modelling the markup estimation problem.

2.6 Bid Unbalancing

Generally, bids are compiled by contractors and submitted to the owner in the form of unit price(s) assigned to the contract item(s). This is done by a procedure of either balancing or unbalancing (front-end loading) the distribution of the indirect costs and the markup, in addition to the direct costs, among the contract items. In the case of a balanced bid, the indirect costs and the markup are distributed among the contract items with shares relative to their direct costs in relation to the total direct costs of the project. On the other hand, a bid is said to be "unbalanced" if the unit prices for some items are higher than they "should be" and prices for other items are lower, without changing the total bid price. Some owners and engineers, thus, may perceive bid unbalancing as an immoral and unethical procedure by which a contractor improperly enriches himself at the expense of the owner. This view, however, is simplistic and incorrect. Researchers and practitioners have regarded bid unbalancing as a form of risk assessment measure, a cash flow management procedure, and a procedure to produce competitive bids and reduce the owner's cost of construction (Nadel 1991; Copare 1990; Stark and Mayer 1983). The contractor typically believes that the owner or engineer has created the need for the unbalancing by the way unit price contracts have been structured, providing no means of separately paying the contractor for indirect costs (Nadel 1991). For example, the contractor is compelled to front-end load his bid in the absence of a realistic mobilization payment provision. Similarly, the contractor is compelled to unbalance when unreasonably high or low quantities are established for some items.

Two motives drive contractors to unbalance bids. One is to reduce the financial cost of doing the work by getting paid as early as possible. If they can reduce financing costs, they can pass the savings along to the owner in a lower bid. The lower bid increases the likelihood of getting the contract. The second motive is to maximize the actually realized contract income without raising the nominal bid price. Despite the potential benefits of bid unbalancing, it has some perceivable risks, including:

- Presenting unrealistic unit prices may cause the owner to doubt the accuracy and the competency of the contractor's bid.
- A unit price that is unrealistically lowered may be used by the owner to develop unit costs for change orders.
- Early positive cash flow may give the contractor a false sense of wealth.

Stark and Mayer 1983 presented a linear programming routine to optimize bid unbalancing with an objective function to maximize the contractor's present worth of his expected revenues. The procedure can be described as follows:

Consider a unit price proposal containing N items with an estimated project duration of T months, for which, a competitive total bid B (total of directs, indirects, and profit) has been determined. Unit prices are to be determined so that cash flow (the early income flow) is maximized while retaining the integrity of the tender. From the project schedule, let $q(n,t)$ be the quantity of the nth item ($n = 1,2,...,N$) to be completed in the tth month ($t = 1,2,...,T$). Eventually, the sum of $q(n,t)$ over t for each item, that is:

$$q(n,1) + q(n,2) + ... + q(n,T) \qquad [2.1]$$

should equal Q_n, the contractor's quantity estimate for that item (this total may or may not be equal to the quantity estimate Q_n in the proposal). Also, define a monthly discount factor v^t, evaluated as $(1 + r)^{-t}$ at month t, where r is the monthly interest rate. X_n is also defined as the unit price for the nth item and a retainage rate of $(1 - a) * 100\%$ where, a is the fractional payment rate. Then, the objective function for maximizing the present worth of the expected revenues can be written as:

maximize:

$$\begin{aligned}
Z = & \{a v^1 q(1,1) + a v^2 q(1,2) + ... + a v^T q(1,T) + (1 - a) v^T Q_1\} \,.\, X_1 \\
& + \{a v^1 q(2,1) + a v^2 q(2,2) + ... + a v^T q(1,T) + (1 - a) v^T Q_2\} \,.\, X_2 \\
& + ... \\
& + \{a v^1 q(N,1) + a v^2 q(N,2) + ... + a v^T q(N,T) + (1 - a) v^T Q_N\} \,.\, X_N
\end{aligned} \qquad [2.2]$$

In this equation, the retainage is assumed to be paid upon completion of the job (at the end of period T). The optimization constraints are as follows:

a) bid constraint: the unit prices when multiplied by the proposed item quantities must sum to the bid total B, that is:

$$Q_1 X_1 + Q_2 X_2 + \dots + Q_N X_N = B \qquad [2.3]$$

b) Bound constraints: lower and upper bounds for the unit prices could be specified. If L_n and U_n denote respective lower and upper bounds on the unit price for the nth item, then bound constraints are:

$$\begin{aligned} X_n &\geq L_n \\ X_n &\leq U_n, \qquad n = 1,2,\dots,N \end{aligned} \qquad [2.4]$$

c) Formality constraints: these constraints could be introduced to force the unit price of one item to be higher or lower than another item. This ensure that the bid does not seem to have calculation errors. For example, a unit price for earth excavation has the appearance of error if it exceeds the unit price for rock excavation since the latter is more costly. In such situation, a formality constraint can be written as:

$$X_e - X_r \leq 0 \qquad [2.5]$$

where, X_e and X_r represent the respective unit prices for earth and rock excavation.

d) Non-negativity condition: to restrict the results to positive values:

$$X_n \geq 0, \qquad n = 1,2,\dots,N \qquad [2.6]$$

e) Timing constraint: such constraint restricts the total revenue received during the first t months of a project to a sum R, which may equal a limit on taxable income or a percentage of the total revenue anticipated. The constraint can be written as:

$$\{a\,q(1,1) + a\,q(1,2) + \ldots + a\,q(1,t)\} \,.\, X_1$$
$$+ \{a\,q(2,1) + a\,q(2,2) + \ldots + a\,q(1,t)\} \,.\, X_2$$
$$+ \ldots$$
$$+ \{a\,q(N,1) + a\,q(N,2) + \ldots + a\,q(N,t)\} \,.\, X_N \leq R \qquad [2.7]$$

The linear programming procedure described is simple and arrives at the global maximum to the objective function. Stark and Mayer demonstrated the working of the procedure on some example bid unbalancing situations. It should be noted, however, that the objective function and the constraints relate only to the contractor revenues and have no relation to the contractor estimated costs and cash flow constraints. For practical applications of the procedure, particularly to the integrated system being proposed, additional constraints need to be added to constrain the difference between the estimated costs and anticipated revenues, at any time period, to the monetary value the contractor is able to finance. Such constraints may prove beneficial and better manage the contractor's cash flow.

2.7 Neural Networks as a Tool for Markup Estimation

Expert systems, robotics and neural networks are among current Artificial Intelligence (AI) research areas of interest to the construction industry. On one hand, expert systems attempt to model the intelligent reasoning and the problem-solving capabilities of the human brain. On the other hand, neural networks attempt to model the brain learning, thinking, storage and retrieval of information, and associative recognition. Since the early 1950's and despite their parallel start, research in neural networks has grown much slower than that of expert systems (Fig. 2.6) due to initial scepticism (Minsky and Papert 1969). By the mid 1980's, neural networks research started to flourish due to several reasons, including (Wassermann 1989; Pao 1989): 1) evolution of new generation of powerful computers since early 1980s; 2) evolution of neural networks paradigms that lack the limitations of earlier models; and 3) transition of research from theory to practical applications.

A limited introductory application of the neural network technique has been proposed for the construction industry (Flood 1989 and 1990). However, the problem Flood presented lends itself more to manufacturing rather than to construction. Recently (Moselhi et al. 1991a,e; 1992a) have identified that significant part of inherent construction industry problems lend themselves to a form of analogy-based problems. Some typical examples were given including: every day decisions on a construction site, last minute bid estimating, and design under pressures for speedy resolutions. Decisions regarding these problems are made, mainly, on basis of analogies from past experience, rather than upon detailed analysis of the elements of a given situation. The study identified the characteristics of neural networks with respect to possible implementation in construction, and several potential applications were outlined. An example network

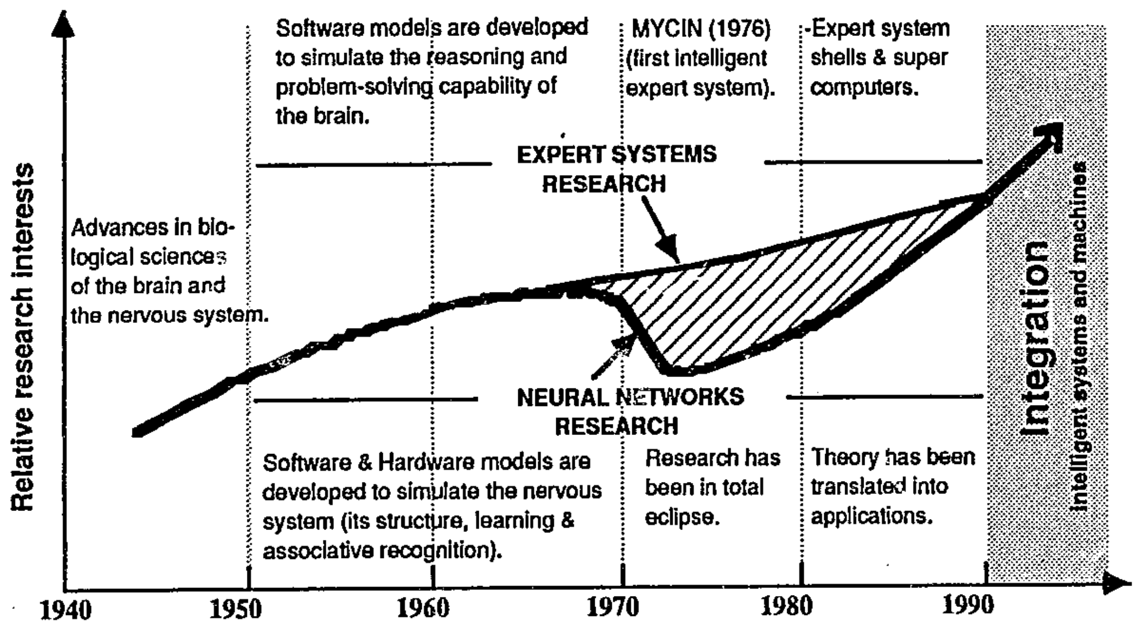


Fig. 2.6: Evolution of research in neural networks and expert systems.

was developed to demonstrate the potential benefits of neural networks and illustrate the simple and efficient modelling of construction problems.

Neural networks generalize solutions to problems by learning from pairs of input pattern(s) and their associated output pattern(s) (a pair of input and output patterns is called a training example). After training, a network is able to provide a speedy response to an input pattern, satisfactorily close to the desired set of outputs used in training. In addition to neural networks' general characteristics of parallel distributed processing and fault tolerance (Caudill and Butler 1990), the benefits of applying neural networks to the percent markup problem can be summarized as follows:

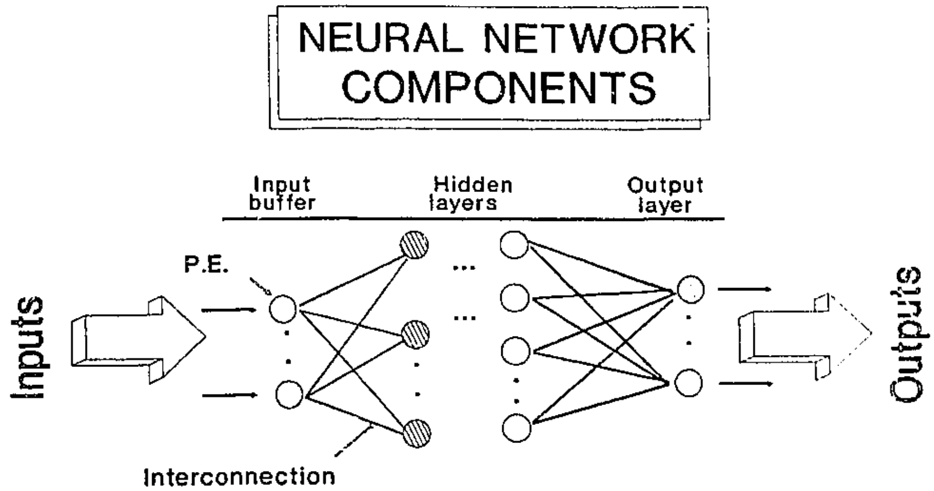
- Neural networks learn from examples and simulate, **by themselves**, the process by which the inputs (problem attributes) produce the output (decision or conclusion).
- The training examples could be elicited from experts or collected from historical data, with limited knowledge acquisition effort. Questions of "WHY" and "HOW", which are often difficult to answer, are not required.
- Neural network output could be multi-attributed. In the markup problem, a neural network is required to estimate a percent markup that maximizes the probability of winning the bid, and also predicts the expected profit at such markup.
- Model development is relatively simple, and requires less time, effort, and

software maintenance, in addition to satisfactory accuracy and high speed in producing the result.

- Neural networks have a great potential to be integrated with other algorithmic and KBES tools, providing reasoning and accurate computational capabilities to enhance the NNs analogy-based models (Moselhi et al. 1991 a, Bailey and Thompson 1990b).

2.7.1 Components and characteristics

Neural networks are defined as a type of information processing system whose architecture is inspired by the structure of human biological neural systems (Caudill and Butler 1990). The human brain is made up of billions of cells called neurons. Each of these cells is like a tiny computer with extremely limited capacity - yet, connected together, these cells form the most intelligent system known. Neural networks are a new class of computer systems, although currently being simulated on the commonly used serial architecture known as digital computers. A neural networks consists of a large number of simulated neurons, connected to each other in much the same way that the brain's neurons are connected. Despite the diversity of network paradigms (architectures), they consist of similar components. In the general form of a neural network, the unit analogous to the biological neuron is referred to as processing element (P.E.). The network consists of many of those elements usually organized into a sequence of layers or slabs with full or partial connections between successive layers specifically designated. Fig. 2.7 shows the



- Very simple processors.
- Interconnections with weight values.
- Various paradigms available.
- Various training algorithms.
- Learn through a set of examples.

Fig. 2.7: Neural Network Components.

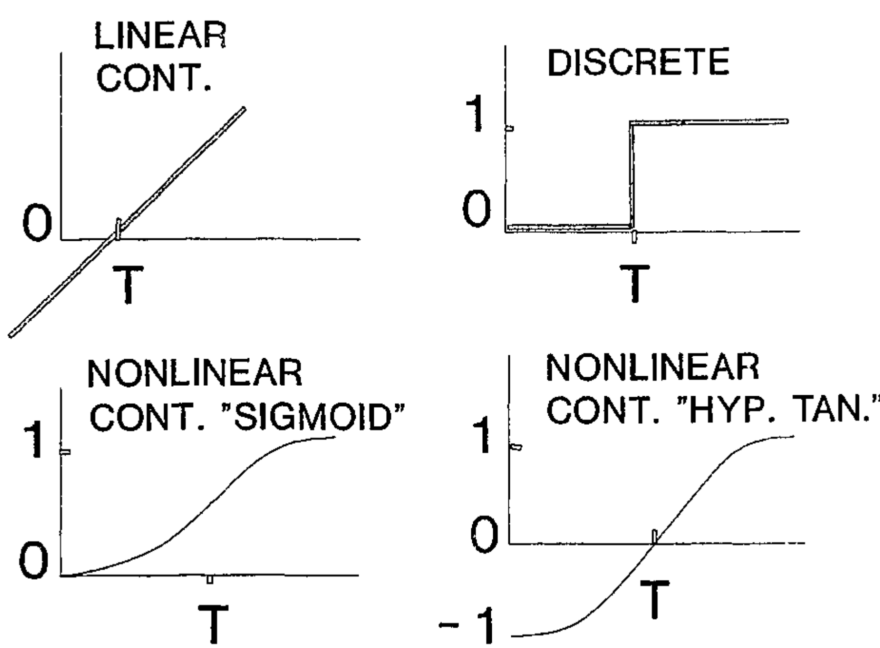


Fig. 2.8: Transfer functions.

general neural network configuration. The neural network has an input buffer (not considered as a layer) to which the input data is presented, and an output layer which holds the response of the network to a given input. Layers distinct from the input buffer and the output layer are called hidden layers. As shown in Figure 2, a processing element (artificial neuron), usually excluding those in the input buffer, performs summation (Σ) and transfer function (**F**) to determine the value of its output. In essence, a set of inputs, each representing the output of another neuron in a preceding layer, are received by the processing element (P.E.). Each input is multiplied by its connection weight (analogous to the human synaptic strength) and all of the weighted inputs are then summed to produce the neuron "NET" or activation value, which is then modified by the transfer function. The transfer function can take many forms, see Fig. 2.8. Neural networks, generally, could be of two types: 1) **"Feed-forward"** or **"Non-recurrent"** where the network P.E. connections and hence the information flow are in one direction as shown in Fig. 2.7; and 2) **"Recurrent"** which exhibits a more general network structure that allows feedback connections through weights extending from one layer to another or to itself.

Training a neural network is the iterative process of changing the values of its weights using a training algorithm, in response to the input pattern(s), until the error between the network's own result(s) and the desired output(s) is within a specified limit. The task is to arrive at a unique set of weights that are capable of correctly associating all example pattern(s), used in learning, with their desired output pattern(s). There are two main types of learning: supervised and unsupervised. Supervised learning refers to the case when the network is presented with some input examples and their desired responses (associations). The desired outputs are used in this case to teach the network the correct responses. Unsupervised learning, on the other hand, refers to the case when only input examples are presented to the network, the network must organize itself on the basis of the input stimuli it receives. Usually, a training algorithm is used and held responsible for specifying how weights adapt in response to a learning example (Table 2.3). Training algorithms also may include deterministic or statistical procedures for the network weights adjustment. If the network training is successful, the network can be recalled by applying a set of inputs to the network, then, the network is expected to produce outputs that are satisfactorily close to the desired set of output(s) used in training.

Table 2.3: Neural network training algorithms.

Training Algorithm	Reference	Learning Type	Weight Adjustment
Hebbian	(Hebb 1949)	Supervised	Deterministic
Perceptron	(Rosenblatt 1959, 1961)	Supervised	Deterministic
Widrow-Hoff	(Widrow and Hoff 1959)	Supervised	Deterministic
Grossberg	(Grossberg 1974)	Supervised	Deterministic
Delta rule	(Rumelhart et al 1986)	Supervised	Deterministic
Boltzmann	(Hinton and Sejnowski 1986)	Supervised	Probabilistic
Kohonen	(Kohonen 1984)	Unsupervised	Deterministic

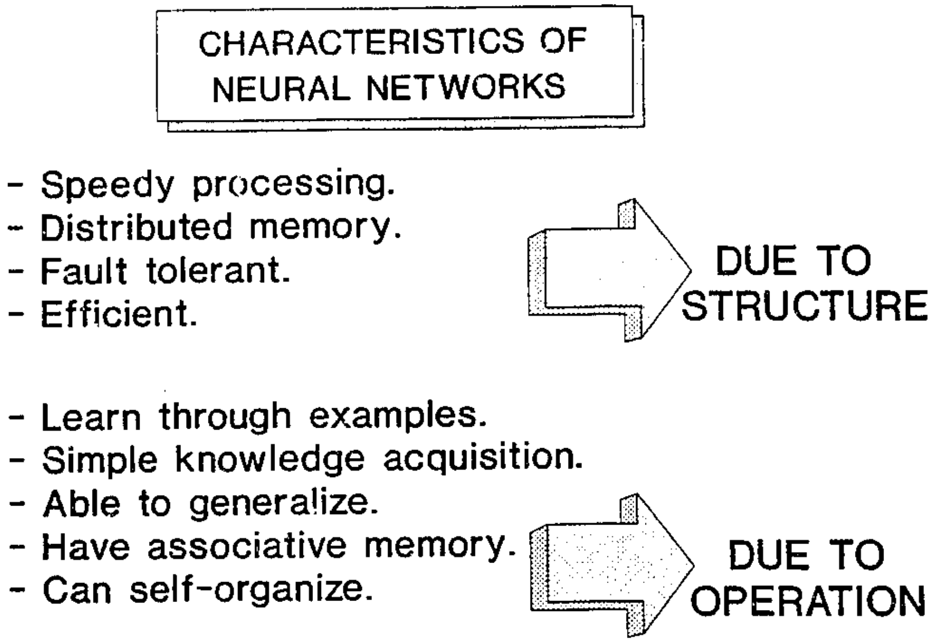


Fig. 2.9: Characteristics of Neural Networks.

Neural networks enjoy several characteristics (see Fig. 2.9) that distinguish them from other AI traditional architectures including expert systems (Wassermann 1989, Pao 1989, NeuralWare 1988, Gallant 1988, and Castelaz et al 1987). Some of these characteristics include:

- Particularly suited for pattern recognition tasks where large number of attributes have to be considered in parallel.

- Learn by example: unlike expert systems, neural networks learn many example patterns and their associations (i.e., desired outputs or conclusions). These examples could be elicited from experts without the need for asking how and why did they come to that conclusion. Problems generally associated with knowledge acquisition are therefore eliminated.

- Produce fast responses, irrespective of their requirement of large computer time for learning. This is due to the parallel structure of neural networks.

- Could extract classification (clustering) characteristics from large number of input examples as in the case of unsupervised learning. If for example, large number of field data are collected from a construction site, a suitable network can identify the different clusters (groups or classes) that characterise the whole population.

- Have distributed memory: the connection weights are the memory units of the network. The value of the weights represent the current state of knowledge of the network. A unit of knowledge, represented for example by an input/desired output pair, is distributed across all the weighted connections of the network.

- Have associative memory: the network responds in an accretive or interpolative way to noisy, incomplete, or previously unseen data. Autoassociative network, where input is equal to desired output, can produce a full output if presented with a partial input. This property is called **"Generalization"**.

- Fault tolerant: since memory is distributed, failure of some processing elements will slightly alter the overall behaviour of the network. Contrary to that, failure of any small part in a traditional computing system will stop its performance. This characteristic is very well-suited for applications where reliable systems need to be developed from less reliable components.

- Could represent uncertainty: measure of **"Belief"** could be incorporated into a neural network formulation.

- Require less amount of storage memory, since there is only one set of network weights capable of representing a large space of stored patterns.

Table 2.4 : Description of Commonly Used Neural Paradigms
(Bailey and Thompson, 1990)

Paradigm	Type of Input Accepted	Type of Output	Transfer Function	Number of Layers	Size of Hidden Layers	Connectivity	Learning Algorithms	Training Method	Network Type
Perceptron	Continuous	Real #, Class	Perceptron	1	N/A	Random	Perception Convergence, Delta Rule	Supervised	Non recurrent
Back Propagation	Continuous	Pattern, Real #, Class	Sigmoid, Hyperbolic Tangent	2 or more	Small to medium	Fully interconnected, random, slabs	Generalized Delta Rule	Supervised	Non recurrent
Counter Propagation	Continuous	Pattern, Real #, Class	Kohonen & Sigmoid	2 or 4	Equal to # of data categories	Fully inter-connected	Kohonen & Grossberg	Supervised, unsupervised	Non recurrent
Boltzmann Machine	Binary Continuous	Pattern, Real #, Class	Varies	2 or more	Small to medium	Fully inter-connected, random	Boltzmann	Supervised	Non recurrent
Hoppfield Network	Binary Bipolar[1]	Pattern, Class	Hard Limiting	1	N/A	Fully inter-connected	Hoppfield	Supervised	Recurrent
BAM	Bipolar	Pattern, Class	Clamped Linear	1	N/A	Fully inter-connected	BAM	Supervised	Recurrent
ART_2	Grey Scale[2]	Pattern, Class	Sigmoid	1	Increases by Data Category	Fully inter-connected	ART_2	unsupervised	

1. (-1 and +1) values as opposed to binery (0 and 1)
2. Activation level opposed to Boolean.

Table 2.5 : Performance Characteristics of Commonly Used Neural Paradigms.

Network Paradigm	Developer	Training Time *	Execution Time *	Info. Content *	Advantages	Disadvantages (Limitations)	Utilization *
Perceptron	Rosenblatt 1961	Medium	Fast	Low	- Very Simple	- limited representation capability.	Low
Back Propagation	Rumelhart et al. 1986	Slow	Fast	High	-(a) Powerful and accurate association. -(b) Suitable for static environment (input does not change with time).	-(1) Could be trapped in local minima or paralyse. -(2) Not suitable for real-time applications. -(3) No incremental learning.	High
Counter Propagation	Hecht-Nielson 1987, 1988	Medium	Fast	High	- Good for rapid prototyping. - (b) above. - Can generate a funtion and its inverse (wassermann 1989). - Has powerful probabilistic capabilities.	- Not as accurate as back-propagation. - (1), (2), and (3) above.	High
Boltzmann Machine	Hinton and Sejnowski 1986	Slow	Slow	High	- Alleviate Local minima problem. - Use probabilistic training method.	- Very slow. - (2), (3) above.	High
Hopfield Network	Hopfield 1982, 1984	Fast	Medium	Low	-(c) Provide dynamic (real-time) performance. -(d) Sufficiently responds to noisy, incomplete, or unseen inputs. -(e) Can be used for optimization.	- Stability not guaranteed. -(4) Memory Limitations (stored patterns). - Difficult to formalize training method. - (1) and (3) above.	High
BAM	Kosko 1987 a, b	Fast	Fast	Low	- (c), (d) and (e) above. - Easier and Systematic training methodology than Hopfield. - More simple and faster than Hopfield. - No stability problem. - Provide adaptive performance.	- (1), (3) and (4) above.	Low
ART-2	Carpenter and Grossberg 1987	Fast	Medium	High	- Facilitate learning new examples without destroying previous experience (incremental learning).		High

* Reference: Bailey and Thompson 1990.

2.7.2 Neural network variations

Several innovative neural network paradigms have recently been developed to provide distinct capabilities. These include: Perceptron (Rosenblatt 1961), Backpropagation (Rumelhart et al. 1986), Counterpropagation (Hecht-Nielsen 1987, 1988), Boltzmann Machine (Hinton and Sejnowski 1986), Hopfield (Hopfield 1982, 1984), BAM (Kosko 1987 a, b), and ART (Carpenter and Grossberg 1987). These paradigms vary a great deal regarding the type of input patterns they accept, the output patterns they produce, the learning characteristics and also the limitations they exhibit. Tables 2.4 and 2.5 summarize the description and the performance characteristics, advantages, and limitations of these commonly used paradigms. Tables 2.4 and 2.5 can, therefore, provide a guide for selecting a suitable neural network paradigm for modelling the markup estimation problem.

2.8 Conclusion

This chapter has reviewed previous work related to the process of bid preparation: MIS and databases, cost estimating, planning and scheduling, risk assessment and bid unbalancing. The literature review reveals that most of the procedures used under the prevailing competitive bidding environment to estimate the construction costs and schedule lack integration and structure. Proper design of a MIS and contractor databases is needed, a structured WBS methodology need to be adopted, and a practical planning and scheduling model of high-rise buildings is required. Also, the impact of a schedule on the estimated direct costs need to be considered.

Markup has been identified as a major risk assessment measure in construction bid preparation. However, the procedures described in the literature for risk assessment at the markup level, yet, fail to model how the project qualitative factors could be realistically considered into the bid estimate. Due to the highly subjective nature of this problem, conventional decision making tools, including expert systems, are inadequate to model the problem. Neural networks, on the other hand, seem to be a promising tool for application to such analogy-based problem.

CHAPTER 3

INTEGRATED BID PREPARATION METHODOLOGY

3.1 Introduction

This chapter exhibits the establishment of the proposed bid preparation methodology. System requirements are identified, based on the literature review of chapter 2, along with the quantitative and qualitative aspects that need to be considered in a practical system. A general procedure of formulating an optimum problem-solving strategy is presented and applied to the bid preparation problem in an effort to optimize the system benefits under the identified system requirements and development constraints. The system architecture and components are then described and a suitable neural network paradigm is selected as an environment for modelling the markup estimation problem. The proposed bid preparation methodology is then established and its detailed specifications described.

3.2 System Requirements

Based on the literature review of chapter 2, the basis of an integrated methodology that aids the preparation of timely, practical, and competitive bid proposals is the incorporation of adequate assessment for three principle functions: (1) direct and indirect cost estimation; 2) risk assessment and optimum markup estimation; and 3) optimum bid unbalancing and cash flow optimization. On one hand, functions

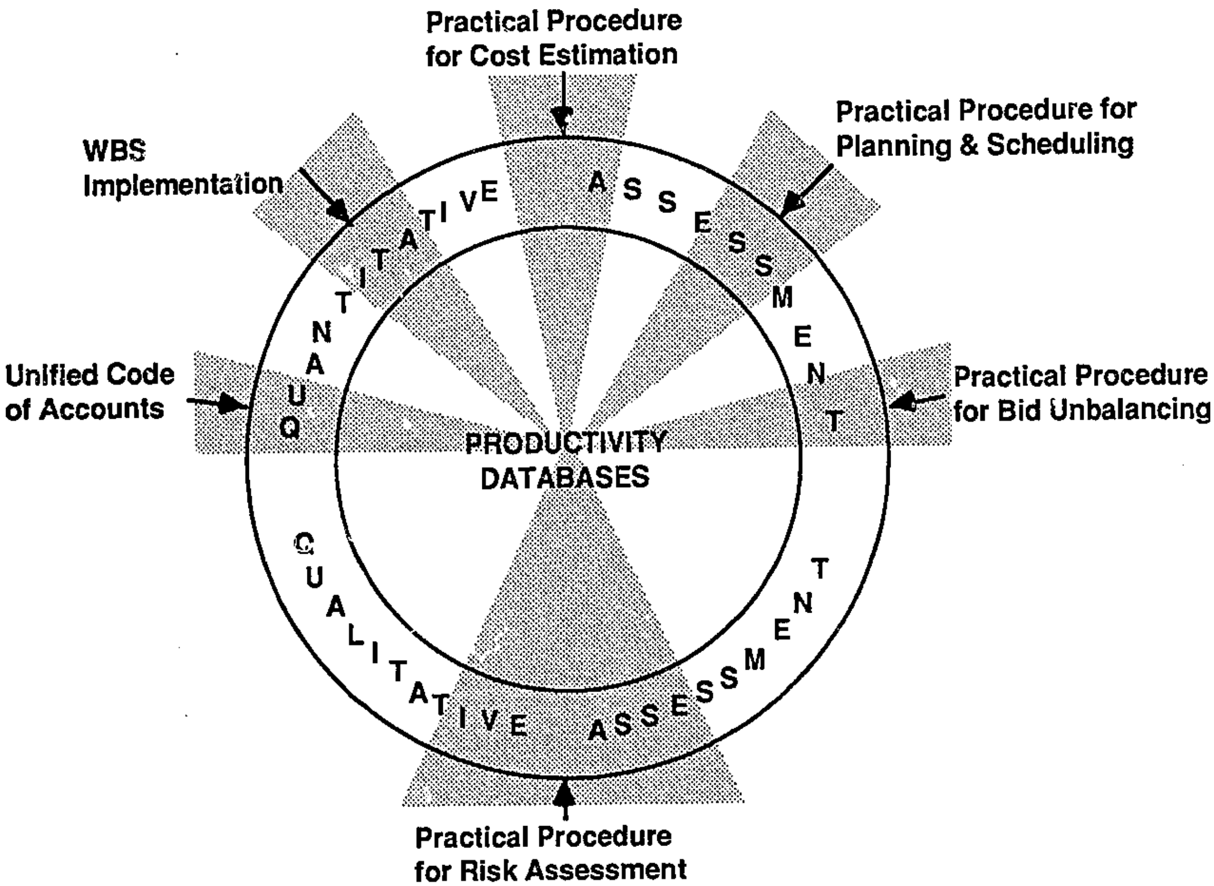


Fig. 3.1: Basis for an effective cost estimation methodology.

1 and 3 are mainly quantitative in nature and can be adequately addressed through effective implementation of (see Fig. 3.1): productivity databases, a unified code of accounts, a properly detailed WBS, practical procedures for cost and schedule calculations, and a suitable algorithm for cash flow optimization. Function 2, on the other hand, is qualitative in nature and requires assessment of a number of risk-related factors that govern the determination of % markup in practice. As shown in Fig. 3.1, a practical procedure is necessary to consider the subjective nature of the decisions involved and establish an adequate problem-solving mechanism utilizing the inherently implicit domain knowledge. In accordance with these quantitative and qualitative aspects of Fig. 3.1, a list of 14 items is compiled (Table 3.1) as the specifications and basic requirements in the proposed bid preparation system.

In addition to such requirements, the system would ideally be designed to have the following characteristics, in order to overcome the deficiencies in current estimating tools and methodologies (described earlier in section 1.2):

- Integrated; adopts a methodology that integrates different management functions that consider resource constraints and the impact of the schedule on estimated costs. Such methodology provides practical estimates, minimizing redundancy and work repetition.
- Practical; incorporates assessment to the quantitative as well the qualitative aspects, utilizing possible tools (algorithmic and AI-based).

Table 3.1: Specifications of a practical bid preparation environment.

	Specification
1.	Designing and implementing productivity databases for resources, tasks, and activities.
2.	Employing a WBS and a unified code of accounts.
3.	Establishing a direct link between WBS elements and appropriate contract items.
4.	Compiling direct costs through user-interactive quantity take-off and direct reference to the system databases.
5.	Allowing both repetitive and non-repetitive activities to be specified at the WBS elements.
6.	Direct interfacing between cost and schedule calculations.
7.	Developing an algorithm for scheduling projects incorporating repetitive and non-repetitive activities.
8.	Updating of generated schedules based on weather conditions and other productivity-related factors.
9.	Updating of costs based on the schedule impact, resource constraints, and work load of owned resources.
10.	Generating various reports that can provide a guide in estimating indirect costs.
11.	Identifying a set of qualitative factors that represent the project risk pattern.
12.	Developing a practical neural network decision aid regarding optimum markup estimation, in response to the identified project risks.
13.	Incorporating a simple and practical procedure for optimum bid unbalancing, cash flow optimization, and final bid compilation.
14.	Establishing a feedback cycle to modify the system databases based on final bid outcomes (win/loose and actual performance).

- Modular; allows future expansions and enhancements.
- Information-intensive; incorporates suitable means (e.g., databases, knowledge-bases, patterns) to store, process; and utilize available data and knowledge in order to improve the practicality of estimates preparation.
- Efficient; user interactive and fast processing.
- Flexible; previous project estimates could be used as templates for new ones and databases are saved and down-loaded with the project files for records.
- Transparent; changes in the databases are directly reflected on the estimate.

3.3 Formulating an Optimum Problem-Solving Strategy

Formulating and designing a suitable problem-solving strategy for the bid preparation problem embodies the selection of a an optimum combination of problem-solving techniques, tools, and development environments that: 1) suit the nature of all sub-problem components, 2) meet the accuracy required; 3) facilitate integration and effective flow of information among all system components; 4) meet user needs and level of sophistication; 5) utilize current industry practice and domain knowledge; 6) maximize the benefits and minimize the limitations of the individual techniques; and 7) are practical and cost effective within the system requirements and development constraints. Nowadays, there exists a large number of problem-solving techniques (algorithmic, knowledge-based, and analogy-based)

PROBLEM-SOLVING TECHNIQUES

	ALGORITHM-BASED	KNOWLEDGE-BASED	ANALOGY-BASED
ADVANTAGES	- Effective number processing and routine computations. - Effective manipulation of large volume of data. - Consistant and reliable results. - Easily verifiable. - Fast processing.	- Knowledge and control are separated. - Mid-run explanations available (Transparent). - Facilitate integration with needed numerical computations. - Preserve experiences, heuristic knowledge and rules of thumb. - Could provide solutions by reasoning.	- Solve problems by analogy, considering large number of attributes in parallel. - Learn through few examples and generalize solutions. - Suitable for parallel computations. - Could provide fast solutions without deductions and/or numerical computations.
LIMITATIONS	- Knowledge and control are inseparable. - Rigid formulation. - Mid-run explanation is normally impossible. - Difficult to maintain.	- Limited to reasoning-based problems. - Experts must be available & able to articulate how they arrive at solutions. - Not suitable for problems including induction or analogy.	- Limited to analogy-based problems. - Does not provide explanation.

Fig. 3.2: Advantages and limitations of the main problem-solving techniques.

in addition to different development environments (conventional computer languages, spreadsheet programs, object oriented programming, hierarchical databases, relational database, object oriented databases, parallel computations, and several customized environments like CAD systems). The advantages and limitations of those techniques vary considerably (see Fig. 3.2), making the selection of an optimum problem-solving strategy for a particular application a difficult task that if not properly performed, a substantial cost overrun in the system design and/or implementation may eventually result, in addition to a less than adequate system performance.

As presented in Fig. 3.3, problems themselves have been classified in the literature into three main categories: 1) structured; 2) semi-structured; and 3) unstructured (Reynolds 1988; Gorry and Morton 1971; Davis and Olson 1985). For each problem type, Fig. 3.3 presents the characteristics, suitable problem-solving technique (from those mentioned above), possible development tools, and the solution processing requirements. For simple problems, which can easily be classified under one of the given categories, Fig. 3.3 can guide the selection of a suitable problem-solving strategy. However, most real-life problems, including the bid preparation problem in hand, are complex and frequently incorporate several components (sub-problems) of different types. A structured procedure is, therefore, required to arrive at an optimum problem-solving strategy that maximizes the benefits and minimizes the limitations of the individual tools utilized. A solution

PROBLEM TYPES	Structured Problems	Semi-structured Problems	Unstructured Problems	
			REASONING-INTENSIVE	ANALOGY-INTENSIVE
PROBLEM CHARACTERISTICS	- Routine & Repetitive. - Guideline for problem-solving already exist.	- Solution requires qualitative assesment of multi-criteria. - Problem requires probabilistic (nondeterministic) analyses.	- Nonroutine. - Reasoning-intensive. - No preestablished decision procedure. - Data requirements are not completely known.	- No preestablished decision procedure. - Pattern Recognition & Analogy-intensive. - Solution is based on Gut feeling.
SUIT. TECH.	ALGORITHM-BASED	ALGORITHM & PROBABILITY-BASED	KNOWLEDGE & HEURISTIC-BASED	ANALOGY-BASED
DEVELOPMENT TOOLS	- Linear Programming. - Network Models. - Dynamic Programming. - Procedural.	- Analytical Hierachy Process (AHP). - Simulation Models. - Multi Attribute Utility Theory.	- Knowledge-Based Expert Systems.	- Neural Networks. - Case-Based Reasonning Systems.
REQUIRE-MENTS.	- Data for computation. - Domain historical records.	- Domain historical records. - Brain storming sessions. - Establishing weights for decision criteria. -Probabilistic analysis.	- Domain knowledge acquisition. - Comprehensive databases. - Brain storming sessions. - Powerful hardware and software for system development and use.	- Set of historical cases. - Powerful hardware for system development.

Fig. 3.3 : Basic Types of Problems.

strategy for complex problems, such as the bid preparation problem, need to be designed with emphasis on front end planning (Moselhi et al. 1990c) in order to achieve the objectives and avoid cost and time overruns.

A structured procedure for arriving at optimum problem-solving strategies is presented in Fig. 3.4 (Moselhi et al. 1992b). The procedure starts by breaking down a complex problem into its basic components (sub-problems) that can be generally categorized into one of the problem types described earlier: structured, semi-structured, and unstructured. In consistency with this step, the bid preparation problem could be broken down into component sub-problems (see Fig. 3.5). As a first level of breakdown, sub-problems could represent the management functions needed to be integrated by the system: 1) cost estimation; 2) planning and scheduling; 3) risk assessment and optimum markup estimation; and 4) bid unbalancing. Those sub-problems can be further decomposed into smaller sub-sub-problems that do not lend themselves to one problem type (Fig. 3.5).

In selecting development tools and an integration media (e.g., hierarchical databases, relational databases, object oriented programming tools, spreadsheets, expert system environment, and customized environments like CAD systems) for the sub-problems, several criteria and constraints have to be accounted for. As shown in Fig. 3.4, generally, system development criteria include: user friendliness, flexibility in representation, maintainability, security, and expandability. In addition

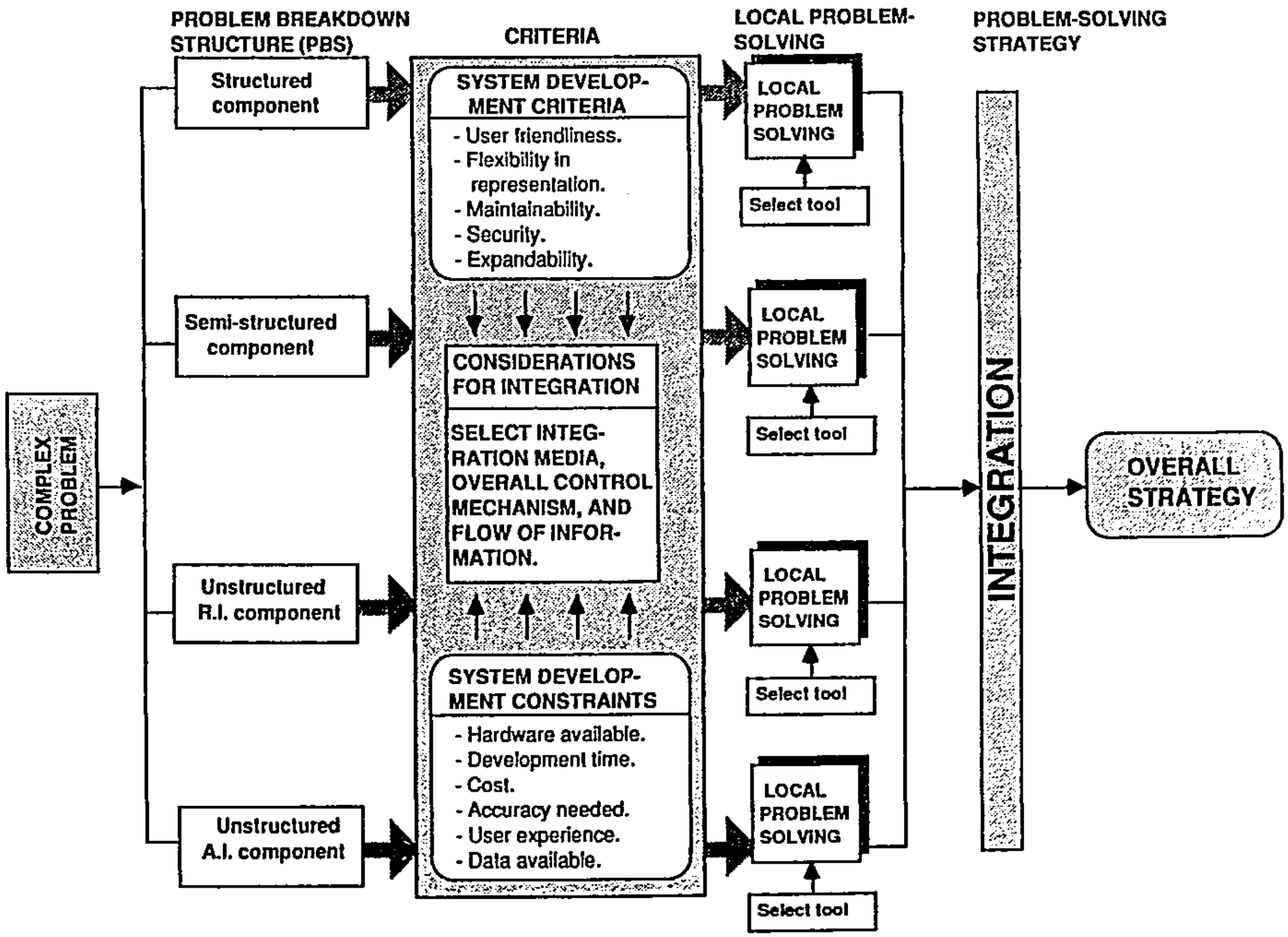


Fig. 3.4: A procedure for selecting an optimum problem-solving strategy.

to those, system development constraints may, generally, include: hardware available, development time, cost, accuracy needed, user experience, and data available. For the bid preparation problem, system development criteria and constraints are established and represented by the system requirements and desired characteristics described in section 3.2 above. Based on these criteria and constraints, local development tools and integration media can be established along with the overall control mechanism and flow of information. This allows local problem-solving and further integration among the sub-problems, forming an overall problem-solving strategy (see Fig. 3.4). Often, for each sub-problem, there are many tools that can be used. Various combinations of the sub-problems' tools may mandate different integration schemes and accordingly different overall strategies will result. For example, work breakdown structure (see Fig. 3.5) is a planning task that can be categorized as a reasoning-intensive unstructured problem and, thus, lends itself well to knowledge-based expert systems application. A possible strategy, therefore, is to use KBES as the development tool and integration media to produce accurate WBS, based on domain experts' knowledge. As an alternative strategy, however, is the use of an algorithm-based technique (e.g., user selection of WBS elements from databases and WBS templates) that could be developed in a less costly manner to simplify the task, although a less accurate WBS may result. An optimum strategy, therefore, is a one that maximizes the overall benefits gained, with respect to the level of effort required and the degree of satisfying the problem requirements and development constraints

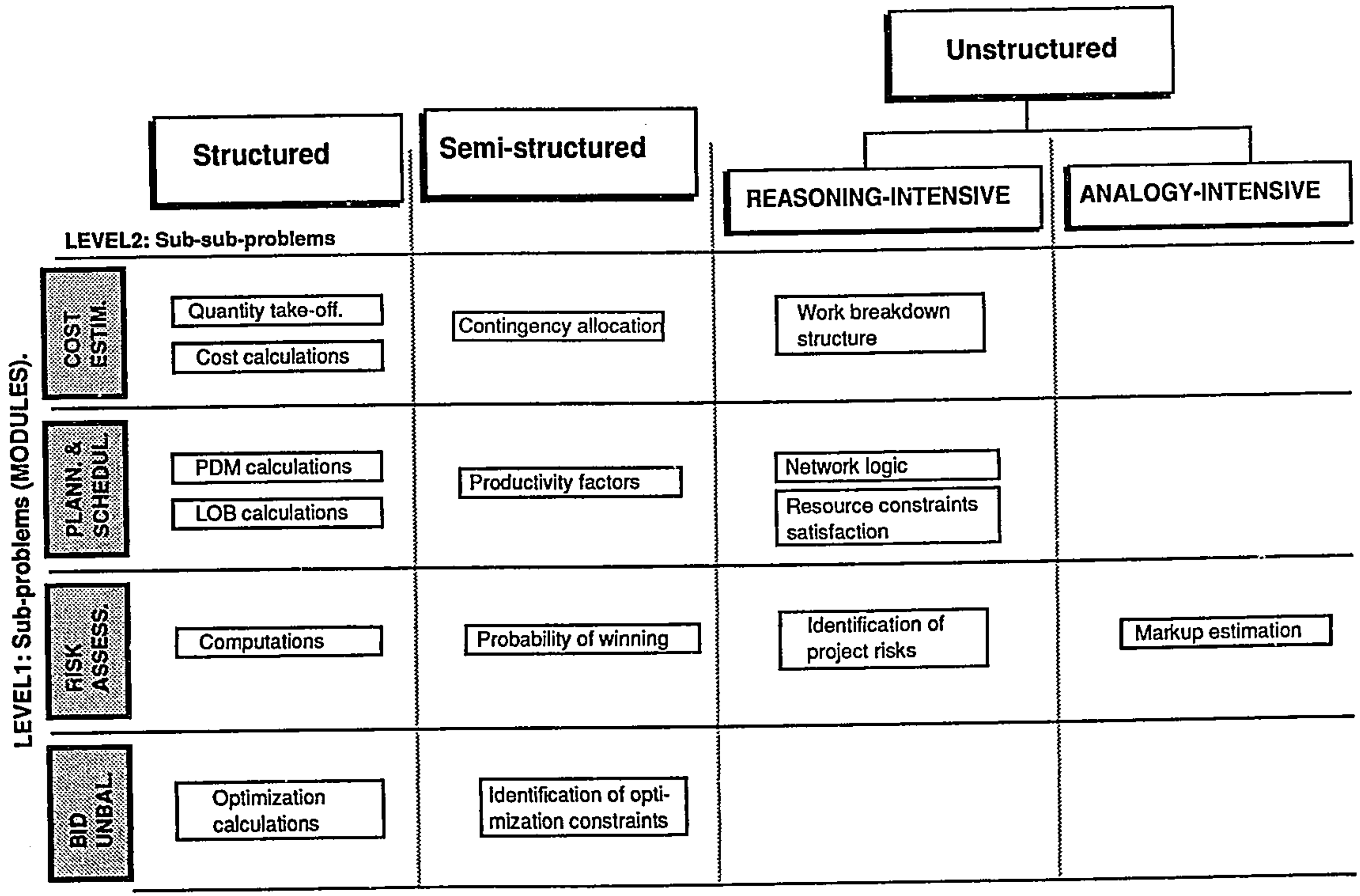


Fig. 3.5: Breakdown structure of the bid preparation problem.

present. The devalopment of such an strategy for the bid preparation problem is discussed in the next section.

3.4 System Architecture and Components

The computerized system, as discussed in section 3.3 above, is designed in a modular architecture, incorporating four main modules (see Fig. 3.6): a cost estimation module, a planning and scheduling module, a markup estimation module, and a bid unbalancing module. Those modules share common databases managed by a database management system that links the system four modules, forming an integrated management information system (MIS), see Fig. 3.6. To establish an optimum problem-solving strategy for the bid preparation problem, in consistency with the procedure of Fig. 3.4, the functions performed within each of the system components and their local development strategies are described next.

3.4.1 Contractor Databases

The data needed for the system different modules has to be contained in a core of databases, separate from the procedure by which these data are utilized. This permits faster and easier modification of the data with no consequent change in the procedure used. The contractor productivity-based databases needed include:

- Resource databases: labour, equipment, crews (combination of labour and equipment), construction material, permanent material (e.g., re-usable forms), and subs.

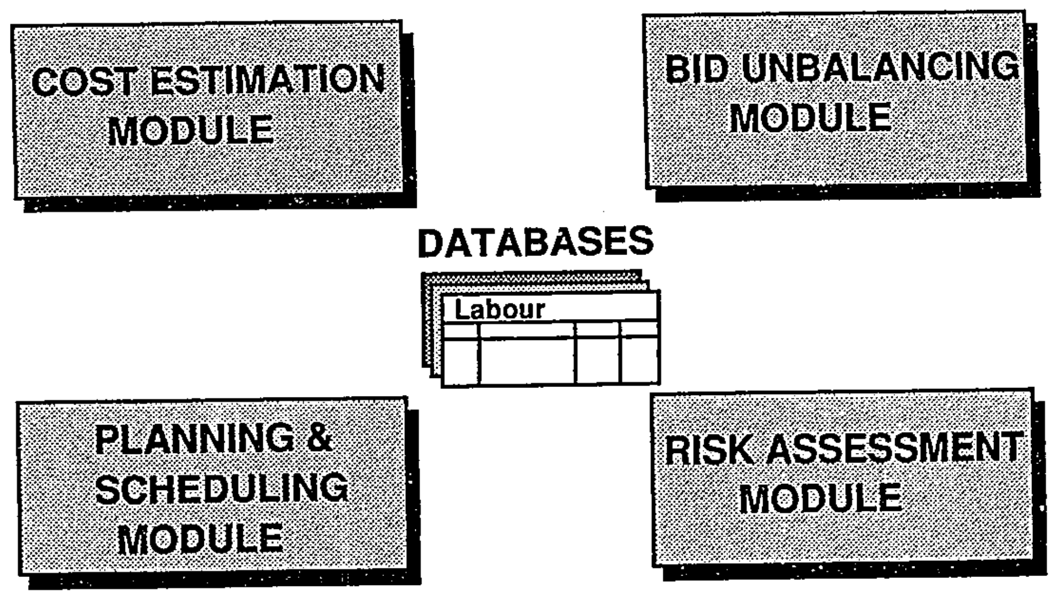


Fig. 3.6: System Components.

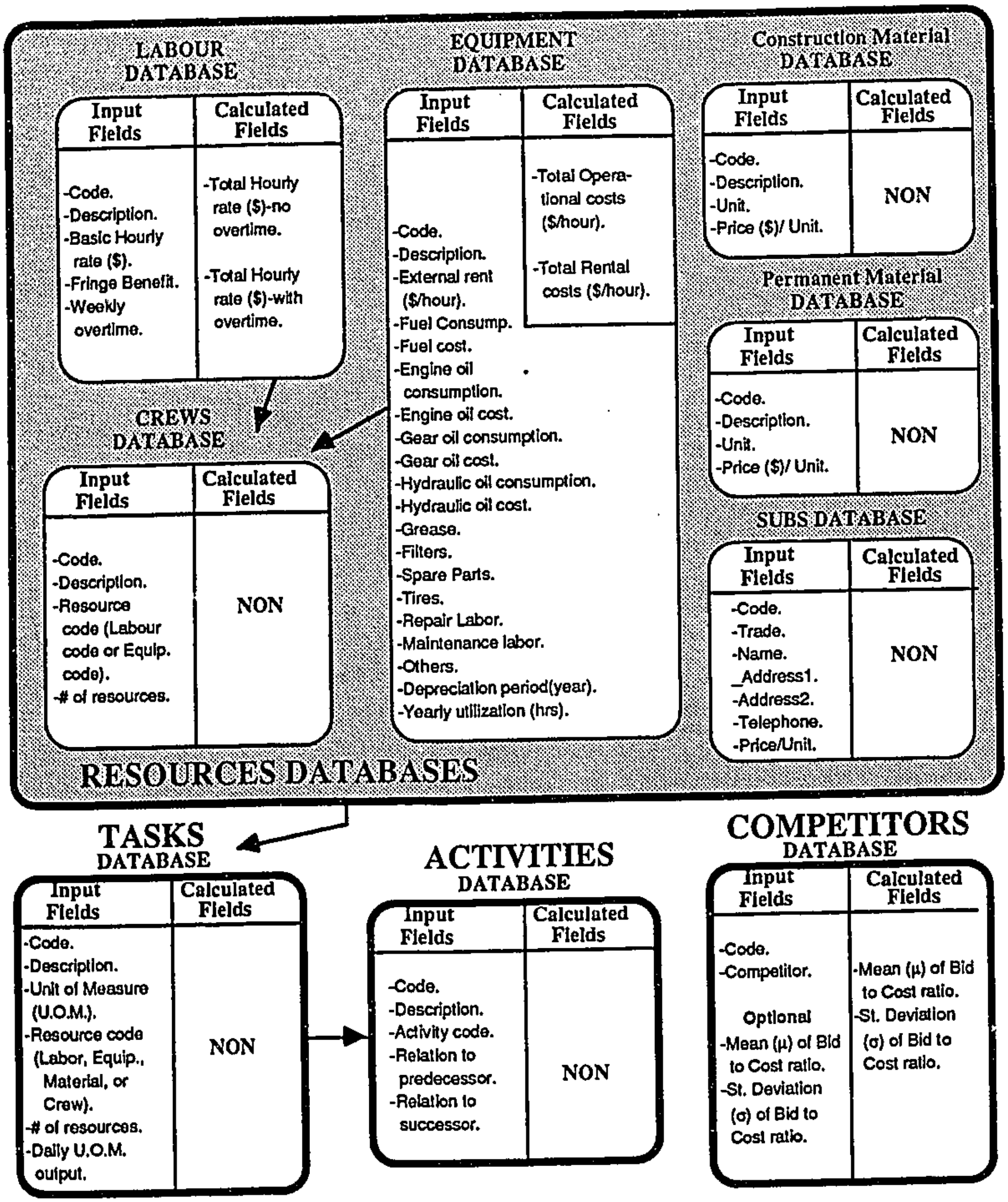
- Tasks database containing a reference to the resources used to perform the task and the productivity of those resources in terms of daily output per unit of measure.
- Activities database containing groups of tasks (work package).
- Competitors database containing data regarding the past performance of competitors in past bids.
- Resource work load database containing information regarding the degree of involvement of owned resources (mainly equipment) in jobs under construction and jobs being estimated with a probability of being won.

The last database containing data regarding the work load of owned equipment can be utilized by a simple procedure for addressing the cost implications of a resource constrained schedule. This is because contractors may assign rental rates, rather than operational rates, to the hourly operation of an equipment if the owned resources are expected to be working on other projects. Such a procedure is important for civil engineering projects with high equipment utilization. Also, the cost implication of dead line constrained schedule could by addressed in a simple manner by calculating the additional cost of labour overtime. Thus, in order to facilitate the assessment of constrained schedules (see section 3.4.3), the labour database has to contain data that facilitate the calculation of overtime hourly rates, and the equipment database has to contain data regarding the hourly rates of equipment if externally acquired.

The record structure of contractor databases is designed and presented in Fig. 3.7. The design is based on the system requirements described earlier and input from Montreal area contractors in building and civil engineering works (Moselhi et al. 1991d). The data necessary for the databases could be obtained from several sources in construction. Published material is one important source which, although take different forms and structure, describe data regarding the combination of crews, average productivity data, different construction activities, and work packages (e.g., Means 1990).

A suitable implementation media for these databases is a relational database environment, with a database management system that facilitate user-interactive input and output of data. The relational database management system would allow capabilities of partial calculations (e.g., calculations of labour hourly rates based on a specified number of overtime hours), overlooking other databases, referencing to other databases (e.g., crews refer to a labour and an equipment databases), updating, and printing of data (Moselhi 1989, 1990). Such data can be readily utilized in the system being proposed.

The database management system should interact with the common databases and provide timely, relevant, accurate, complete, and formatted data for the different management functions. This is achieved through an integration media that is responsible for sending proper instructions to the database management system



Notes:
- Databases are designed based on input from Montreal area contractors in heavy civil and buildings.
- All the data bases allow user specification of the date when last modified.
- Databases data could be saved with the project data files.

Fig. 3.7: Record structure of contractor databases.

and correspondingly utilize the outputs of its operation. Possible integration media are: a knowledge-based expert system, a database program, or an interfacing algorithm, with the latter being the simplest to develop.

3.4.2 Cost Estimation Module

The cost of construction are two main components: direct and indirect. Detailed estimation of the direct cost of construction constitutes of the following tasks (Fig. 3.5): 1) work breakdown structure of the project into work tasks and activities; 2) quantity take-off from complete drawings; and 3) computation of direct costs attributed to the identified work tasks based on resource productivity and subcontractors' quotations, in consideration of the method of construction used. On one hand, work breakdown structure is a planning task that can be categorized as a reasoning-intensive unstructured problem. On the other hand, quantity take-off and cost computations are mainly algorithmic in nature and lend itself well to algorithm-based development tools. As mentioned previously, although WBS lends itself well to knowledge-based expert systems application, the use of an algorithm-based technique (e.g., user selection of WBS elements from databases and WBS templates) is an alternative that is less costly to develop and produces a WBS that is accurate enough for implementation at the bidding stage. In the integrated methodology being proposed, proper implementation of the WBS in addition to faster and more accurate cost computation can be achieved through:

- Hierarchical decomposition of the project into different levels of detail, from a broad level (definable areas), down to a very detailed level incorporating well-defined work elements (Work Packages) usually of reasonable size and duration. These work packages are tied to the company code of accounts and the database(s) of resources, unit costs, and productivity data.

- Distinguishing repetitive and non-repetitive WBS elements so as to permit utilization of the repetitiveness in producing faster cost computations and further application of proper planning and scheduling tools, see Fig. 3.8 for a WBS template for high-rise buildings.
- Linking the WBS elements at various levels to the contractor's organization breakdown structure (OBS), identifying different responsibility levels and their appropriate reporting formats.
- Linking the WBS elements to appropriate contract items to facilitate the aggregation of the direct costs in the proper bid proposal format. This further enables the application of a procedure for optimum bid unbalancing.

Indirect costs and contingency need also to be estimated. While indirect costs cover the project overheads and a part of the firm general overhead, contingency covers mainly the cost for unforseen conditions. As discussed in the introduction of chapter 1, the proposed bid preparation methodology does not emphasize on using formal techniques (e.g., Monte Carlo simulation) for contingency estimation, due to the generally limited time permitted for contractors to prepare bids. Rather, the methodology compensates for contingency in the form of practical and more

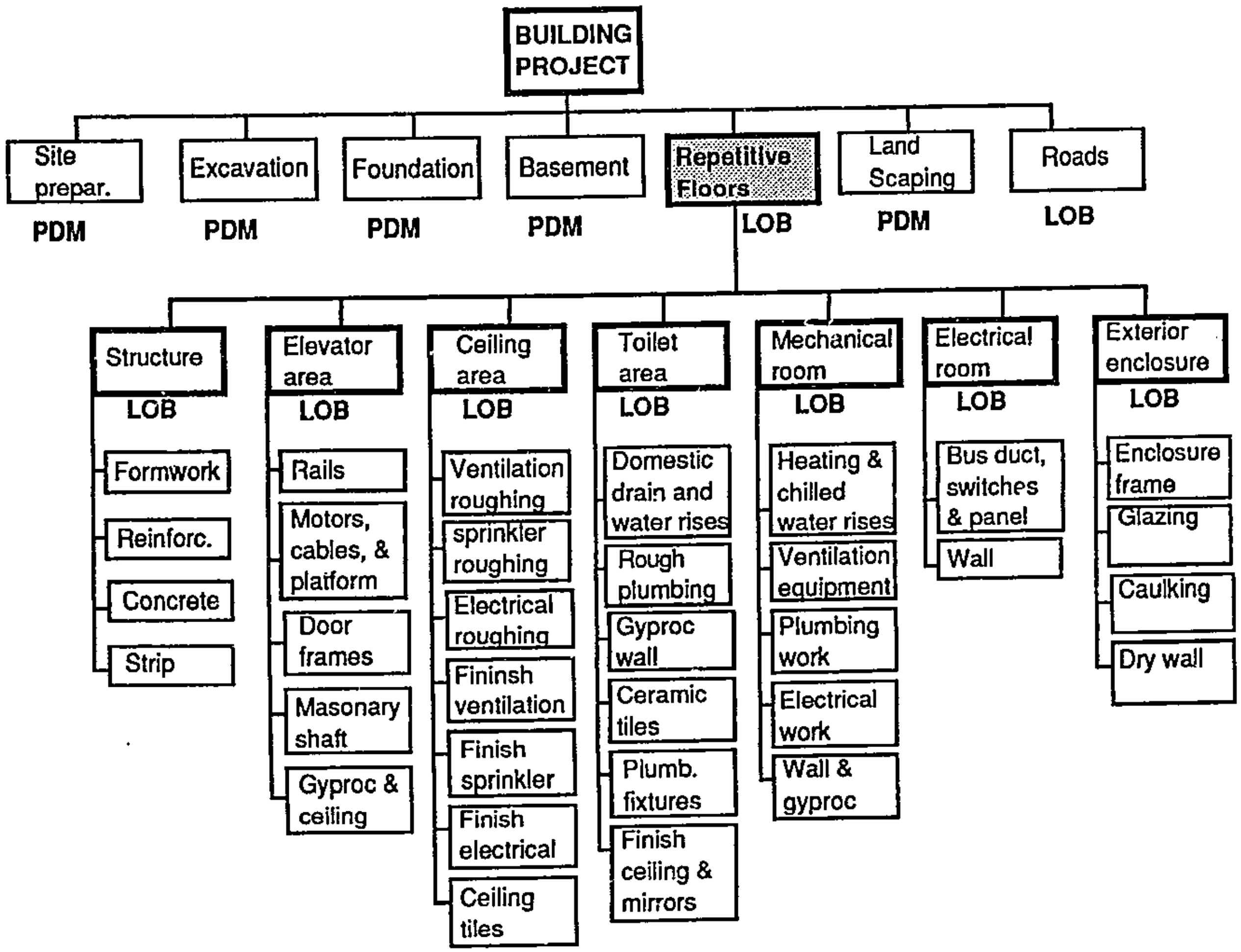


Fig. 3.8: WBS template for a high-rise building.
(based on Laramee 1983)

certain cost estimates, an additional indirect cost item that could be estimated by the contractor, and a properly estimated markup value. Formal techniques could, however, be utilized for contingency allocation, if bidding time permits such analysis.

To estimate indirect costs, contractors usually have a list of possible items and estimated costs are assigned to those applicable to the project. Published data (e.g., Bentley 1987) provide lists of possible items and sets of guidelines in the estimation of indirect costs which could be utilized in the proposed system. Since indirect costs are mainly schedule-based, realistic costs are practically estimated based on cost and schedule calculations. When the total of the indirect costs and contingency is calculated, it cannot readily be attributed to a certain work package or a certain contract item. Distribution of these costs over the contract items is discussed later in the bid unbalancing module (section 3.4.5).

From the above discussion, a suitable development tool for the cost estimation module is algorithmic procedures that facilitate: user-interactive and template-based WBS, automated cost computations, and user-interactive indirect cost estimation. A suitable integration media would also be an interfacing algorithm that links the cost estimation module to necessary databases and other system modules.

3.4.3 Planning and Scheduling Module

The purpose of performing planning and scheduling at the bidding stage is to evaluate the cost implications of the project schedule. This is attained through 3 aspects: 1) assessment of practical productivity factors and their impact on the activities durations and accordingly their direct costs; 2) assessment of the impact of the resource constrained schedule on the project direct costs; and 3) assessment of the total project duration, permitting practical indirect cost computations. In addition, planning and scheduling enables the calculation of an expense curve (S-curve) for the project, permitting the application of a procedure for optimum bid unbalancing considering the contractor's cash flow constraints.

Planning and scheduling can constitute the following tasks (Fig. 3.5): (1) specifying logical relationships among the WBS activities; (2) performing algorithmic PDM and LOB computations; (3) assessing productivity factors; and (4) allocating resources. Although some of these tasks may lend itself to reasoning-intensive solutions (e.g., tasks 1 and 4), algorithmic tools can be used to produce acceptable accuracy at the bidding stage. Therefore, task (1) of specifying logical relationships among the project activities could be identified through user-interactive and template-based procedures. Algorithmic procedures are also suitable for task (2) of PDM and LOB calculations. Procedures for PDM and LOB have been fully described in the literature (e.g., Hendrickson and Au 1989; Harris 1978; Laramee 1983; Lumsden 1968) and can be readily utilized to perform the computations of initial schedule data (early start, late start, early finish, late finish) of each activity. This initial schedule is calculated based on normal productivity for the resources with no overtime strategy. Often, however, the initial schedule may not be practical since resource productivity and activity duration can be affected by several factors including weather conditions at the scheduled times. The initial schedule also may not meet the project deadline and may exhibit resource conflicts. Thus, tasks (3) and (4) need to be performed in order to update the initial schedule and accordingly the direct and indirect costs. Task (3) addresses the productivity factors that affect the activities durations. These factors include: trade congestion, weather conditions, overtime, and learning curve effect. Usually it is very difficult to address the combined effect of those factors on the activity duration and, yet, a generally agreed-upon model is not available to facilitate such analysis. However, for a simple and quick consideration of practical productivity factors, researchers (e.g., Adrian 1987; Laramee 1983) have proposed monthly productivity factors that can be applied to modify the resource production rates of weather sensitive activities. The factors proposed by Laramee (Table 3.2) were developed to suit Montreal area **weather** conditions and, for simplicity, are used in this study. More refined factors could be obtained using detailed weather data (e.g., daily temperature and relative humidity records) and may incorporate the impact of more than one factor such as congestion of trades and change orders (Moselhi and Nicholas 1990). In the present development, the user can change the default productivity factors to suit other regional climates and the particular job conditions.

Table 3.2: Monthly productivity factors.

Month	Factor
January	0.70
February	0.70
March	0.75
April	0.80
May	0.90
June	1.00
July	1.00
August	1.00
September	0.90
October	0.80
November	0.75
December	0.70

Once practical productivity factors are applied to the different activities according to its initially planned start and finish times, activities duration and direct costs will change accordingly. Further schedule updating, however, may become necessary if the project dead line is not met. This may mandate a cycle of schedule refinement through specifying overtime strategy for some activities, impacting the total project duration and cost. In order to facilitate the process of schedule updating, the schedule has to be clearly and meaningfully presented to the user through a good user interface and enhanced representation graphics, allowing better readability and comprehension of the schedule and its updating requirements.

Resource conflict resolution (task 4) has to be addressed in terms of cost and schedule. In the literature, several heuristics have been used (e.g., minimum total float) to prioritize the activities that require the same resource at the same time and then allocate the resource to the activity with high priority, delaying the others, within their floats, to non-conflicting times. These heuristics are used to ensure that, at any time, the resource availability limits are not exceeded, and also to level the resource demand, reducing the fluctuation in resource demand. These heuristics may resolve the resources conflicts with respect to the scheduled times of the activities, however do not address it in terms of cost. For example, if at any given point the resource conflict can not be resolved and the resource demand is higher than available, additional costs are incurred by acquiring external resources and/or employing other strategies like overtime. Thorough analysis of schedule resource conflict resolution and its cost implications is an extremely difficult task due to the multi-resource, multi-activity, and multi-project nature of construction (Moselhi and Lorterapong 1993).

To facilitate the calculation of the project cost incurred due to resource conflicts, a more simple and faster procedure that suits the decisions at the bidding stage is proposed. The procedure keeps track of the monthly work load and availability limits of owned resources, mainly equipment, in on-going projects. Once the schedule and initial direct costs are developed for a project under bidding, the demand for owned resources is accumulated monthly and compared to the availability limits. If the demand level is higher than the availability limit, then, an additional cost adjustment can be calculated as follows:

$$\text{Cost adjustment (\$)} = (DL - AL) * (E - I) \quad \text{where,}$$

DL = Demand level of an owned equipment (hours).

AL = Availability level (hours).

E = Hourly rate (\$/hr) of the equipment if externally acquired (e.g., rented).

I = Hourly rate of the owned equipment (operational or operational plus depreciation) that is used in the calculation of initial direct costs.

These cost adjustments are then accumulated for all the owned equipments and added to the indirect costs, to be optimally distributed on the different contract items. The procedure is simple and practical, particularly for civil engineering projects with high equipment utilization. Most building projects, however, do not include activities with high equipment utilization and, if any, the work could be subcontracted. However, the simplicity of the algorithm makes it potentially applicable for contractors who would like to utilize their own equipment to reduce cost and provide more competitive, yet realistic, bid prices.

Similar to the cost estimation module, the main nature of the developments needed for the planning and schedule module is algorithm-based. To facilitate planning and schedule calculations, the interface algorithm should facilitate the link among the WBS elements, the activities and resources databases, and the planning and scheduling algorithms. Such algorithms are necessary to incorporate both calendar and working days in order to consider for pre-specified time constraints. After final cost and schedule refinements, various reports pertaining to the project activities, direct costs, schedule, and resource use, can be generated at different management levels. Some of these reports, particularly those pertaining to accumulated resource use, provide a guide for indirect cost estimation.

3.4.4 Risk Assessment Module

The risk assessment module incorporates assessment of the qualitative factors into the cost estimation methodology. Accordingly, optimum markup and the probability of winning the job could be estimated. The tasks needed to be performed in the risk assessment module (see Fig. 3.5) are: user identification of project risks through a brain-storming session, markup estimation using an analogy-based model, algorithmic computations, and assessment of the probability of winning through probabilistic evaluation. The first task is performed by the contractor (user) who, from a list of possible risks, identifies and inputs the types and intensities of those applicable to the project. Based on the identified risks, an automated markup estimation model performs the rest of the tasks.

Based on the literature review of chapter 2, neural networks could provide a practical model for the optimum markup problem. The model should utilize contractors' past bids (successful and unsuccessful) as training data, having the inputs as the qualitative factors pertaining to those projects and the desired outputs as the optimum markup decided in those cases. Such a model generalizes the procedure by which contractors arrive at proper markup decisions. In addition, the model should provide an assessment of the impact of such markup on other important aspects including the actual profitability of the project and its potential for claims and change orders. These aspects may influence bidding decisions and may require contractors to make some corrective or precautionary measures.

Considering the advantages and limitations of the various neural network paradigms (Tables 2.4 and 2.5), the most suitable neural network paradigm selected for the optimum markup estimation problem is the Backpropagation, based on the following criteria:

- High accuracy is desired.
- High interpolative performance is desired.
- Relatively longer training time is accepted since limited number of training examples are expected to be available to train the network. Also, the network is expected to be of small to medium size with moderate hardware requirements for its implementation.
- Incremental learning and real-time performance are not desired.

The development of the neural network model for the markup problem using the backpropagation paradigm is covered in chapters 4, 5 and 6.

3.4.5 Bid Unbalancing Module

Bid preparation frequently constitutes the compilation of all the project costs (direct, indirect, and markup) into unit price(s) assigned to the different contract item(s), to be submitted to the owner as the contractor's bid for the project. The linear programming model of Stark and Mayer (1983), described earlier in chapter 2, is used to optimally unbalance the bid, maximizing the contractors revenues as an objective. The optimization is performed under a set of constraints, including: expected differences in bid quantities, and estimated upper and lower limits for the unit prices. Those constraints are mainly input to the model based on the contractor's expectation and experience. Simple heuristics could be used to enhance the performance of the algorithm, for example, the upper and lower limits for the unit prices of a particular contract item could be set as a function of the actual direct cost and the quantity of such item (e.g., 1.2 and 0.9 of item direct cost/item quantity, respectively). The optimization routine, as such, considers the total project cost (direct + indirect + markup), rather than the individual components, in optimizing the unit prices. To take full benefit of the optimization routine, an additional set of constraints is incorporated into the formulation. The constraints limit the contractor's cash out-of-flow (the difference between the estimated costs and anticipated revenues at any time period) to the monetary value the contractor is able to finance. Such cash flow analysis requires data pertaining to the expected expense curve (S-curve) and the expected income profile, which could be obtained based on the project cost and schedule estimates. The new cash flow constraints could be expressed as follows (using the same notations used in chapter 2):

Given the estimated costs being accumulated monthly as C_t where t (1 to T), and the contractor's monthly financial capability is CF (see Fig. 3.9), then, the set of T - 1 constraints are:

$$
\begin{aligned}
C_{t+1} & - \{ a X_1 \, q(1,1) + \ldots + a X_1 \, q(1,t) \} \\
& - \{ a X_2 \, q(2,1) + \ldots + a X_2 \, q(2,t) \} \\
& - \ldots\ldots\ldots\ldots \\
& - \{ a X_N \, q(N,1) + \ldots + a X_N \, q(N,t) \} \leq CF, \\
& \qquad t = 1,2,\ldots, T-1
\end{aligned}
\qquad [3.1]
$$

3.5 Bid Preparation Methodology

Based on the previous discussion, the components included in each of the system four modules and their connecting databases are illustrated in Fig. 3.10. Also, the proposed step-by-step methodology that effectively incorporates both the

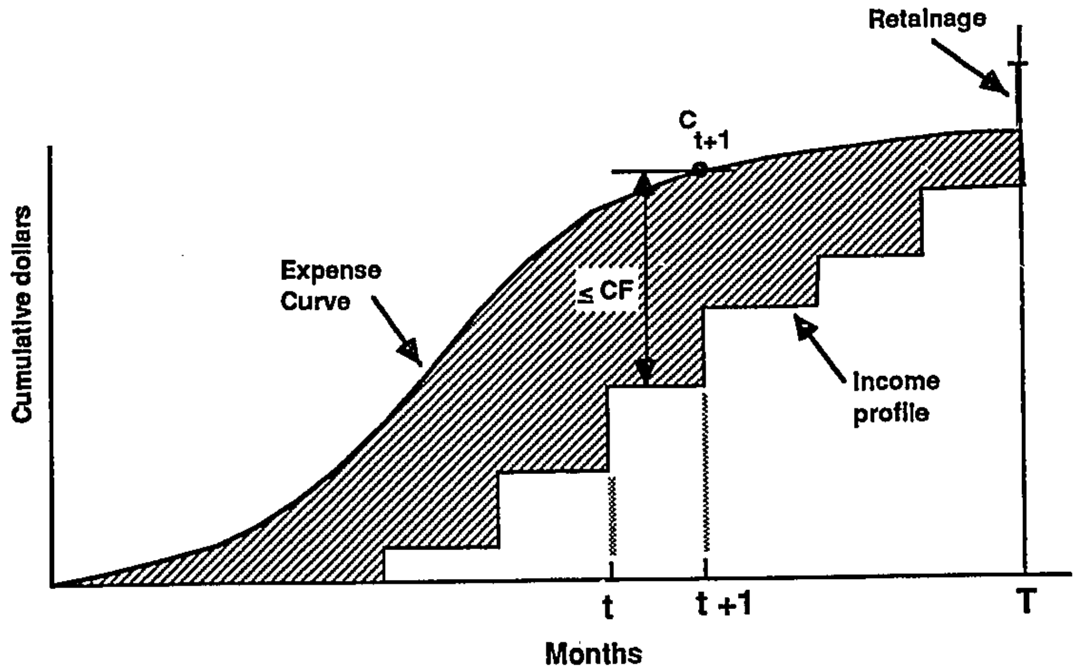


Fig. 3.9: Contractor cash flow curves.

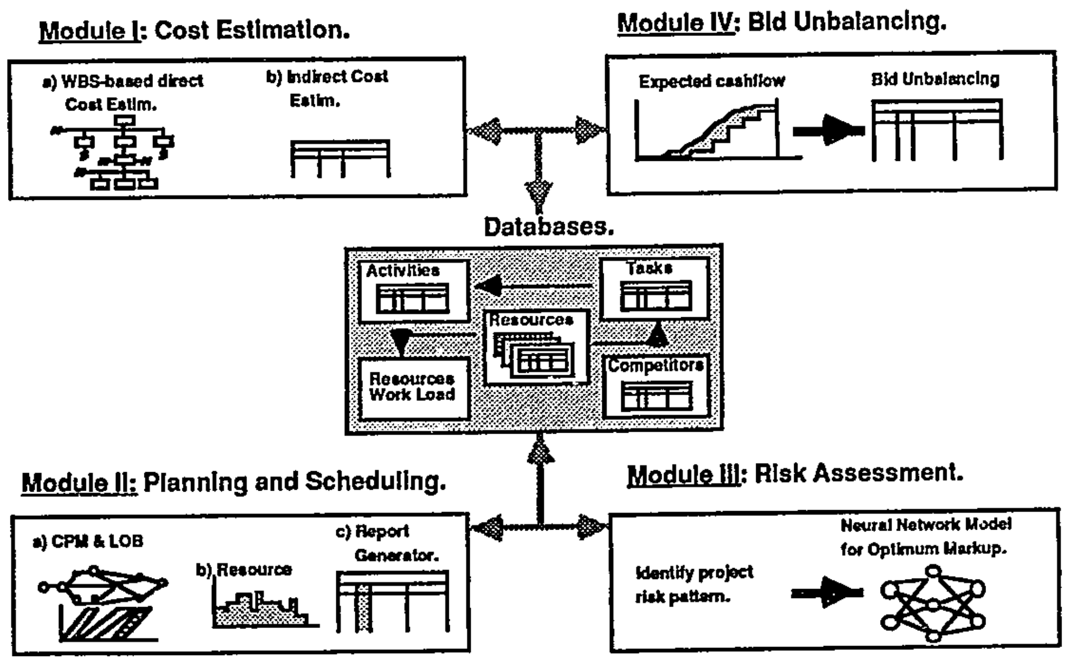


Fig. 3.10: Estimation system modules.

quantitative and qualitative assessments of bid preparation is outlined in Fig. 3.11 and further detailed in Fig. 3.12. The flow diagram of Fig. 3.12 shows, for each step of the methodology, the functions performed and the necessary input/output data links. This diagram facilitates the implementation and the development of the system prototype (discussed later in chapter 7).

In the first step of this methodology, bid documents are received and decision to bid on the job is made. In the next step, work breakdown structure is performed, based on study of the project drawings and specifications. Each element in the WBS is automatically assigned a unique code of accounts and linked to the appropriate database(s) and an appropriate contract item. Direct costs associated with all WBS elements are then calculated based on the activities and tasks included and their resource type, cost, average productivity data from the appropriate databases, and the quantity of work estimated by the user. The output of this step is the labour costs, equipment hours and costs, total materials quantities and costs, subs costs, and total manhours used, in addition to the total of direct costs, for the different WBS levels and contract items.

In the next step, activities are to be scheduled and the cost implications of such a schedule addressed. After initial schedule calculations, practical productivity factors are assigned to the activities to account for expected weather conditions at the scheduled dates, trade congestion, over time, and other productivity-related

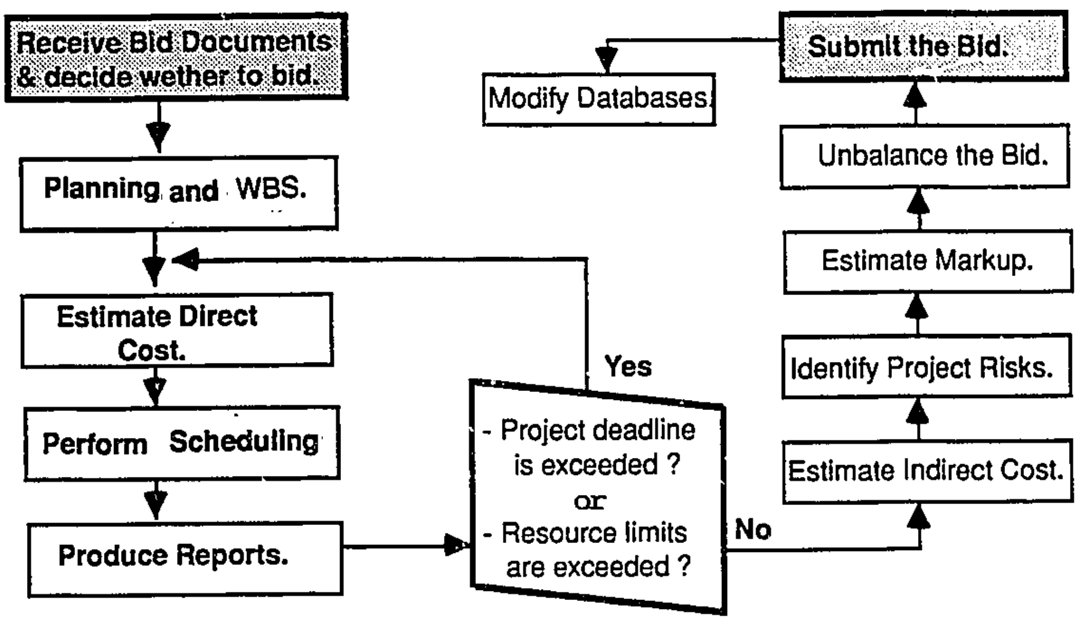


Fig. 3.11: Flow diagram of an integrated methodology for cost estimation and bid preparation.

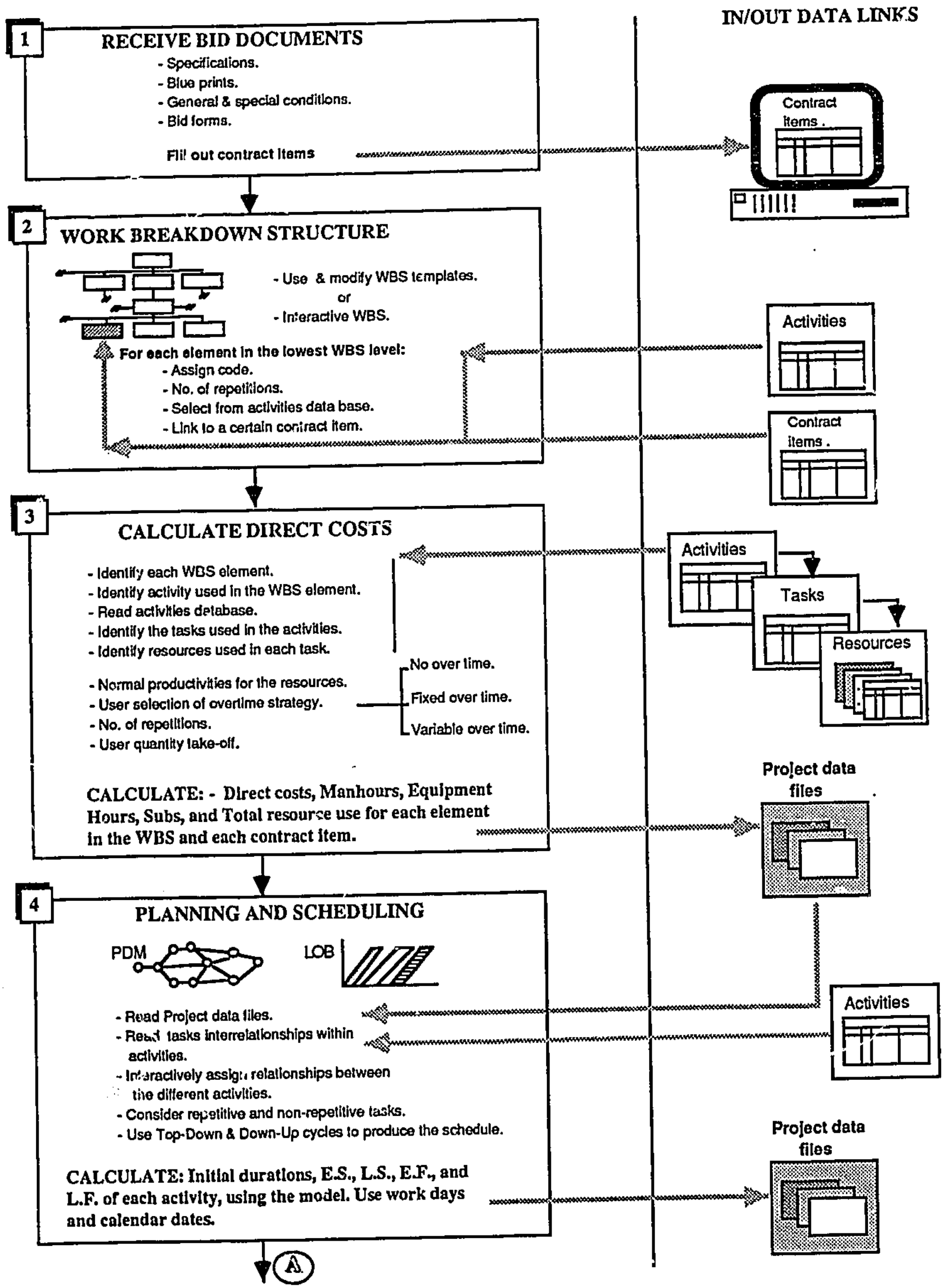


Fig. 3.12: Proposed methodology for estimating in a competitive bidding environment.

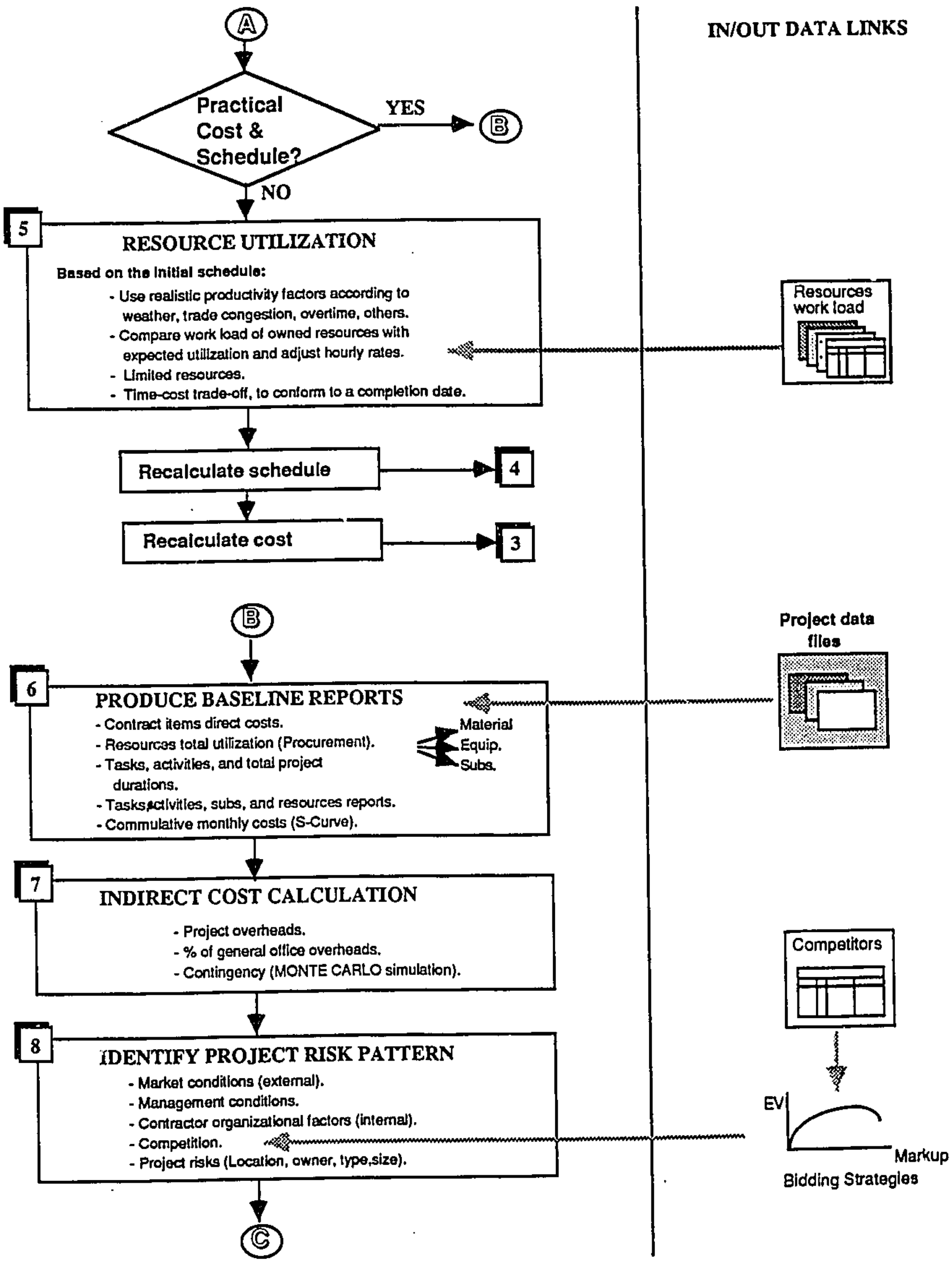


Fig. 3.12 (continued).

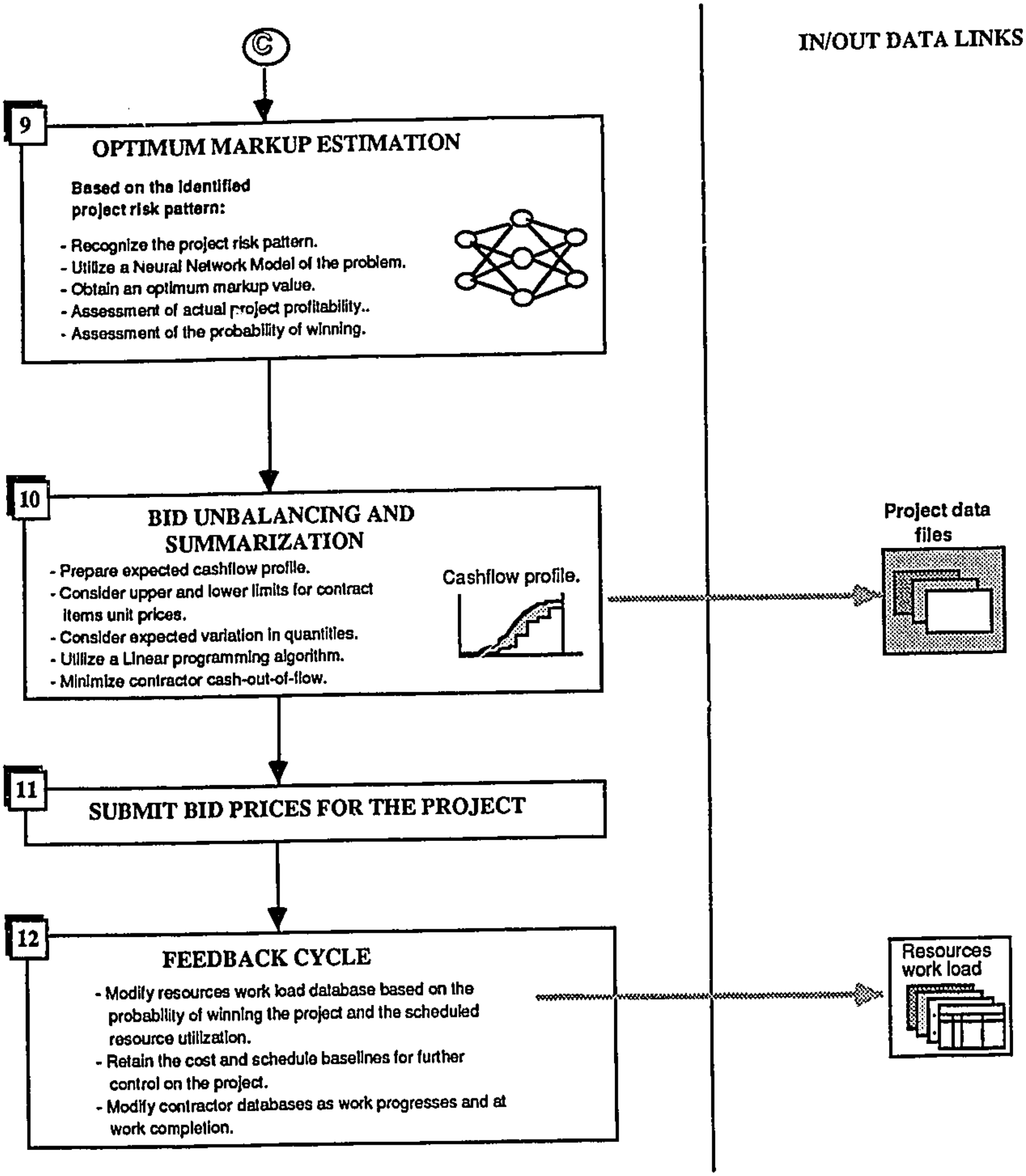


Fig. 3.12 (continued).

factors. Those factors, in addition to the workload of owned resources, are considered through a cycle of cost, schedule, and resource updates. Accordingly, in the next step, reports pertaining to the project activities, direct costs, schedule, and resource use, can be generated at the different management levels. Some of these reports, particularly those pertaining to accumulated resource use, provide a guide for indirect cost estimation which is performed in the next step. These include the project overheads and a part of the firm general overhead, and the total cannot readily be attributed to a certain contract item. A list of possible indirect cost items could be utilized and estimated costs are assigned to those applicable to the project.

The following step is to identify the project expected competition and collect available information about the competitors and any other factors that contribute to a better classification of the project degree of risk including: project size, type, location, ownership, and prevailing market conditions. Accordingly, optimum markup and probability of winning the job are estimated, utilizing a neural network trained to provide a decision aid for the optimum markup problem. Once this is done, the bid could be prepared for submission to the owner. A suitable bid unbalancing procedure can then be utilized, such as that of Stark and Mayer, to provide the final unit prices for the different contract items. A possible strategy to be used with the linear programming algorithm is to utilize some heuristic rules as previously discussed. The last step in the methodology incorporates a feedback cycle to refine the information contained in the company databases and the neural network model, based on actual project outcomes (win/loose the job, actual profit, and actual productivity levels).

3.6 Conclusion

This chapter proposed a methodology for an integrated cost estimation environment, utilizing available tools (algorithms, database management systems, and AI-based techniques) that can benefit form current industry practice and provide an adequate decision aid during the bid preparation process. A procedure for developing optimum problem-solving strategies is outlined and applied on the bid preparation problem based on a set of identified system requirements. The system consists of four main modules: (1) an algorithmic work breakdown structure-based cost estimation module; (2) an algorithmic planning and scheduling module; (3) an Artificial Intelligence-based Neural Network module for risk assessment and optimum markup estimation; and (4) a combined heuristic and algorithmic bid unbalancing module. These individual modules are integrated through a database management system, incorporating a unified code of accounts, forming a comprehensive management information system (MIS).

The first two modules cover the quantitative aspects of cost estimation, providing a feed back cycle to refine estimated costs and schedules, accounting for resource constraints. The risk assessment module utilizes a neural network designed to estimate an optimum markup value and predicts the probability of winning the job at such level of profit, in response to the project risk pattern. Among the several neural network architectures available, the backpropagation paradigm has been identified as the most suitable environment for modelling the markup estimation problem. The system optimally unbalances the final bid, in an effort to improve the contractor's cash flow while maintaining his competitiveness. The hybrid system facilitates decision making, allowing detailed pre-bid estimates of costs and durations to be performed with minimal redundancy and in a timely manner, improving the efficiency of the bid preparation process. This chapter emphasized the conceptual and design stages of the system development, establishing a structured methodology for cost estimation and bid preparation in a competitive environment. A prototype of the integrated system is presented and discussed in chapter 7.

Chapter 4

SETTING UP THE BACKPROPAGATION ENVIRONMENT

4.1 Introduction

In this chapter, the basic mathematics of neural networks that led to the formulation of the backpropagation paradigm (the paradigm suggested for the markup estimation problem) are introduced. An elaborate description of the backpropagation and the generalized delta rule algorithm for its training are provided. The capabilities of the paradigm as well as the various problems that face the development of its practical applications are described along with several remedial techniques and heuristics. An application development methodology that properly sets up the backpropagation environment to suit a problem's solution requirements is proposed. The methodology utilizes common techniques and heuristics used to overcome backpropagation problems and guide the development of practical applications with less effort. Such a methodology can readily be utilized for developing the markup estimation model.

4.2 Basic Mathematics of Neural Networks

The improved understanding of the human neural systems, during the 1940s, has enabled researchers to develop theories on and simulations to the complicated functioning of biological systems. Artificial neural networks have therefore been developed with a structure modeled after the organization of the human brain. The similarity is actually slight, yet even this modest emulation of the brain has yielded impressive results. As demonstrated by early neural network models (Pitts and McCulloch 1947), a single neuron (Processing element; P.E.) can perform certain simple pattern detection functions. However, the power of neural computation comes from connecting P.E.s into networks, as evidenced by the human biological system. The simplest network is a group of P.E.s arranged in a layer as shown on the right side of Fig. 4.1. The circular nodes on the left serve only to distribute the input vector X; performing no computation and hence are considered an input buffer rather than a layer. The set of inputs X has each of its elements connected to each P.E. through a separate weight W.

In matrix notations, it is convenient to consider all the network weights as elements of a matrix W. The dimensions of the matrix are m rows by n columns, where m is the number of inputs and n is the number of P.E.s (Producing the output). Each P.E., in its simplest form, outputs a weighted sum of the incoming inputs, as shown in Eq. 4.1.

$$\underset{1xn}{O^T} = \underset{1xm}{X^T} * \underset{mxn}{W} \qquad [4.1]$$

where, X^T and O^T are the transpose of the input and output vectors. It should be noted here that the P.E.s, in this case, have a linear activation function (Fig. 4.1). In the more general case when more than one input vector is presented to the

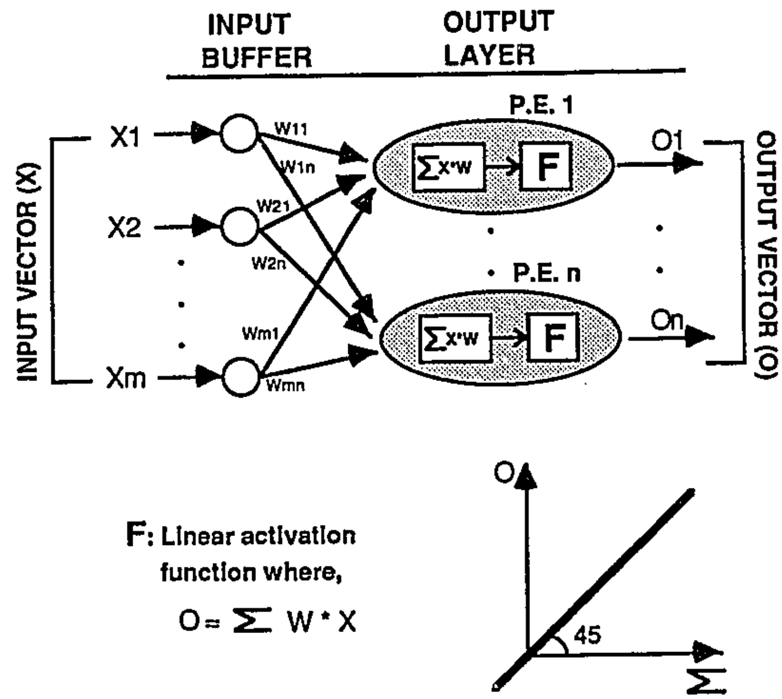


Fig. 4.1: A simple one-layer neural network with linear activation function for the P.E.s.

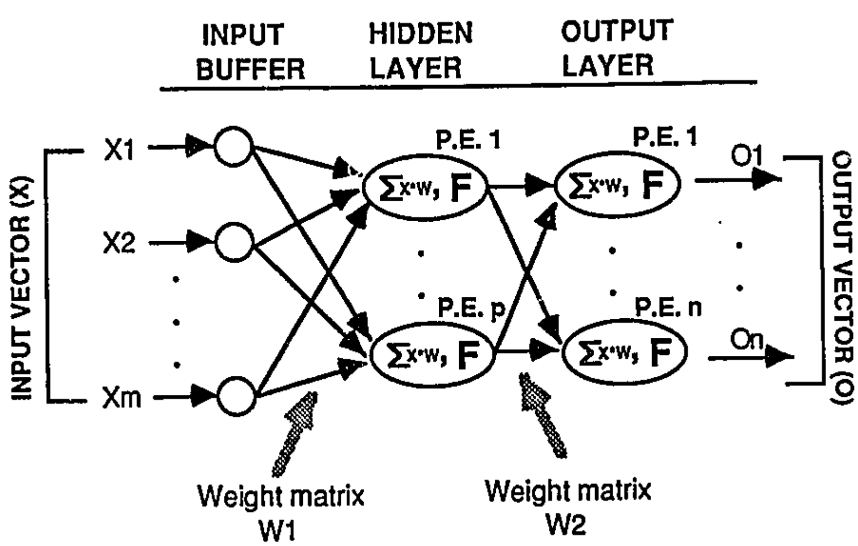


Fig. 4.2: Two-layer neural network with linear activation function for the P.E.s.

network, Eq. 4.1 can be modified as:

$$\underset{Pxn}{O^T} = \underset{Pxm}{X^T} * \underset{mxn}{W} \quad [4.2]$$

where, P is the number of input vectors used.

A neural network works well if it can learn the association between the set of inputs *X*s and another set of data called desired outputs *D*s, similar to a human expert association between a set of causes and their corresponding conclusions or outcomes. Normally, the domain knowledge can be partially presented to a neural network in terms of a number of training examples P consisting of *X*s and their associated desired outputs *D*s. A trained network is one having a weight matrix W determined so that when the network is presented with the set of inputs *X*s, the outputs produced (*O*s) are very close to the desired outputs *D*s. If this is the case, the network is said to be knowledgeable about the P examples, with the weight matrix encoding the generalized domain knowledge and can be used to estimate outputs for data outside the partial domain used in training.

Using Eq. 4.2, with linear activation function, the weight matrix W of the network can be calculated, rather than determined through an iterative training process, given the set of inputs *X*s and set of outputs *D*s, as follows:

$$\underset{mxn}{W} = \underset{mxP}{\left(X^T\right)^{-1}} * \underset{Pxn}{D^T} \qquad [4.3]$$

However, the weight matrix W cannot be directly calculated based on Eq. 4.3 if the X^T matrix is singular or in case $P \neq m$ (i.e., matrix is not square). Such a problem in establishing the network weights still persists in case of multi-layer networks (Fig. 4.2) having linear activation function for the P.E.s. In the multi-layer case, the output of a layer consists of multiplying the input vector by the first weight matrix, and then, if there is no nonlinear activation function, multiplying the resulting vector by the second weight matrix. This may be expressed as:

$$O = (X * W_1) * W_2 = X * (W_1 * W_2) \qquad [4.4]$$

This shows that a two-layer network is exactly equivalent to a single layer having a weight matrix, equal to the product of the two-layers' weight matrices. Non linear activation functions have, therefore, been reported as vital to the expansion of the network's capability beyond that of the single-layer network (Wasserman 1989), despite the lack of a direct mathematical formula for the determination of "calculated" network weights. Hence, network training algorithms have evolved to establish the network weights in an iterative manner. The training process is an incremental weight variations mathematically derived to gradually reduce the error between the network own outputs and the desired outputs of a set of training examples to some specified acceptable limits.

In the early days of artificial Neural Networks, McCulloch and Pitts (1943) presented network paradigms called "PERCEPTRONS", which are single-layer binary networks, having a threshold logic unit (TLU; a discrete step function - Fig. 4.3) and trained based on the perceptron training algorithm (Rosenblatt 1962). The perceptron training method can be summarized as follows:

1. Initialize the weight matrix W with random numbers.
2. Apply an input pattern X and calculate the output O

   $$O = \Sigma \; X * W$$

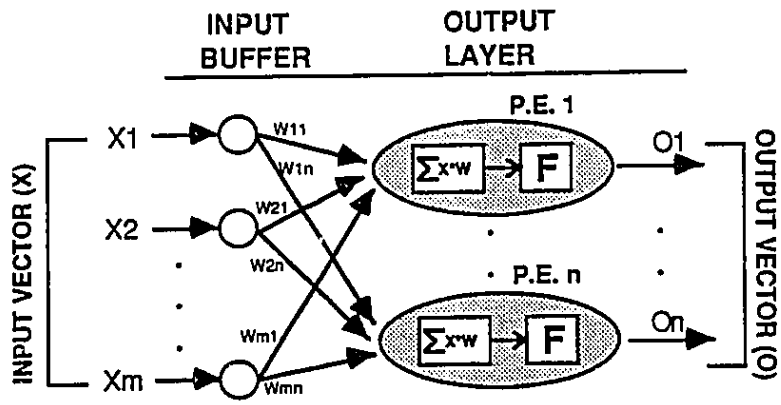
3. a) If O = desired output then go to 2;

   b) If O $\neq$ desired output and O = 0 (i.e., network is under estimating) then, $W_{new} = W_{old} + X$. (i.e., add each input to its corresponding weight).

   c) If O $\neq$ desired output and O = 1 (i.e. network is over estimating) then, $W_{new} = W_{old} - X$ (i.e. subtract each input from its corresponding weight).

4. Go to step 2.

In the perceptron training algorithm, the weight change necessary to correct a misclassification is equal to the input value X (steps 3.b and 3.c). This is sufficient because, for binary networks, misclassifications and necessary corrections (X) are always 0 or 1. However, the algorithm is not appropriate for training networks with continuous inputs and outputs where misclassifications and input values are real numbers that are not equal.



F: **Threshold Logic Unit (TLU) where,**

$O = 1$ if $\sum W * X > T$
or
$O = 0$ if $\sum W * X < T$

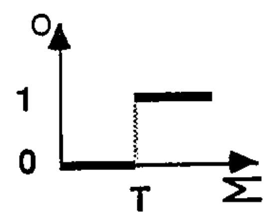


Fig. 4.3: A one-layer perceptron with threshold logic unit (TLU) for the P.E.s.

An important generalization of the perception training algorithm, called the delta rule, extends this technique to continuous inputs and outputs (Widrow-Hoff 1960). The algorithm introduces a term δ, which is the difference between the desired output D and the actual output O:

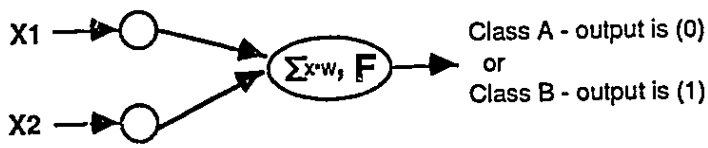
$$\delta = D - O \tag{4.5}$$

In the delta rule algorithm, step 3.a corresponds to $\delta = 0$; step 3.b corresponds to $\delta > 0$ (network is under estimating); while step 3.C corresponds to $\delta < 0$. Accordingly, the weight change formulas could be replaced by:

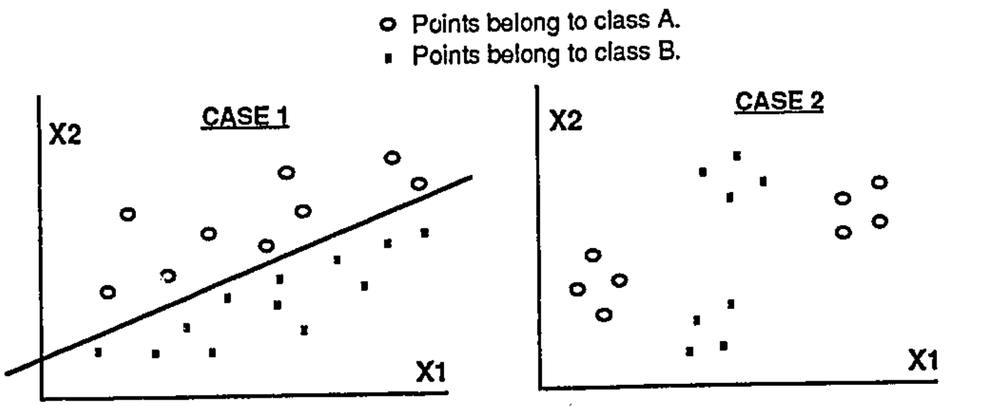
$$W_{new} = W_{old} + \Delta W \qquad \text{where,} \quad \Delta W = \eta * \delta * X \tag{4.6}$$

where, η is a "learning rate" coefficient ($0 < \eta < 1$) which is introduced to allow control of the average size of weight changes. Therefore, the simple delta rule expression states that the change in the weight matrix should be proportional to δ and to the input pattern.

In essence, the single-layer perception training algorithm learns to separate the input patterns into two classes; one with output higher than the threshold and hence producing a 1, and the other with output lower than the threshold and hence producing a zero. This, imposes a restriction that the problem and accordingly the training data must be linearly separable (Fig. 4.4 - case 1). Although the proof of the perception learning theorem (Rosenblatt 1962) demonstrated that a



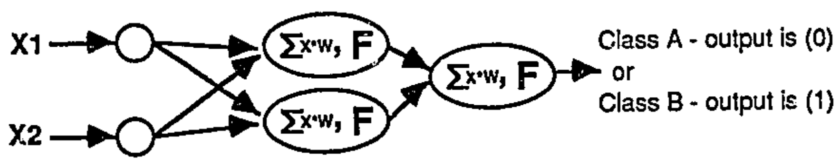
A one-layer perceptron for simple classification
between points of class A and points of class B.



A linearly separable problem space.
One line can separate the two classes.

A linearly non-separable problem space.
No one line can separate the two classes.

Fig. 4.4: Linear separability problem.



A two-layer perceptron for solving a linearly
non-separable classification problem.

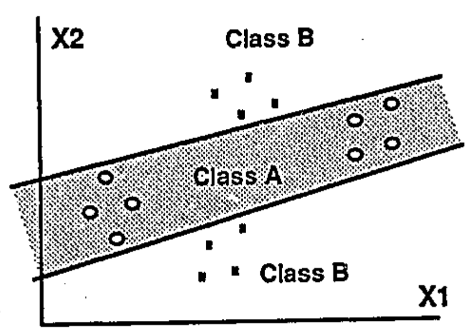
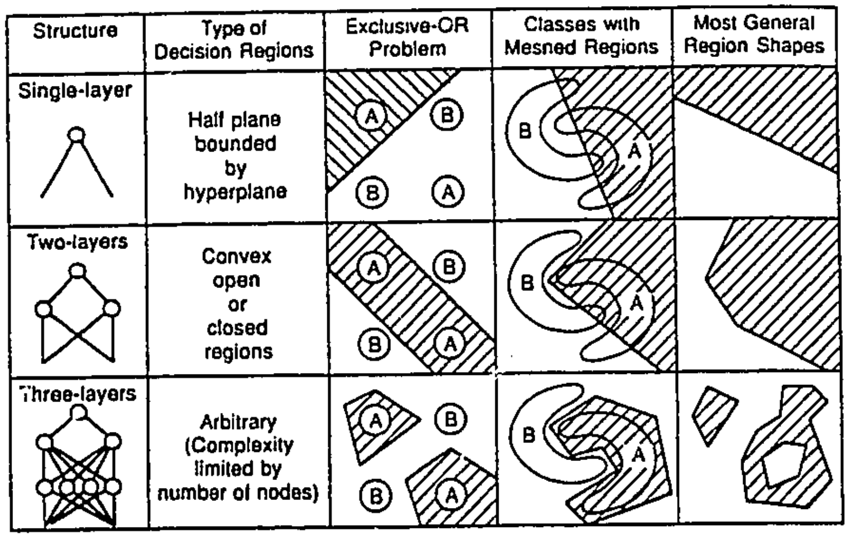


Fig. 4.5: Function of two-layer perceptron.

perception could learn anything it could represent, the single-layer perception is seriously limited in its representational ability (Minsky and Papert 1969). Fig. 4.4 - case 2, shows a linearly non-separable data which a perception cannot represent, no matter what values are assigned to the weights and the threshold and how they are adjusted.

By the late 1960s, the linear separability problem was well understood. It was also known that this serious representational limitation of single-layer networks could be overcome by adding more layers. Fig. 4.5 shows how a 2-layer perception can provide a solution to the linearly non-separable problem of Fig. 4.4 - case 2. As shown in Fig. 4.5, the data space is divided into regions that separate the different output classes. A three layer network (i.e., 2 hidden layers) is still more general; its classification capabilities is limited only by the number of P.E.s and weights. Fig. 4.6 (Pao 1989) shows the role of hidden nodes and layers in multi-layer perceptions. Despite early recognition of the power of multi-layer networks with nonlinear transfer function, for many years there was no theoretically sound training algorithm for adjusting their weights.

In view of perceptron limitations, backpropagation NN paradigm (Rumelhart et al. 1986) have evolved to provide solutions to several problems facing the effective and systematic training of multi-layer perceptions, including:



A simple view of the role of hidden units in multilayer perceptrons, for two inputs and the hard-limiting case. (Lippman, R.P., 1987. An introduction to computing with neural nets, *IEEE ASSP Magazine*, Vol. 4, p. 14, ©1987, IEEE. Adapted from and reprinted with permission.)

Fig. 4.6: Role of hidden layers in multi-layer perceptrons.

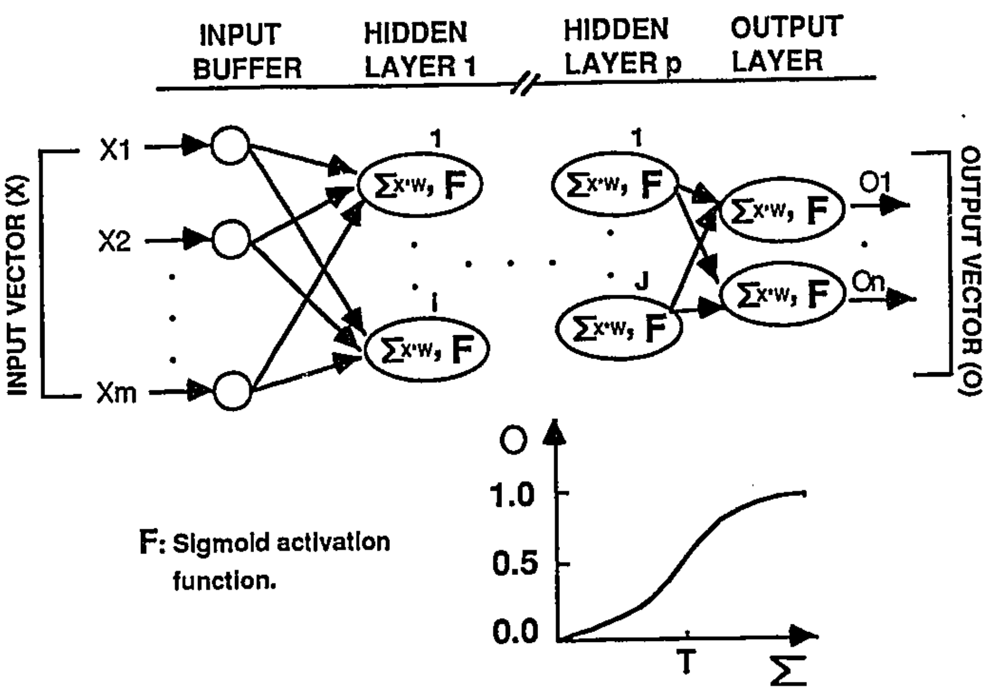


Fig. 4.7: A multi-layer, feedforward, Backpropagation network.

- how the weights connected to the hidden layers can be adjusted given the fact that hidden layers' desired outputs are not known ?
- what type of nonlinear activation junction should be used for multi-layer networks ?
- how to ensure the convergence of a training algorithm for multi-layer networks ?

4.3 Formulation of Backpropagation

A clear and concise description of the backpropagation algorithm has been introduced by Rumelhart, Hinton, and Williams (1986), and others had arrived at similar formulations (Parker 1982; Werbos 1974). Overcoming the limitations of single-layer perceptrons with linear activation functions, backpropagation networks is made up of sets of P.E.s having a nonlinear continuous activation functions, arranged in multi-layers (Fig. 4.7). Each P.E., except those at the input buffer, multiplies each of its input by a weight, and the products are summed. This summation of Products is termed NET, and must be calculated for each P.E. in the network. After NET is calculated, an activation function F is applied to modify it, thereby producing the signal OUT. Fig. 4.7 shows the nonlinear activation function usually used for backpropagation. The function called a "Sigmoid", is expressed as:

$$O_j = F(Net_j) = \frac{1}{1 + e^{-[Net_j - \theta_j] / \theta_o}} \qquad [4.7]$$

where, θ_i is the i's P.E. threshold or Bias, and θ_o is the Sigmoid shape term (Fig. 4.8). The Sigmoid functions is desirable for backpropagation networks for a number of reasons (Pao 1989):

- It meets the backpropagation requirements as a function that is: a) continuous and bounded from both sides (outputs can have any value in the range of 0 to 1); b) everywhere monotonically increasing (i.e., S-shaped); and c) everywhere differentiable.
- It has a simple derivative, expressed by Eq. 4.8.

$$\frac{\partial 0}{\partial Net} = 0\,(1 - 0) \qquad \textit{this is derived as follows:}$$

$$\textit{Taking} \qquad I = (\,Net_i - \theta_i\,)\,/\,\theta_0 \qquad \textit{of Eq. 7} \quad \textit{then,}$$

$$\begin{aligned} \frac{\partial O}{\partial net} &= \frac{e^{-I}\cdot}{(1+e^{-I})^2} = \frac{((1 + e^{-I}) - 1)}{(1 + e^{-I})^2} \\ &= \frac{1}{(1 + e^{-I})}\,(1 - \frac{1}{(1 + e^{-I})}\,) \\ &= 0\,(1 - 0) \end{aligned} \tag{4.8}$$

- It provides a form of automatic gain control. For small signals (closer to the origin) the slope of the activation functions is steep, producing high gain. As the magnitude of the signal becomes greater, the gain decreases. In this way, large signals can be accommodated by the network without saturation, while small signals are allowed to pass through without excessive attenuation (Wasserman 1989).

As expressed in Eq. 4.7, each P.E. in a backpropagation network is modified with a trainable bias (Fig. 4.8 - Pao 1989). This affects the origin as well as the shape of the activation function. The bias term works as a horizontal shift for the origin of the transfer function to suit the magnitude of signals incoming to the P.E., thereby permitting more rapid convergence of the training process (Wasserman 1989). The P.E.s biases are usually estimated as trainable weights for an additional input node attached to all network P.E.s, having a constant value of ($X = +1$). As such, taking a bias into account, the simplified P.E. in Fig. 3.7 can be more generally represented as shown in Fig. 4.9.

Using P.E.s with trainable biases, Rumelhart et al. (1986) formulated a training algorithm called "Generalized delta rule" for training their multi-layer feed forward networks. Similar to the "delta rule" algorithm described earlier, the objective of training the network is to determine a unique set of network weights and biases that enable the network to produce outputs *O*s that match the set of desired outputs *D*s pertaining to the training examples when fed only with the respective inputs *X*s. Having the P training examples made up of input vectors (*X*s) paired with target vectors (*D*s), the examples are presented to the network in each training iteration and the network is let to produce its own outputs (*O*s). In general, these outputs (*O*s) will not be the same as the targeted or desired values (*D*s). The error between D and O is termed δ. For each training example, the square of the error is:

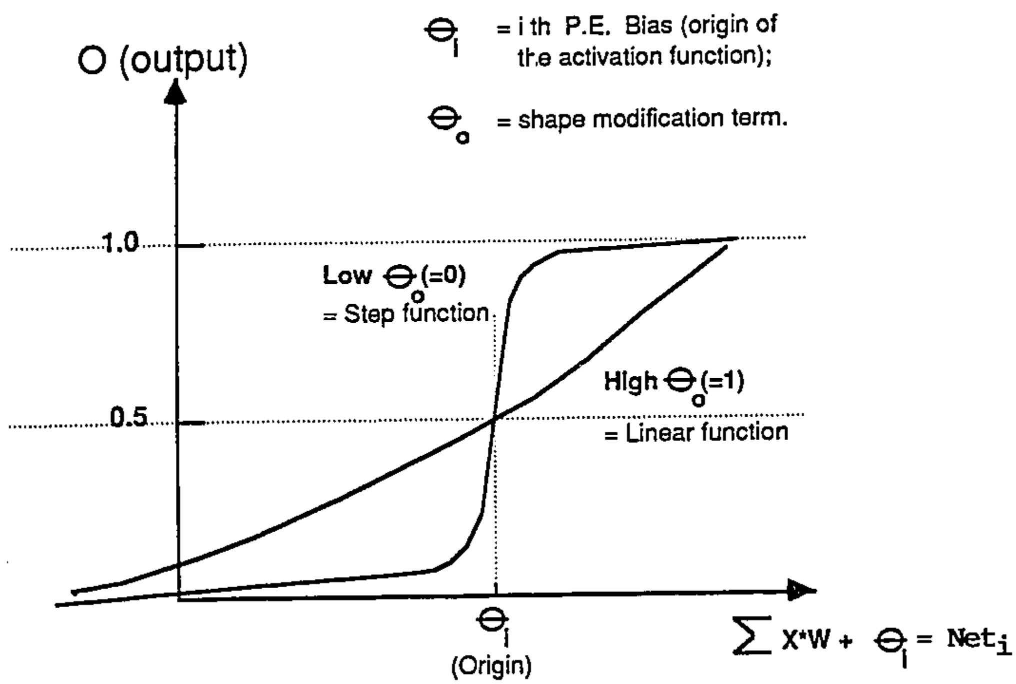


Fig. 4.8: Sigmoid activation function with bias and shape modification term.

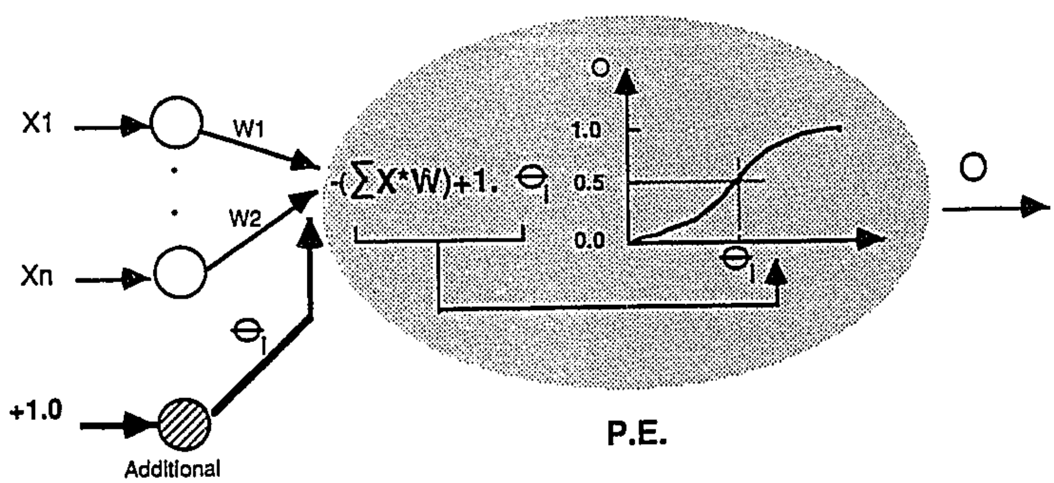


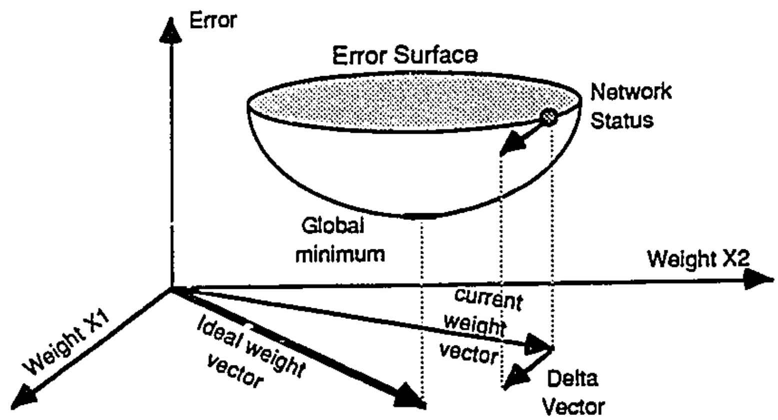
Fig. 4.9: Simple P.E. in a backpropagation network having a trainable bias.

$$E = \frac{1}{2} \sum_{i=1}^{n} (D_i - 0_i)^2 \qquad [4.9]$$

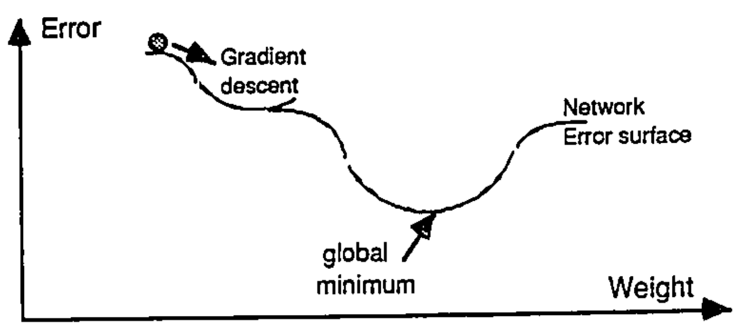
where, n is the number of P.E.s. in output layer. Also, the average squared errors over all the training examples is:

$$E = \frac{1}{2P} \sum_{J=1}^{P} \sum_{i=1}^{n} (D_{iJ} - 0_{iJ})^2 \qquad [4.10]$$

where, the factor of one-half is inserted for mathematical convenience (Pao 1989). In the generalized delta rule algorithm, weights are adjusted by an incremental variation proportional to the error (δ), formulated in a manner that reduces the error E in Eq. 4.10 as rapidly as possible. The problem that faces training here is how the weights connected to the hidden layers can be adjusted given the fact that hidden layers' desired outputs are not known (i.e., δs can not be directly calculated). Rumelhart et al. have overcome this problem and viewed training as an optimization process and thus used a "Hill-Climbing" or gradient descent method to minimize the error (E) in a guaranteed-to-converge manner. A gradient search can achieve improved values for the weights and biases by taking incremental change (ΔW) proportional to $-\frac{\partial E}{\partial W}$ that is: $\Delta W = -\eta \frac{\partial E}{\partial W}$ where, (η) is the learning rate coefficient. $\frac{\partial E}{\partial W}$ can be evaluated using the chain rule to express the partial derivative in terms of two partial derivatives (Pao 1989), as follows:



(a) Network error surface for a simple problem.



(b) Error surface topology for multi-dimentional complicated problem.

Fig. 4.10: Error surface and weight change in backpropagation networks.
(adapted from Caudill 1988)

$$\Delta W = -\eta \frac{\partial E}{\partial W} = -\eta \left(\frac{\partial E}{\partial Net} * \frac{\partial Net}{\partial W} \right) \qquad [4.11]$$

Since Net = Σ W * X, and X (inputs) = outputs of previous layer's P.E.s, then, the right hand partial derivative becomes:

$$\frac{\partial Net}{\partial W} = \frac{\partial \sum W * X}{\partial W} = X$$

Therefore, Eq. 4.11 becomes:

$$\Delta W = \eta \left(- \frac{\partial E}{\partial Net} \right) * X \qquad [4.12]$$

Then, defining $\delta = \left(- \frac{\partial E}{\partial Net} \right)$ and using the chain rule again:

$$\delta = - \frac{\partial E}{\partial Net} = - \frac{\partial E}{\partial O} * \frac{\partial O}{\partial Net} \qquad [4.13]$$

The left hand partial derivative formed by the chain rule is the partial derivative of Eq. 4.9 with respect to O, that is:

$$\frac{\partial E}{\partial O} = -(D - O)$$

The right hand partial derivative of Eq. 4.13 can also be evaluated as:

$$\frac{\partial O}{\partial Net} = F^{I}(Net) = O(1 - O)$$

which is the partial derivative of the Sigmoid activation function as previously formulated. Having the partial derivatives been evaluated, then, Eq. 4.13 can be re-written as:

$$\delta = (D - O) * O(1 - O) \tag{4.14}$$

Consequently Eq. 4.12 for calculating the incremental weight change can be evaluated as:

$$\begin{aligned} \Delta W &= \eta * \delta * X \\ &= \eta * (D - O) . O(1 - O) * X \end{aligned} \tag{4.15}$$

It should be noted that the first expression in Eq. 4.15 is similar to the delta rule of Eq. 4.6. The generalized delta rule, however, uses the chain rule to evaluate δ in terms of partial derivatives. For the output layer P.E.s, δ can be directly evaluated using Eq. 4.14 since the desired output vectors (D) are known. However, hidden layer P.E.s have no desired outputs to enable the application of Eq. 4.14. This has been a major problem in training multi-layer networks until backpropagation provided a workable algorithm. For the hidden layers, δ in Eq. 4.14 must be generated without the benefit of having a target vector. Backpropagation accomplishes this by propagating the errors (δs) calculated at the output layer back through the network, layer by layer, generating a δ value for each P.E. in the hidden layers. These (δs) are used, in turn, to adjust the weight of the hidden layers. For the <u>last</u> hidden layer, the value of δ can be derived as follows:

$$\delta = -\frac{\partial E}{\partial O} \cdot \frac{\partial O}{\partial Net} = -\frac{\partial E}{\partial O} \cdot F^{I}(Net)$$ from Eq. [4.13]

According to Rumelhart et al. (1986), the factor $-\frac{\partial E}{\partial O}$ can be indirectly calculated as:

$$-\frac{\partial E}{\partial O} = \sum \delta * W$$

where δ and W correspond to the output layer. Then, δ becomes:

$$\underset{\substack{\textit{last}\\ \textit{hidden layer}}}{\delta} = (\underset{\textit{output layer}}{\sum \delta * W}) \,.\, F^{I}(net)$$

$$= (\underset{\textit{output layer}}{\sum \delta * W}) \,.\, O\,(1 - O) \qquad [4.16]$$

As such, this formula represents the deltas at an internal P.E. in terms of the deltas at an upper layer. Thus, starting at the highest layer (output layer), δs can be evaluated using Eq. 4.14, and we can propagate the "errors" backward to lower layers.

4.4 Training Procedure

Keeping Fig. 4.7 in mind, the procedure for training a multi-layer backpropagation network can be summarized in the following step:

1. A set of training examples is prepared; input vectors (Xs) and associated desired outputs (Ds - scaled between 0 and 1).

2. Network weights (including biases) are initialized to small random numbers (usually between -0.5 and 0.5).
3. A counter for the number of training cycles is incremented.
4. A training example is selected from the training set, applying the input vector to the network input buffer.
5. Network forward calculations are performed on a layer-by-layer basis using Eq. 4.7. The outputs of the P.E.s in a layer are used as inputs to the succeeding layer. The output layer then produces the network output vector (Os).
6. Output layer deltas (δs) are calculated using Eq. 4.14.
7. Network backward calculations are performed at the output layer. The weight connections feeding the output layer are adjusted by a value (ΔW) calculated using Eq. 4.15. The new weights become: $W_{new} = W_{old} + \Delta W$.
8. Network backward calculations are performed for the hidden layers. Starting at the last hidden layer (the layer just before the output layer), values for the P.E.s' deltas (δs) are calculated using Eq. 4.16. With these δs in hand, the weight connections feeding a hidden layer are adjusted similar to that of step 7. Backward calculations are then continued moving to on earlier hidden layer, until the weights feeding the first hidden layer are adjusted.
9. Steps 4 through 8 are repeated for all the training examples and the

square of the errors with respect to all outputs and all examples are added. The average mean squared error of the network E is then calculated using Eq. 4.10. This step completes one training cycle (iteration).

10. If the error calculated in steps 9 does not satisfy the set targeted error (specified), the procedure branches to step 3, starting a new training cycle.

The essence of backpropagation is therefore a minimization of the mean squared error by moving down the gradient of the error curve (Caudill 1988). In a simple system, the error curve is a smooth paraboloid, or bowel-shaped curve. In this case, the network is guaranteed eventually to get to the bottom of the bowel; no bumps or detours exist to trap the network, see Fig. 4.10(a). In the more general case, however, the network is not a simple one-dimensional system, and the error curve is a highly complex, multi-dimensional, more-or-less bowel-shaped curve, see Fig. 4.10(b), that can have all kinds of bumps, valleys, and hills the network must negotiate before finding its lowest point (the minimum mean squared error position).

In the literature, backpropagation technique for training multi-layer feed forward neural networks has been received with different, but all correct, interpretations. According to Sontag and Sussmann (1991) backpropagation is based on: (a) proposing an error function that penalizes misclassifications and then, (b) attempting to minimize this function using a gradient descent method (the name backpropagation arises from the use of the chain rule to compute partial derivatives recursively through the network layers). Also, according to Rigler et al. (1991), backpropagation technique is defined as one form of a nonlinear optimization problem. In essence, it is a very ingenious method of computing the elements of the gradient of a least squares objective function that assigns a value to the squared error measure of performance of a neural network. The latter definition well relates backpropagation learning technique to other standard statistical methods, namely nonlinear least-squares regression analysis. White (1989) has discussed this view and proved that the powerful approximation capability at multi-layer feed forward networks enable them to be viewed as an alternative statistical approach to solving least-squares problem. In addition to the definitions provided above for the backpropagation technique, the developers (Rumelhart et al.) and other researcher like Werbos (1988) have viewed it as a broad and general theory of intelligence that contributes to our understanding of parallel computing and the complicated functioning of the human brain.

4.5 Pros and Cons of Backpropagation

Backpropagation networks are reported as capable of providing accurate approximation to any implicit function or relationship that links the inputs of the training examples to their desired outputs (White 1989). This is provided that a proper network topology is used. Due to this "universal approximation" property

(Iris and Miyake 1988; Hornik 1991), multi-layer feed forward networks are useful for a host of practical applications. Recent experiments with backpropagation have depicted its suitability to a wide variety of applications, including machine recognition of hand-written English words (e.g., Burr 1987) and freeway incident detection (Cheu et al. 1991).

Despite the capabilities of backpropagation, it suffers from several problems that need to be resolved in order to properly design and implement practical NN applications. These problems can be outlined as:

1) Ill-defined knowledge representation and problem structuring.
2) Slow training and sensitivity to the initial set of network weights.
3) Training can be trapped in local minima or paralyse.
4) Non-guided design of an optimal network configuration for adequate generalization.
5) Difficult interpretation of the network weights (Black Box effect).

Each of these points are discussed and their impact on network performance is described. Common heuristics and techniques used to overcome backpropagation problems are outlined along with areas of potential improvements to the paradigm.

4.5.1 Knowledge representation and problem structuring

Traditionally, neural network research has focused more on learning algorithms and less on knowledge representation. This is because learning algorithms are general, simple, and mathematically interesting. Knowledge representation, on the other hand, tend to be problem-specific, ad hoc, complex, and much less mathematically tractable (Anderson 1990). The importance of representation to intelligent problem-solving performance, according to Anderson, stems from the observation that many of the human thought processes are based on superb representations of sensory and perceptual data combined with a reasonably good memory. The learning mechanisms, conversely, are frequently of less importance.

As opposed to expert systems where the knowledge is represented, generally, in a unified and consistent form of IF..THEN rules, neural networks' knowledge representation and problem structuring are versatile and ill-defined tasks, with no hard and fast rules that can be used on any given problem (Cardoso 1991). Choosing an appropriate representation can change a difficult problem into a simple one and may make the difference between solving the problem or not. In most cases, practice and intuition, coupled with a trial and error process may define a proper type of representation.

The method by which knowledge is presented to a neural network can be illustrated as shown in Fig. 4.11. Transformation and scaling of data are simple techniques that are widely used with the backpropagation paradigm. Fig. 4.11 (Moselhi et al. 1991a) shows how input and output patterns of an example

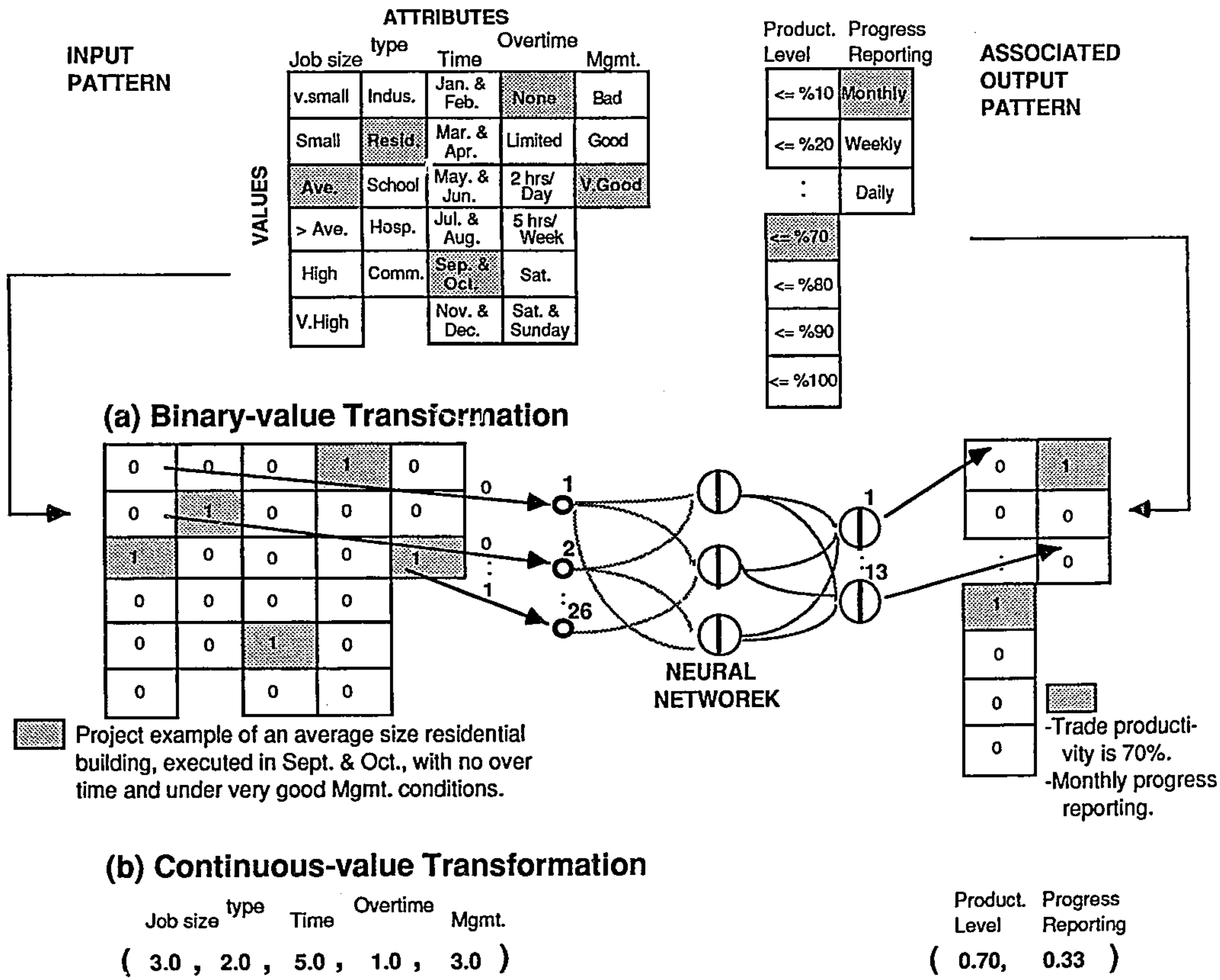


Fig. 4.11 :Formalization of Input and Output patterns into input and output vectors readable by the neural network.

problem, for estimating the productivity level of a certain trade, could be transformed to suit neural network representation. The anticipated productivity level of a certain building trade is usually estimated by experts based on several factors, including job size, building type, date of execution, overtime, and management conditions. Like human experts, neural networks use these factors (problem attributes) as an input to form the project pattern needed to perform the decision-making process. The output of the decision process represents the trade productivity level and the frequency of progress reporting. As shown in Fig. 4.11, a pattern is defined by a grid consisting of a set of attributes, arranged in columns, and the possible values (numeric or non-numeric) arranged in rows. The pattern need not be rectangular in shape, i.e., different attributes may have different numbers of possible values, and attributes may be arranged either in rows or in columns.

Fig. 4.11 shows two main methods of transforming the problem input and output into a form suitable for neural network representation. In a binary-value transformation, a project can be described by assigning ones for the applicable attributes and zeros elsewhere on the grid, as shown in Fig. 4.11(a). A binary vector is then constructed from all the zeros and ones in their sequence on the grid and fed to the input buffer of the designed neural network. The number of P.E.s in the input buffer of that network is determined by the number of elements in the transformed vector (26 in the given example). In a similar manner, the number of P.E.s in the output layer of the example network should be set to 13 (see Fig. 4.11(a)).

In the continuous-value transformation method (Fig. 4.11(b)) the project is transformed into a vector of real numbers, each assigned for a given input attribute. The assigned values could be made to represent a relative desirable score for all possible values of the attribute. Alternatively, the value of the attribute could simply be an index identifying the position of the value on the grid as in Fig. 4.11(b). Such index values could be pre-processed by a number of possible ways, including: scaling, normalization, and function processing (e.g., sine). These methods have been described in some references providing guidelines for collecting necessary data and preventing problems associated with the magnitude and variability in the input data (Bailey and Thompson 1990b; Lawrence 1991; Crooks 1992; Maren et al. 1990). Adopting a certain method depends on the characteristics of the paradigm (e.g., the existence of a bias node), and the type of data available. The output vector, as seen in Fig. 4.11 is also constructed from attribute values confined (i.e., scaled) to a range from 0 to 1.0. In this representation (i.e., continuous transformation), the number of P.E.s in both the input buffer and the output layer should be set as the number of attributes that represent the input and output patterns, respectively.

As presented in the crew productivity example, the problem is structured in a simple and direct manner; all inputs are presented at the input buffer and outputs presented at the output layer. This is because the problem is small in size with definitely relevant and consistent data (a non-consistent data would include mixed short term and long term data, for instance). However, more often this is not the case. Problems vary in size (number of input and output attributes), complexity, data type, and solution requirements. There is no one single approach that can be used for properly structuring a problem to suit these constraints. Table 4.1 outlines some of the techniques described in the literature for developing some neural network applications and can possibly be adopted to other applications.

4.5.2 Slow training and sensitivity to initial set of weights

The slow training or backpropagation is a result of being a gradient descent procedure (Wassermann 1989; Pao 1989; Fahlman 1987; Rigler et al. 1991). To overcome this problem, researchers have proposed various techniques to accelerate the gradient search. The simplest approach is to use a large learning rate coefficient (η), however, this might result in high oscillations that may cause the network to miss the global minimum point on the error curve (Fig. 4.12). Researchers, therefore, developed algorithms incorporating simple heuristics to dynamically adjust (η) during training (Chan and Fallside 1987; Fahlman 1988; Franzini 1987; Jacobs 1987, 1988; Silva and Almeda 1990). Weir (1991) also presented a deterministic approach for self-determination of adaptive learning rate (η). The approach achieves rapid convergence, making it possible to train neural networks on first time basis.

Table 4.1: Heuristics and techniques for problem structuring.

- A neural network should be confined to a limited domain. Separate networks should be designed for contradictory areas of the domain (e.g., short term and long term). Designing a singly huge network seems similar to a human attempting to be an expert for several fields, the result being someone who is less knowledgeable in all areas (Vanluchene and Sun 1990).

- Problems that have large number of output attributes could be structured in two possible ways. The first is as a single large network; and the second is as a group of smaller networks each is trained on a portion of the output. A single network has high learning capacity since it has large number of P.E.s and interconnections, however, requires a large training time and may be hard to be as accurate with all the output attributes. Smaller sub-networks are faster to train, however, have less learning capacity due to the fewer P.E.s and connections used. Generally speaking, however, neural networks with a greater number of inputs than outputs do better (Lawrence 1991).

- Large-scale networks could be structured by dividing the problem into several independent sub-networks, each has the same input and output layers. Each sub-network is trained on all inputs and only a part of the output while the other parts set to zero. Each sub-network is large enough to be trained well and become specialized in recognizing their outputs. After training, the networks are combined to constitute a one network with a common input buffer, a common output layer, and several task-specific hidden layers (Waibel and Hampshire 1989).

- Problems that have large number of input attributes with unclear relevance and importance can be structured in a modular structure, similar to that presented by Kadaba et al. (1990). In this approach, a form of data compression is used to produce a concentrated high order form of the inputs and outputs. This is achieved by two networks designed with hidden layers of few P.E.s. Both networks are trained to produce their own data. After training, the hidden layer P.E.s represent a compressed version of such data. These hidden layers are then utilized by a third network that uses the compressed representation of the inputs and outputs to train faster and possibly better generalize.

- Domain knowledge and the relative importance of the input attributes could be incorporated into a network design in many ways:
    - A bias can be preset towards the most important attributes.
    - Extra connection weights could be added, such as direct connections from the input buffer to the output layer (Lisboa and Perantonis 1991).
    - Additional output attribute could be added to the network, representing the quantity that is thought of importance to the problem (Caudill 1991b).

- Problems that require representation of uncertainty can be structured to incorporate a measure of "belief" in two ways:
    - Selecting input values to represent a measure of belief in the attribute.
    - Adding an additional attribute representing the measure of belief in the data used.

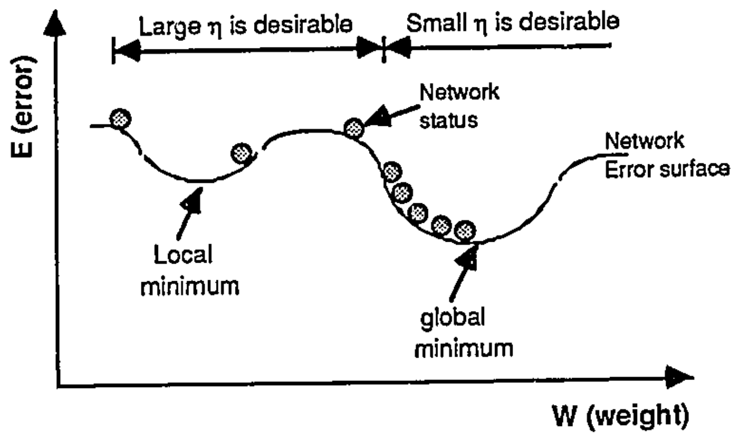


Fig. 4.12: Effect of learning rate (η) on network convergence.

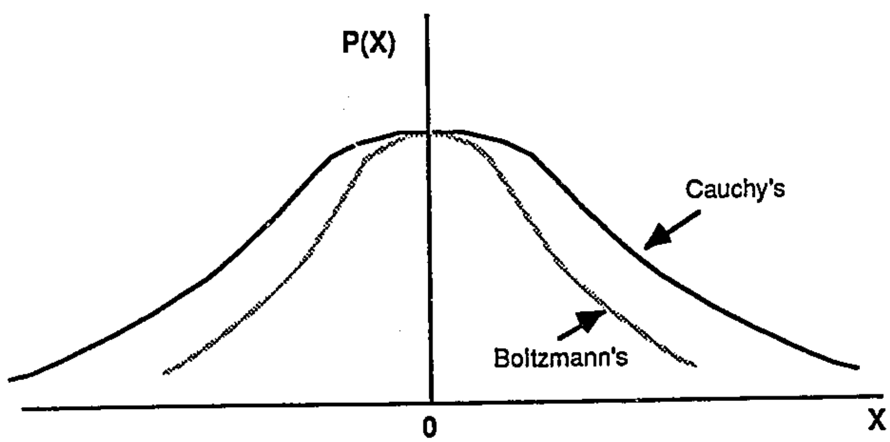


Fig. 4.13: Cauchy versus Boltzmann Distributions.

Other modifications to the conventional backpropagation algorithm are proposed to speed the convergence and reduce its dependency on the initial set of network weights. These include the work of Scalero and Tepedelenlioglu (1990) of suggesting a modified form of backpropagation using the standard least squares adaptive filtering techniques (e.g., Kalman filter (Haykin 1986)), which are known to have rapid convergence properties than those of the least mean square (LMS) approach conventionally adopted in backpropagation. The technique, despite the added mathematical complexity, converges much faster and in a more steady manner than backpropagation for pattern recognition problems.

Rezgui and Tepedelenlioglu (1990) have also studied the effect of the shape of the activation (transfer) function on the speed of convergence of the backpropagation algorithm. Based on their study, they suggested a more robust algorithm that dynamically adopts the transfer function slope during training. Testing this algorithm on training an XOR problem showed better convergence speed and less dependency on the initial set of weight values.

Since the transfer function is relatively costly to compute, Tveter (1990) used a simple technique to speed the backpropagation calculations through piece-wise linear approximations of the transfer function. This approach saves time, especially on computer systems that lack floating-point hardware. The use of integer weights also speeds calculations, although may cause local minima problem.

Another simple and easy to implement procedure for improving the training speed of the backpropagation algorithm while enhancing the stability of the process is described by Rumelhart et al. (1986). Called "Momentum", the method involves adding a term to the weight adjustment that is proportional to the amount of the previous weight change. Thus, the weight adjustment formula Eq. 4.15) is modified as follows:

$$\Delta W = \eta . \delta . X + \alpha . W_{old}$$

$$also, \quad W_{new} = W_{old} + \Delta W$$

where, α the momentum coefficient, is commonly set to 0.9 (Wassermann 1989).

4.5.3 Local minima and network paralysis

As shown in Fig. 4.12, backpropagation training could be trapped in local, rather than global minimum for the error function. As with network speed improvement, proper choice of the learning rate (η) can overcome such problems. Large (η) can skip the local minimum to another point on the error surface, while small (η) can be used to go deeper into the valley of the global minimum. The difficulty in selecting the correct (η) is the fact that the typography of the error surface is not predefined. As such, the techniques used to improve the training speed through adaptive selection of (η) could be one possible solution to the local minima problem. Statistical training methods also can help avoid this trap. Statistical training methods, as opposed to the delta rule deterministic method, make pseudo-random changes in the weight values, retaining those changes that result in improvements (Wassermann 1989). The size of the random weight change can be determined in many ways. For example, Boltzmann training method (Hinton and Sejnowski 1986) utilizes a Gaussian distribution for the weight change. Cauchy training machine (Szu and Hartley 1987) utilizes a similar distribution, however, with a different characteristics (Fig. 4.13). Both methods utilize a Monte Carlo method to produce the desired weight change.

The random search methods of Boltzmann and Cauchy can overcome the difficulty of local minima solutions that are inherent in backpropagation and other gradient-descent methods. However, the random, rather than direct, nature of the training process often result in long convergence times which are almost 100 times those of backpropagation (Wassermann 1989). Wassermann (1989) have, therefore, proposed an algorithm combining the two techniques. His algorithm makes weight adjustment equal to the sum of adjustments calculated by the backpropagation algorithm and the random step utilizing the Cauchy algorithm, producing a system that finds global minimum and converges more rapidly than a system trained by either method alone. Experiments with such algorithm have shown satisfactory performance. However, the algorithm does not prevent network paralysis problem caused by saturation of some of the network P.E.s.

Network paralysis is the condition when weight modification comes to a virtual standstill without improvement with further training. The problem occurs when some P.E.s are connected to weights so large that high values for the P.E.s' (NET) values are produced. This causes (O) to approach its limits (i.e., P.E.s are saturated). At high (O) points, the derivative of the transfer function approaches zero. As described previously, the backpropagation algorithm calculates the magnitude of the weight change using this derivative as a factor in the expression. For saturated P.E.s, the near-zero derivative causes the weight change to approach zero. If saturation is widespread throughout the network, training can slow to a near halt. This situation refers to a flat region on the network error surface. In such regions, there is very little change in the errors after update.

This phenomenon has been observed by previous researchers and several solutions have been suggested. Wassermann (1989) suggested randomizing the weights of P.E.s that are found to be in saturation. This, however, is disadvantageous in that it can badly disrupt the training process, sometimes prolonging it indefinitely. To overcome such problem in the combined backpropagation/Cauchy algorithm, Wassermann adopted a procedure for detecting saturated P.E.s by inspecting their (O) signals. All weights feeding saturated P.E.s are then operated upon by a transfer function much like that used to produce the (O) signal. The algorithm produced encouraging results when tested on small problems, however, is not fully evaluated, especially on large problems (Wassermann 1989).

Other solutions to the P.E.s saturation problem have been suggested by introducing modification to the transfer function so that its derivative is not zero and hence the network can very quickly skip the flat region in the error surface (Samad 1990; Ramirez and Arghya 1991). Samad has suggested replacement of the derivative term by a fixed value of 0.25. The value 0.25 corresponds to the steepest slope of the sigmoid function. This small change in the backpropagation algorithm has resulted in considerable improvements in saturation prevention as well as speed of convergence. Similar results have also been obtained by Ramirez and Arghya, suggesting modification to the derivatives at output layer and the hidden layers.

It should be mentioned that several of the techniques suggested for improving the speed of convergence and the dependency of training on the initial set of weights, by nature, reduce the effect of P.E.s saturation. Speedy training usually jumps over the flat regions in the error surface. Also less dependency on the initial set of weights means less possibility for P.E.s saturation. The algorithm of Rezgui and Tepedelenlioglu (1990) that adapts the slope of the activation function during training was reported to increase the speed of convergence and minimize P.E.s saturation as well.

4.5.4 Network configuration for adequate generalization

Designing (or hand crafting) a neural network configuration suitable for a particular problem is highly problem dependent. The configuration of a neural net has a huge impact on its performance. The network builder has many degrees of freedom (backpropagation parameters - Table 4.2) in defining a configuration for the network. Among those factors, the number of hidden layers and their respective number of P.E.s are most important. All this flexibility presents a problem: no defined methodology as yet exists for defining the configuration of a network to the needs of a given problem domain; it is still very much an art more than science (Barletta 1991). This forces the system developer to try different network configurations, using variations of Table 4.2 parameters, to find an optimal setup. In addition, this trial-and-error process is frequently coupled with high computational time required for network training.

Table 4.2: Backpropagation user-determined parameters.

1.	Type of input accepted, and output produced.
2.	Transfer function.
3.	Number of hidden layers.
4.	Number of P.E.s in hidden layers.
5.	Connectivity.
6.	Learning algorithm.
7.	Learning rate coefficient (η).
8.	Momentum coefficient (α).
9.	Number of training cycles.
10.	Halting conditions (acceptable error).

In the literature, combined heuristics and experiential knowledge have been used to simplify the task, arriving at an optimal network in the sense that it can learn the training data reasonably well (Bishop et al. 1991; Caudill 1991; Bailey and Thompson 1990a,b). Some of commonly used heuristics are listed in Table 4.3.

After training, the main performance measure, beside learning speed, that should be considered when testing a neural network is generalization. A neural network generalizes well if satisfactory response is produced when presented with patterns outside the training set (Le Cun 1989) - see Fig. 4.14. Generalization, thus, works as a measure of: (1) the sufficiency of the training data to cover much of the problem's solution space; (2) the suitability of the network configuration to the problem being modeled, irrespective of how well was the network performance during training. The number of training examples (sample size) required to achieve reliable generalization has been investigated by researchers including (Shaw-Taylor and Anthony 1991; Ehrenfeucht et al. 1989; Haussler 1989; Baum and Haussler 1989). The studies theoretically show that the likelihood of correct generalization depends on the size of the hypothesis space (total number of networks that can be trained on the examples), the size of solution space (set of networks that give good generalization), and the number of training examples. If the hypothesis space is too large and/or the number of training examples is too small, then there will be large number of networks that can effectively train on the examples but fail to lie in the solution space. Thus, poor generalization is to be expected. Conversely, if good generalization is required, a large number of training examples is required. Specifically, the number of examples scales like the

Table 4.3: Heuristics for selecting backpropagation parameters.

Transfer Function	- Binary input-output pairs can use threshold logic unit (TLU) as a transfer function. Bipolar (-1 and +1) input-output pairs use the hyperbolic tangent function. Continuous value input-output pairs use form of Sigmoid transfer function (Bailey and Thompson 1990a).
	- TLU transfer function and Piece-wise linear function produce either 0 or 1, thus suppressing the uncertainty and ambiguity in the decisions reached. This is desirable in yes-no decisions (Knaus 1991).
Number of Hidden layers	- Hidden layers act as layers of abstraction, pulling features from inputs. Increasing the number of hidden layers increase both the time and number of training examples necessary to train the network properly. As a rule of thumb, Start with one hidden layer and add more as required (Bailey and Thompson 1990a).
	- Multiple slabs within a single hidden layer increase the neural network processing power. Each slab (group of P.E.s) act as a detector for one or more input features (Bailey and Thompson 1990a).
	- A network with binary inputs and binary transfer function for hidden layer P.E.s, one hidden layer is sufficient to generate arbitrary mapping between inputs and outputs. In this case, J-1 P.E.s in the hidden layer are enough where, J is the number of training examples (Arai 1989).
	- If a problem attributes are chosen as the most important, independent, and adequately representative of the problem space, using more than one hidden layer in the network may not be necessary or practical. This is because the features that the hidden layers are normally used to detect are directly presented as inputs, with less need for higher level of abstraction (Caudill 1990; Dutta and Shekhar 1988).
	- A network with continuous-value inputs may require more than one hidden layer to achieve any arbitrary mapping (Chester 1990).
No. of P.E.s in a NN layers	- In binary inputs and outputs, the number of P.E.s in the input buffer is equal to the sum of the possible values for all the input attributes. Similarly, the number of output layer P.E.s is equal to the sum of the possible values for all output attributes.
	- In continuous-value inputs and outputs, the number of P.E.s in the input buffer and output layer are equal to the number of input and output attributes respectively.
	- The proper number of P.E.s for the hidden layer is often determined by experimentation. For a fixed number of training cycles, several neural network configurations are developed varying only the number of P.E.s in the hidden layer. The optimum configuration is the one achieving minimum error during the limited training session (Moselhi et al. 1991a).
	- Too few P.E.s in the hidden layer, as compared to the number of training examples, impairs the network to correctly map inputs to outputs. On the other hand, too many P.E.s promote table lookup performance and impedes generalization. The large number of P.E.s allow the network to "memorize" the examples presented to it without extracting just the salient features (Bishop et al. 1991).
	- The smallest network that can learn the examples well is expected to have better generalization performance. However, a network should be larger than the minimum necessary to perform the task (Bishop et al.1991).

No. of P.E.s in hidden layers	- With a single hidden layer, a suitable initial size is 75% of the size of the input buffer. For more than one hidden layers, reducing the size of each subsequent layers (Bailey and Thompson 1990a).
	- The number of P.E.s in a single hidden layer may be taken as the of the input buffer or output layer P.E.s (Neuroshell 1989).
	- The number of P.E.s in a single hidden layer may be taken as (2 * m + 1) where, m is the number of P.E.s in the input buffer (Kolmogorov 1957; Caudill 1988).
Network Connections	- Generally, fully connected adjacent layers within multi-layer networks is best (Bailey and Thompson 1990a).
	- Extra network connections could be used to directly link some input buffer P.E.s to some output layer P.E.s. These extra connections, if properly designed, may stabilize learning, reduce local minima, and speed the training process (Tveter 1991; Lisboa and Perantonis 1991).
Training Parameters	- If the training set exhibit some redundancy, the network weight update is better performed after presenting the network with each example, rather than after the gradient have been accumulated over the whole training set (Le Cun 1989).
	- Learning rate coefficient (η) can generally be set at 0.7. The momentum term (α) can also be set at 0.9 (Pao 1989).
	- Network can be considered to have learned the training examples reasonably well if the total summed squared error is of the order 10E-3 to 10E-4 (Bishop et al. 1991).

logarithm of the number of functions that the network architecture can implement (Le Cun 1989).

Lower and upper bounds for sample size are established for a network with a single hidden layer, as a function of the accuracy parameter, confidence limit, and the number and dimensionality of hypothesis space. The use of the lower and upper bounds equations are not easy to apply due to the complicated formulations involved and the mathematical experience required. However, the study performed by Shaw-Taylor and Anthony on multiple-output networks showed that the sample

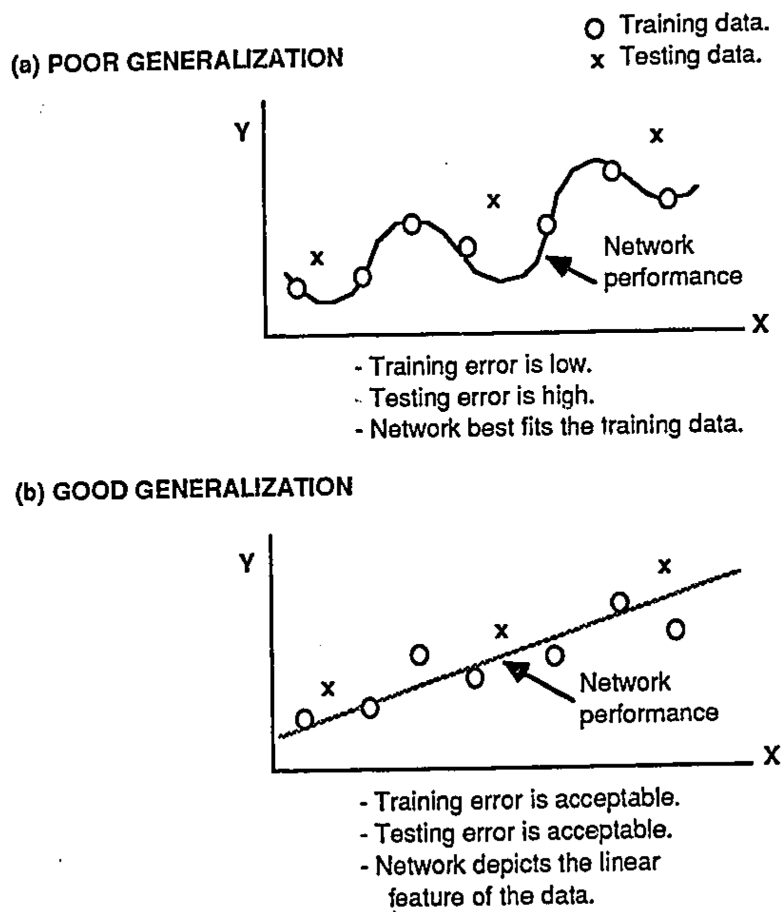


Fig. 4.14: Generalization performance of Neural Networks.

size of reliable learning can be bound from above by a quantity independent of the number of outputs of the network. This may imply that a network having a number of output attributes can generalize as equally well as a number of networks, each is trained to map the inputs to only one of the output attributes. However, such an implication deserves more investigation. A simple heuristic rule for backpropagation networks, however, is provided by Eberhart and Dobbins (1990), requiring about 10 examples for each possible output classification as minimum.

Provided that a reasonable number of training examples exists, the problem of insufficient generalization can be improved by modifying the network configuration used. The heuristic trial and error procedure for network design by hand-crafting, described earlier, relies on intensive guesswork and often leads to lack of generalization although the network may learn the training examples well. This is particularly true for large problems and when the mechanisms underlying the input-output relationships are not known so as to permit human engineering of a suitable network structure (Dodd 1991). Furthermore, even when the latter knowledge is known, it is not always obvious how to build the network appropriately since the conditions under which good generalization can be obtained are not well understood (Le Cun 1989). These problems have led to increasing research efforts directed towards: 1) investigating more systematic and problem-less network design approaches; 2) studying the generalization phenomenon; and 3) proposing automated approaches for optimizing the network configuration and achieving better generalization.

Researchers have adopted interesting approaches in developing network paradigms that have a less number of free parameters and thus, do not suffer from several backpropagation problems. Brent (1991) developed an algorithm for faster learning, in which it is not necessary to specify the number of hidden layers in advance. The algorithm is based on a key idea to construct a decision tree and then simulate the decision tree with a neural net. Brent, however, demonstrated the practicality of the technique for pattern recognition only in terms of faster convergence and not in generalization performance. Pao (1989b) also proposed a much problem-less paradigm. The "Functional Link" nets proposed by Pao basically perform a nonlinear transformation of the input patterns, thus producing enhanced patterns, before feeding them to the input buffer. If these enhanced patterns are correctly transformed, a"flat" network (one that has no hidden layers)utilizing the generalized delta rule training algorithm (the algorithm used in backpropagation) can be more quickly and correctly trained on the training set. Similar to the effort of Brent, however, generalization performance is not addressed.

Le Cun (1989) discussed the importance of reducing the number of free parameters in a neural network to increase its likelihood of correct generalization, without reducing the size of the network and correspondingly its computational power. This can effectively by done when some a priori knowledge about the task is available. Techniques such as weight sharing, described in (Rumelhart et al. 1986) for the so-called T-C problem, utilizes some symmetries that the problem has to impose equality constraints among the connection weights. A more general and problem-independent hypothesis, however, is that backpropagation networks with few units on the first hidden layer generalizes better than networks with many units (Rumelhart et al. 1986). In order to test this hypothesis, among other aspects including the effect of noisy data on generalization, Sietsma and Dow (1991) developed a technique of first training and testing a network which is known or suspected to be larger than required, and then trimming off excess P.E.s to obtain the smallest network. The procedure starts with a big network that has already been trained and perform 4 steps: (1) Pruning out non varying and parallel P.E.s; (2) reducing a layer to the minimum set of P.E.s able to accurately learn the training examples; (3) inserting extra layers, if necessary; and (4) relearning the solution. The technique exhibits several interesting characteristics and results:

- It represent an automated methodology for arriving at an optimum network configuration that learns the training data and may generalize better.
- The results showed that adding noise to the training data greatly improves the generalization performance, this is in agreement with other research work (e.g., Elman and Zipser 1988).
- Fewer P.E.s led to better generalization for networks trained with clean input

patterns. For networks trained with noise-distorted data, more P.E.s are utilized by the network and pruning excess P.E.s may prove unnecessary.

Pruning as an automated methodology for network optimization has interested many researchers for its relative ease of implementation and resemblance to the process of human cell death and regeneration. Different researchers, however, have used different laws to control the process of removing P.E.s (death of cells), adding P.E.s (regeneration of cells), and adjusting the network weights so as not much information is lost in the network as a result of adding or removing P.E.s. Karnin (1990) proposed a simple procedure for pruning big backpropagation trained networks by means of setting a subset of the network weights to zero instead of removing P.E.s. The requirement of pre-training big networks, however, was eliminated by algorithms developed by researchers like Bailey (1990) and Hirose et al.(1991) that start with 1 P.E. in the hidden layer and change the number of hidden units, and some of the weights (Bailey 1990), dynamically during training. Such algorithms require limited knowledge engineering and network design effort, converge fast, and do not suffer from local minima. However, the effect of the pruning process (and in more general sense the network size) on generalization is not very well understood. As shown by Sietsma and Dow (1991), using noise-distorted data may prove to be a more simple procedure to train networks for better generalization.

The need for a problem-independent and automated network optimization technique that can acceptably trade off between good generalization and over-specification (i.e., learning only the training examples) has stimulated the search for artificially intelligent optimization tools. A programming strategy known as "Genetic Algorithms", optimization procedures inspired by biological systems' improved fitness through evolution, have recently been utilized to optimize the design of a neural network architecture (Caudill 1991c; Bishop et al. 1991; Dodd 1991; Montana and Davis 1989; Stein 1991). As defined by Goldberg and Kuo (1987), genetic algorithms (GAs) are: "Search procedures based upon the mechanics of natural genetics, combining a Darwinian survival of the fittest philosophy with a random yet structured information exchange among a population of artificial chromosomes". The procedure of evolving an optimum neural network for a particular problem is outlined in Fig. 4.15, where GAs perform an optimising search whose search space is defined by a special coding (called gene or chromosome) of the network parameters to be optimised. The GA technique used can be defined, according to Austin (1990), as "An iterative procedure maintaining a population of structures that are candidate solutions to specific domain challenges. During each temporal increment (called a generation), the structures in the current population are rated for their effectiveness as domain solutions, and on the basis of these evaluations, a new population of candidate solutions is formed using specific 'genetic operators' such as reproduction, crossover, and mutation".

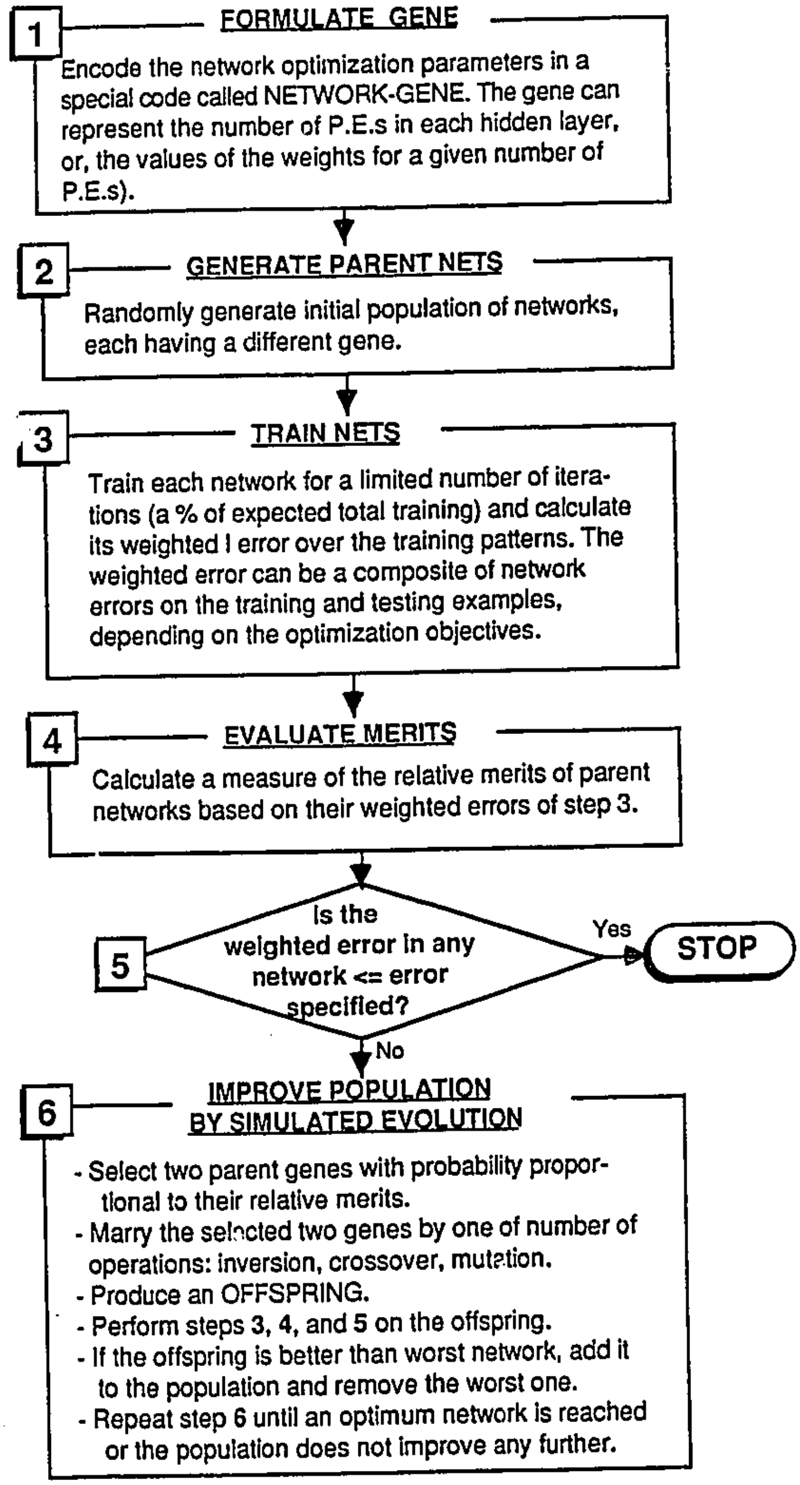


Fig. 4.15: A Procedure for Optimizing a Neural Network Using Genetic Algorithms.

From the procedure outlined in Fig. 4.15, the components of a GA for neural network optimisation are:

(1) A set of genes, each representing a vector of randomly generated values (binary or real numbers) for all the network parameters to be optimized. For neural networks, a gene may constitute of the number of hidden P.E.s needed to be determined, or, if this number is pre-determined, the values of the connection weights.

(2) An evaluation (or objective) function that returns a rating for each gene given to it. The rating represents the degree by which a gene satisfies the optimisation constraints. The error resulted after a limited training session (usually 5% to 10% of total expected training cycles (Caudill 1991c)) could be a possible evaluation function. Seeking good generalization performance, the error after limited training then testing some unseen data can be used for evaluation.

(3) A set of operations that are responsible for the process of an offspring evolution from parent genes. Possible operations include: inversion; crossover; and mutation (see Fig. 4.16). Inspired by biological systems, crossover is the most efficient operator used while Mutation occurs at a very low probability, on the order of 0.001 (Caudill 1991c).

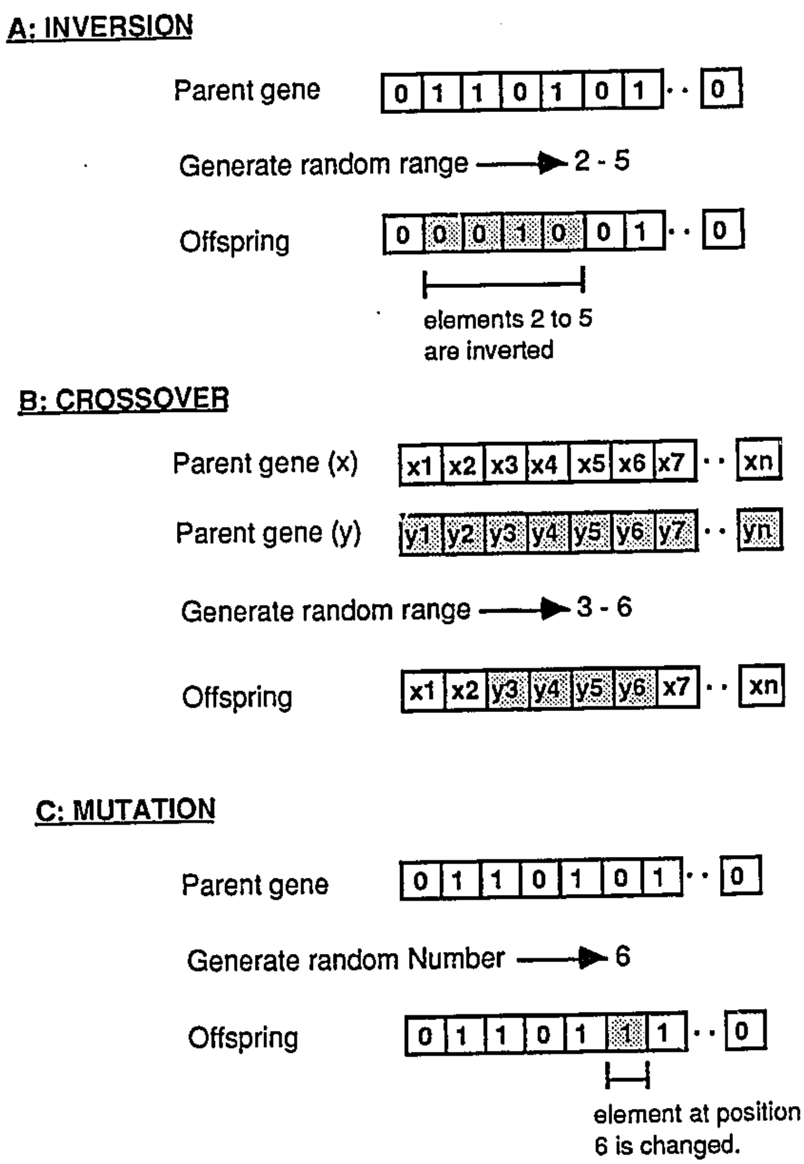


Fig. 4.16: Genetic Algorithm operations for an offspring evolution from parent genes.

Experimentation with the GAs techniques for neural network optimization showed improved generalization performance over normal training using backpropagation (Bishop et al. 1991; Montana and Davis 1989). The application of the technique, however, requires a lot of memory and system resources in order to maintain not just one network but a whole population (Caudill 1991c). In spite of these drawbacks, researchers (e.g., Dodd 1991; Chang and Lippman 1991; Austin 1990) have reported several advantages of applying the GA techniques to some extremely difficult problems. These advantages include (Holland 1975):

- It is an optimisation method that in insensitive to correlations between network parameters.
- It is a search mechanism that explores several regions of search in parallel.
- The mechanism is not generally bothered by local minima.
- No a priori information concerning the problem space is required for the algorithm to function.

From a biological perspective, Caudill (1991c) has discussed the plausibility of the GAs technique as an evolutionary process in animals. Caudill noticed distinct Lamarkian flavour to the GA technique. Lamark was a pre-Darwin biologist who contended that knowledge acquired during an animal's lifetime could be transmitted to offsprings, thus moving evolution forward. An example is the physical act of stretching the Giraffes neck that, according to Lamark, causes genetic changes that are transmitted to new generations, leading to successive lengthening in the giraffes' necks. Although neural networks remarkably follow Lamark's hypothesis, Caudill argued the possibility that genes could be modified in the course of gross physical actions. The problem with the Lamarkian evolutionary ideas is their contradiction to existing genetics knowledge and Darwin's notion of natural selection.

4.5.5 Interpretation of network weights

The major property that deems neural networks superior over conventional algorithmic and other AI-based systems is their ability to learn from examples. Unlike expert systems, however, neural networks neither automatically explain their reasoning nor provide an audit trail that fully explains how the system reaches its conclusions (Touretzkey and Pomerleau 1989). All the knowledge that the network acquires during learning are implicitly encoded in its numerical weights, bias values, and P.E.s activations. In large networks with multi-layers and large number of P.E.s and connections, a huge number of numerical values for the weights will result, making it extremely difficult to interpret and draw any meaningful explanation for the solution process. This, in addition to the non-transparency of the process by which a network maps the inputs to outputs, have contributed to the "Black Box" image of neural networks (Garson 1991).

Several research efforts have tried to demystify the "Black Box" image of neural networks by proposing innovative techniques that facilitate the interpretation of the network weights and the understanding of their underlying logic (e.g., Samad 1990; Bochereau and Bourgine 1990; Howell 1990; Touretzky and Pomerleau 1989; Garson 1991). Samad (1990) have explained the mathematical formulations of backpropagation in the form of heuristic arguments. Such an explanation can provide an intuitive understanding of the algorithm. Samad showed that the heuristic rules deducted from the original algorithm can be used to identify useful variations and extensions of backpropagation and enable the understanding of their impact on the algorithm performance. Samad, thus, proposed simple transfer functions that can be used to speed the training process. Bochereau and Bourgine (1990) presented a methodology by which a neural network can be represented by "equivalent" logical and arithmetical formulas. A neural network thus becomes capable of explaining its reasoning as a classical expert system. The technique involves analysis of the activations of the network hidden and output layer P.E.s and the weights after training. Equivalent rules can then be extracted. The technique is interesting, however, is limited to networks with one hidden layer, up to 52 P.E.s in the input buffer, and up to only 10 P.E.s in the hidden layer, where input and output P.E.s are booleans.

In networks with large number of inputs and outputs, the network P.E.s receive multi-dimensional data that all have to be presented in meaningful way to facilitate their analyses. Researchers have, thus, proposed techniques that facilitate neural networks' data representation (Hinton and Sejnowski 1986; Howell 1990; Touretzkey and Pomerleau 1989). The multidimensional data of a network's P.E.s are represented by placing a specially designed symbol at the location of each P.E., and the various parts of the symbol can be scaled in proportion to the magnitude of the dimensions of the weight vector (Howell 1990). Different P.E.s symbols are shown in Fig. 4.17 (Howell 1990). An example P.E. is given in Fig. 4.18 where the elements of the star correspond to the value of the weight connecting that P.E. to the six dimensional input space (i.e., six input attributes). Such symbol shows that the P.E. receives strong signals from attributes 2, 5, and 6. Correspondingly, the activation of such P.E. can be correlated to those attributes. The example demonstrates that these techniques may unravel the logic behind a neural network solution. The main problems, however, are the large time it takes to understand what the data is trying to tell, in addition to the additional programming requirements, particularly for large problems.

Garson (1991) presented a simple and innovative technique to interpret the relative importance of each input attribute to the conclusions reached by the network, for the purpose of causal analysis. Based on the technique, the rules governing the causal relationships could be inferred from the training examples. This process is reminiscent of inductive expert systems, which also infer rules from training examples. However, since inductive expert systems are mainly used for univariate analysis, they are limited to less complicated problems.

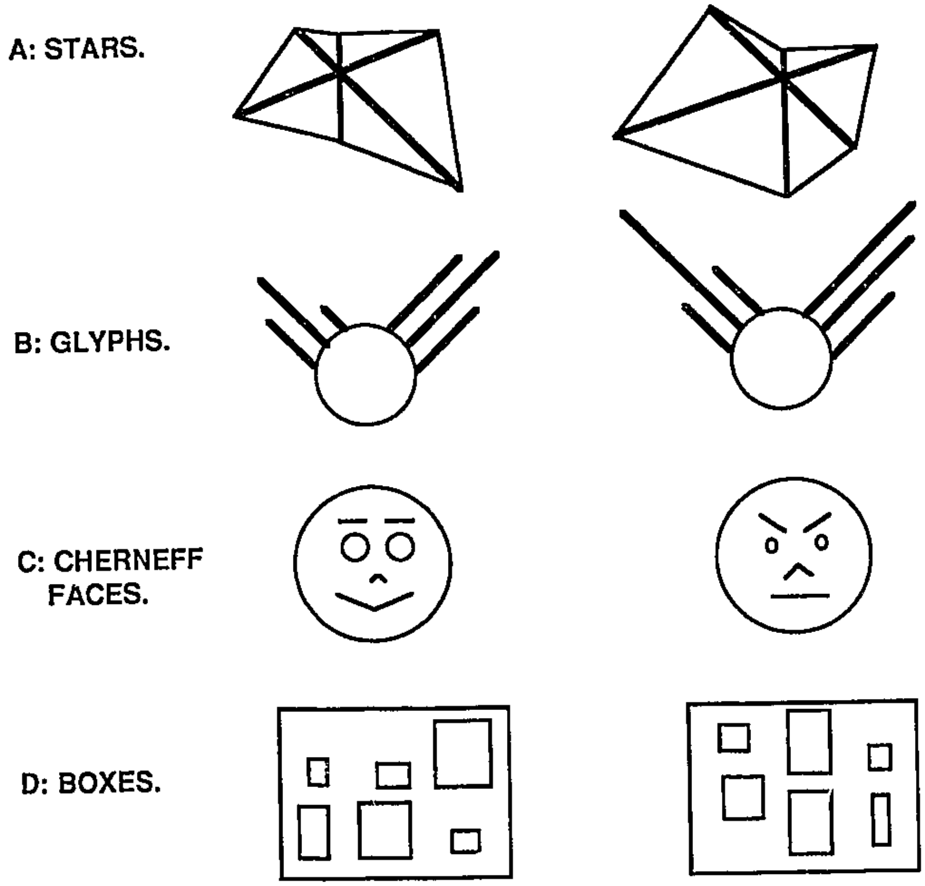
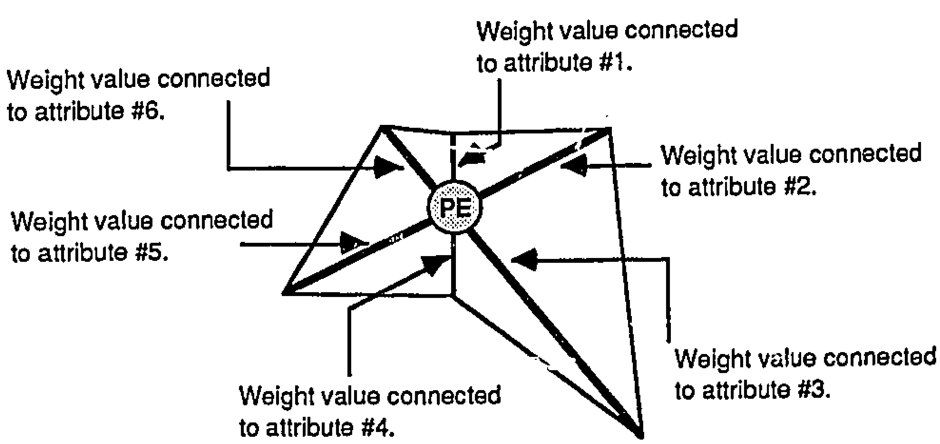


Fig. 4.17: Examples of P.E.s symbols.



The PE receives stronger signals from attributes 2, 3, & 5.

Fig. 4. 18: A detailed STAR representation.

The technique developed by Garson evaluates the relative importance of an attribute (V) through a process of *partition output layer connection weights*, for a network with one hidden layer, into components associated with each input attribute. The technique uses the absolute values of all weights, without considering the P.E.s thresholds into the computations. The process can be illustrated as shown in Fig. 4.19, where a weight connected to the output layer O_J can be divided into components each corresponds to one of the input attributes. For a hidden P.E. (J) the component of O_J associated with v is proportional to the weight incoming from attribute V (I_{VJ}) in relation to the sum of weights incoming from all attributes. This can be expressed as:

$$\textit{Component of } O_J \textit{ associated with } V = \frac{I_{VJ}}{\sum_{K}^{n_V} I_{VJ}} \cdot O_J$$

Performing this calculation for all the hidden layer and summing the shares of each input attribute results in a total share or score for each attribute. The relative importance of each attribute can then be calculated as a percentage of the sum of scores (as calculated by Eq. 4.17).

Garson demonstrated the performance of the technique on a simple network trained on examples generated based on known causal rules. After training, the technique was used to infer the relative importance of the input attributes, which were more accurate than those depicted by common statistical techniques

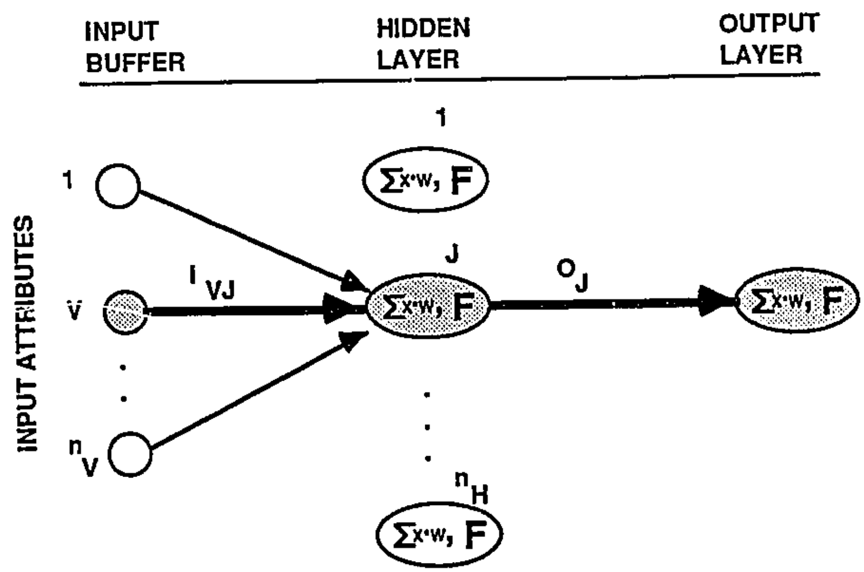


Fig. 4.19: Partitioning of output layer weights into components associated with input attributes.

$$\text{Relative importance of attribute } (V) = \frac{\displaystyle\sum_{J}^{n_h}\left(\frac{I_{VJ}}{\sum_{K}^{n_V} I_{VJ}} \cdot O_J\right)}{\displaystyle\sum_{i}^{n_V}\left(\sum_{J}^{n_h}\left(\frac{I_{VJ}}{\sum_{K}^{n_V} I_{VJ}} \cdot O_J\right)\right)} \qquad [4.17]$$

(regression analysis). The method, though simple, stands in sharp contrast to misleading views of neural networks as "Black Boxes" whose iterative processes are beyond human comprehension, even if predictions are good.

4.6 Application Development Methodology

In contrast to traditional development methodologies for software systems and more recently, expert systems, neural network development places a stronger emphasis on experimentation and multiple simultaneous development tracks. It often requires iterative refining of network parameters, problem redesign and re-formulation, and beginning with general solutions and tightening the set of feasible approaches. Also, the development of a neural network application is highly problem dependent. The objective in proposing a development methodology is to add identifiable tasks and milestones to a process that often lacks structure and organization. Obvious benefits include cost control, increased accuracy and consistency, efficient use of resources, less development problems, and increased user satisfaction. The life cycle of an application, as shown in Fig. 4.20, consists of four principle phases: 1) concept; 2) design; 3) implementation; and 4) recall.

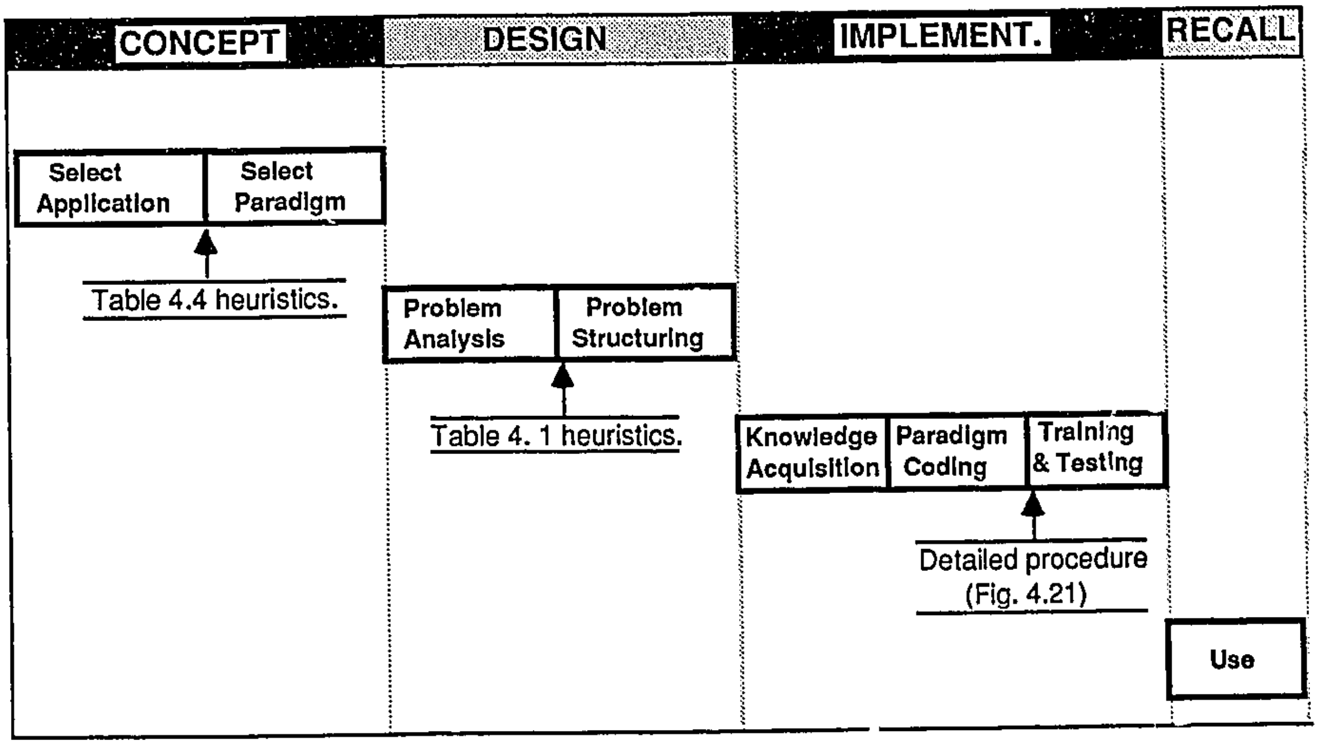


Fig. 4. 20: Life cycle phases of a neural network application.

In the concept phase, in addition to traditional feasibility-assessment activities, the proposed application is validated and a neural network paradigm is selected as a suitable environment for building the application. One approach to validate an application is to compare its characteristics to those of potential neural network applications, as performed in chapter 2 (see Table 2.2). The selection of a neural network paradigm can also be based on a comparison of the application requirements to neural paradigm capabilities. For backpropagation paradigm, in particular, some heuristics that can be used to validate its suitability for an application are listed in Table 4.4.

Table 4.4: Heuristics for selecting the backpropagation paradigm.

- Nonstatistical neural paradigms (e.g., Backpropagation) work best for networks with less than 1,000 P.E.s. Statistical paradigms (Boltzmann machine) require significantly less (in the order of 200 P.E.s) (Bailey and Thompson 1990a).
- Backpropagation is a network model most suited to small networks of 3 layers. In software simulations with no hardware accelerator or other assistance, a three-layer network with about 200-300 or fewer total P.E.s can be built. On a High speed workstation, the network size can be stretched to about 500-600 P.E.s in the network. If commercially available hardware accelerator card is used, substantially large networks can be built (Caudill 1991a).
- Backpropagation is especially bad at learning problems that are not monotonic (a monotonic is ever increasing or decreasing, but not symmetric, e.g., $y = x^2$ is non-monotonic) (Caudill 1991b).

Having selected a suitable neural network paradigm, the first step in the design phase is problem analysis which constitutes of: (a) identifying all data that in any way relate to the application area; (b) removing data sources that are regarded as peripheral or unreliable; (c) filtering out data sources that are impractical for technical or economical reasons; and (d) exploring methods of combining or preprocessing data to make it more meaningful (e.g., ratios are often more significant than numbers and provide the relevant relationship explicitly). Based on such thorough analysis, the problem of governing attributes, representing the system input, and the problem output attributes, representing the system conclusions, can be identified. These attributes, if well selected, normally represent a high order of abstraction from the data. They are independent with little correlation between any two of them. Such arrangement ensures that the attributes adequately define the problem and require less network design and implementation effort.

The second design step is to identify some feasible approaches for problem structuring. As previously mentioned, Table 4.1 outlines some of the techniques described in the literature for developing some NN applications and can possibly be adopted to other applications. It is important, however, to identify more than one approach since successful implementation of either of them is not guaranteed to suit the problem. This step is necessary before conducting knowledge acquisition since alternative approaches may require additional information to be elicited. Using Table 4.1 as a guide, feasible problem structuring approaches are determined based on several factors including: (a) number of input and output attributes identified and their data types; (b) constraints on training examples (number, time needed, and cost needed); (c) clarity or fuzziness of the problem; and (d) availability of domain heuristic knowledge that may guide the search.

Once there is a clear idea about some feasible structures and the information needed to be elicited, the implementation phase starts with knowledge acquisition and data preparation. In this step, the focus is on gathering as much training data as possible within practical limits of time, money, and computer resources. More important than the quantity of examples obtained is their quality and representativeness of the data set. A good training set should contain routine, unusual, and boundary-condition cases (Bailey and Thompson 1990b). One measure of the data's representativeness is the breadth of the problem covered by the training cases, including the different types of patterns in a pattern-classification problem, variety of significant cases in a decision-making process, and salient features of a continuous function. Often, collecting the training set for a neural network is a tedious process. Potential data sources include historical records, test data, case studies, instrument readings, simulation results, and hypothesized results. Once enough data are elicited, it can be prepared for use by a neural network, retaining some of these data for validation and testing after training. Preparing the data can take many forms (Bailey and Thompson 1990b):

- Transforming inputs and outputs into proper form (ratios, classes) and data types (binary, continuous).
- Preprocessing, such as filtering, may expedite the training process.

- Converting the data to ASCII or binary vectors.
- Normalizing (Euclidean, logarithmic) the data.
- Scaling the data, which is widely adopted, minimizes the effects of magnitude among inputs and increases the effectiveness of some learning algorithms.

The second implementation step is to code the paradigm selected. Alternatively, the user may employ a neural network package (development environment) from those currently being marketed (Stubbs 1990). Following that step is an iterative procedure of selecting a network configuration, training the network, and then, testing its responses. If backpropagation is used as the network paradigm, a detailed implementation procedure that seeks good generalization performance is outlined as shown in Fig. 4.21 (Hegazy et al. 1993b). Such a procedure utilizes the heuristics and techniques used to overcome backpropagation training problems and improve its generalization performance (Table 4.5). The implementation procedure described in Fig. 4.21 is general, starting with the hand crafting for its simplicity, however, if failed to train well or achieve good generalization, various modifications could be tried in a cycle of network re-configuration and re-training. If the problem persists, an automated network optimization technique can be used. It is possible, however, to restrict the implementation procedure to either technique depending on the criteria provided in Table 4.6. The table gives a comparison of the environment suitable for the three known network implementation techniques: hand crafting, pruning, and genetic algorithms.

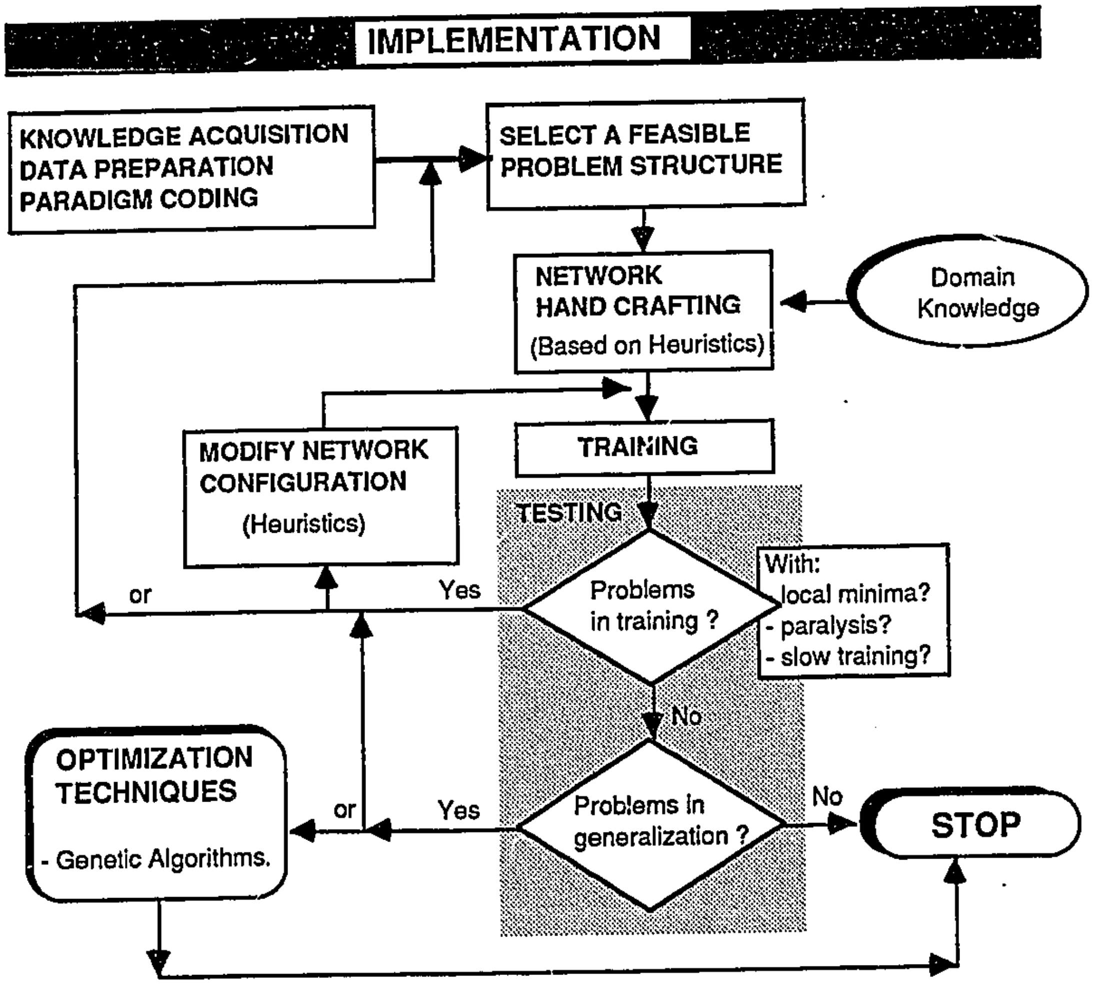


Fig. 4.21: A NN implementation procedure that seeks good generalization.

Table 4.5: Summary of simple techniques and heuristics used to improve backpropagation performance.

Backpropagation Problem	Heuristic rule
Local Minima	1- Restarting; re-initialize the network weights to some new set of random values, i.e., starting from a different point on the error surface (Caudill 1991a).
Local Minima	2- Shaking the network weights; vary the network weights by adding to each a random number as much as 10% of the original weight range (Caudill 1991a).
Local Minima and Slow Training	3- Starting with a learning rate coefficient (η) of 0.7 (Pao 1989). The coefficient may be reduced during training or dynamically adapted (Jacobs 1988).
Slow Training	4- Adding a momentum term to the weight adjustment formulas. The momentum coefficient (α) is kept as high as 0.9 (Pao 1989).
Inadequate Training and/or Generalization	5- Using alternate data representation and problem encoding technique (Bailey and Thompson 1990a; 1990b).
"	6- Increasing the number of training examples (Le Cun 1989).
"	7- Using simulation, test results, and field observations to generate training examples (Bailey and Thompson 1990a; 1990b).
"	8- Adding noise to the training examples (Sietsma and Dow 1991).
"	9- Using modular neural network structure and data compression techniques (Kadaba et al. 1990).
"	10- Using a network that is slightly larger than minimum necessary to perform the job (Sietsma and Dow 1991).
"	11- Using automated network optimization techniques: Genetic Algorithms, pruning (Dodd 1991; Bishop et al. 1991).

Table 4.6: Comparison between common NN implementation techniques.

Criteria	Implementation Techniques		
	Hand-Crafting	**Pruning**	**Genetic Algorithms**
- Problem size and complexity (no. of attributes).	Small	Medium-Large	Medium-Large
- Need for problem restructuring and re-training.	High	May be	Rarely
- Number of training examples needed.	Large	Medium-Large	Medium-Large
- Automated performance (problem independent).	No	Not Fully	Yes
- Software requirement.	Low	Medium-High	Medium-High
- Hardware requirement.	Medium	Medium-High	Medium-High
- Computational cost during training.	Medium-High	Medium	High
- Generalization performance.	Poor-Adequate	Adequate	Adequate-Good

Once a network application is successfully implemented, it can be recalled or used for direct problem solving. In this phase, the network can be presented with inputs not used in training and the network is expected to produce its responses, out of the experience gained during training.

4.7 Conclusion

Backpropagation networks are multi-layered feed-forward networks (sometimes referred to as hierarchical networks) with two or more layers (excluding the input buffer). In this chapter, the mathematical derivation of the backpropagation algorithm has been described. Problems that face the development of practical applications using backpropagation are identified and discussed. It seems that these problems are somehow related since, for example, improving the training speed often results in less local minima problems and may even improve generalization. Several of the techniques and heuristics used in the literature to overcome these problems are compiled and outlined. In general, a proper combination of these techniques will eventually improve the training efficiency and consequently the performance of a designed network. In order to guide the process of developing practical neural network applications, particularly with the backpropagation paradigm, an application development methodology is proposed. The benefits of such a methodology would be faster and less problematic application development in addition to higher likelihood of fulfilling the problem solution requirements. The methodology incorporates the heuristics and techniques used for problem structuring, network hand crafting, and network automated optimization methods that are geared towards better generalization. Such a methodology is utilized in modelling the markup problem, as described in the chapters ahead.

CHAPTER 5

DEVELOPING THE NN MODEL FOR MARKUP ESTIMATION

5.1 Introduction

In previous chapters, existing markup estimation models were reviewed and their limitations identified, along with the characteristics that render the markup problem more suitable for NN modelling. In addition, neural networks were established as systems with analogy-based capabilities derived from a process of learning from a set of examples representing previous encounters of problems. NNs utilize these holistic examples (without their underlying logic), as patterns, to simulate complicated decision processes and their related knowledge. Accordingly, they become able to devise solutions to new situations even with incomplete and/or noisy information. In order to facilitate the development of a practical neural network model for the markup problem, an application development methodology was established and presented.

Based on the methodology proposed, this chapter describes the development of a markup estimation model using the backpropagation NN paradigm. The present model utilizes the domain general knowledge depicted from 78 contractors in Canada and the U.S. and accounts for the quantitative as well as qualitative factors that affect bidding decisions. The markup estimation process is analyzed and attributes governing the decision process identified. Two alternate designs for the NN model are examined and their results compared. The first model is based on a single neural network architecture and the second is based on a five-network hierarchical system. As opposed to a single large network, the hierarchical model consists of four sub-networks, pertaining to the assessment of: job uncertainty, job complexity, market conditions, and company capabilities. The results of the four sub-networks form the input to a macro-level neural network designed to estimate the optimum markup, for a given project environment. A questionnaire survey is developed to elicit the required knowledge, from general contractors in Canada and the U.S., pertaining to bidding situations or encounters of some past projects. Analysis of the survey responses are utilized to structure, design, implement, train, and test the two neural network models. Details of the design, knowledge acquisition and validation, implementation, and testing of the neural network models examined are described along with the heuristics used to overcome the development problems encountered. Efforts pertaining to improving the generalization capabilities of the neural network model, including the application of the genetic algorithms technique, are described.

5.2 Model Design

The development of the markup model in hand follows the structured methodology (section 4.6 of chapter 4, Figs. 4.20 and 4.21) for developing practical NN applications. The methodology incorporates four main phases: 1) concept; 2) design; 3) implementation; and 4) recall or use for direct problem solving. For the markup estimation problem in hand, issues pertaining to the concept phase have previously been discussed, leading to the selection of backpropagation paradigm as a most suitable development environment.

The design phase consists of two main tasks: 1) problem analysis; and 2) problem structuring. Concerning the first task, characteristic factors that need to be considered in formulating a successful bidding strategy have recently been identified in a recent survey conducted among the top 400 contractors in the U.S. (Ahmad and Minkarah 1988b), as described in the literature review of chapter 2. Based on the survey responses, a list of 31 factors was arranged in two different orders, indicating the relative importance of those factors to the bid/no-bid decision and to the percent markup estimation. Based on the survey findings, Ahmad (1990) identified that the bid/no-bid and percent markup decisions are predominantly attributed to the group of factors that arise from uncertainty. Accordingly, these factors are used in the present model and grouped to form the risk pattern associated with individual projects.

With respect to problem structuring, a direct architecture for the markup model is clearly a one large neural network having all the identified factors as inputs. However, the large number of factors (problem attributes) involved, generally, may require a large number of training examples, for the development of a practical model. Since the number of training examples that could be obtained through knowledge acquisition is rather limited, and in view of the unavailability of a simple and rational procedure to determine the "necessary" number of training examples as a function of the problem attributes and the accuracy required, the single-network model and a hierarchical alternative were examined. The hierarchical model was seeked on the assumption that it may require less number of training examples to produce the same level of accuracy of the single-network model. The model takes advantage of the inherent hierarchical structure of the problem itself. Arranging and grouping relevant factors in a hierarchical architecture may often be possible through problem decomposition (Ahmad 1990). Both modelling approaches have their potential benefits, and a comparison between their results is necessary to determine the model that best suits the present problem.

Before acquiring the knowledge necessary for model development, the two alternative designs for the markup model: 1) a single network; and 2) a hierarchical system, have to be structured with their inputs and outputs identified. The first model is straight forward, as shown in Fig. 5.1, with thirty input attributes representing the project environment pattern and seven output attributes representing the system outputs. Two of these input attributes (markup definition (7) and markup components (8), see Fig. 5.1) are included in order to accommodate for their different interpretations by contractors, ensuring generic representation of the domain knowledge. On one hand, markup definition, as perceived by different contractors, could be: 1) a % of the direct cost; 2) a % of the

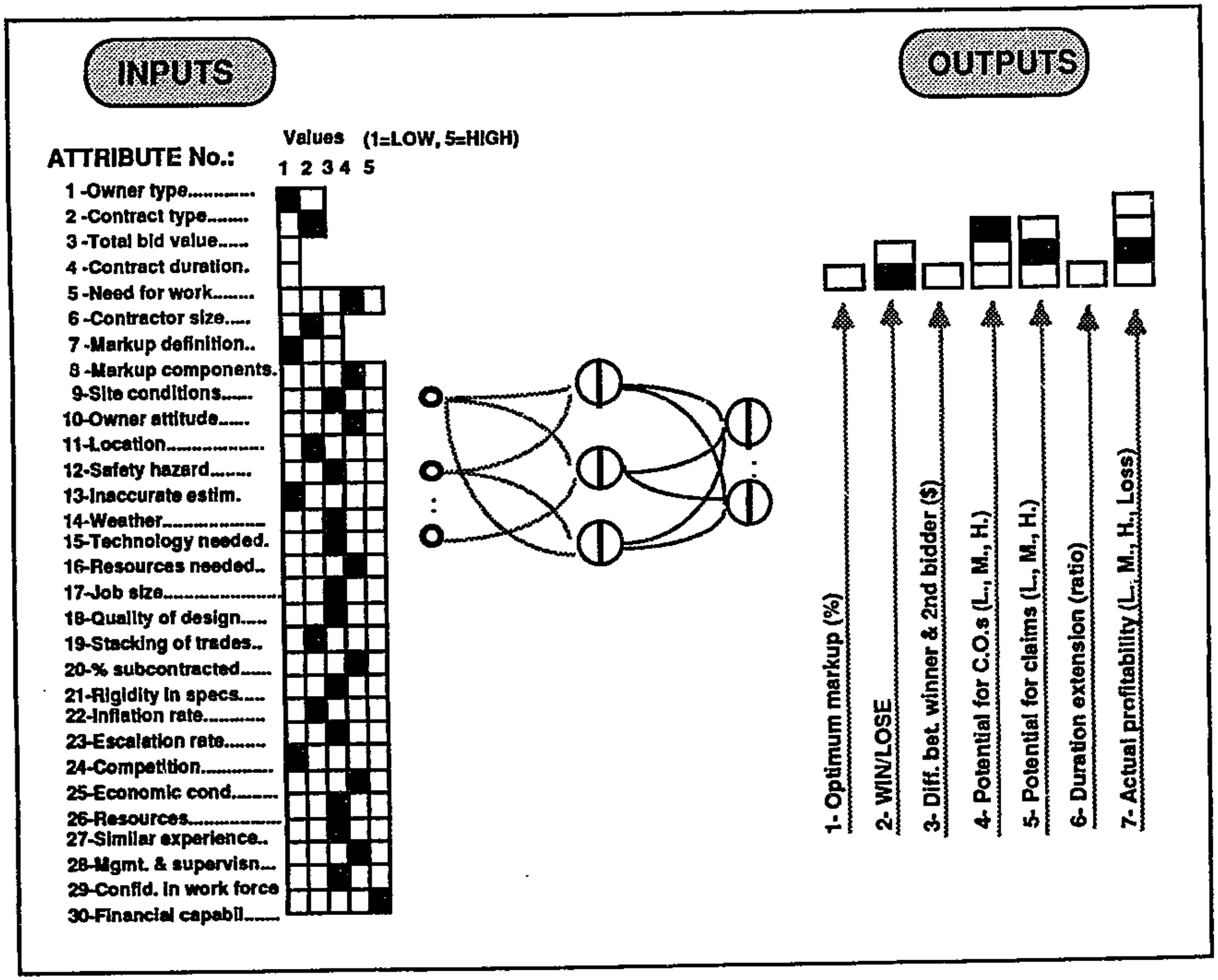


Fig. 5.1: A single neural network for the optimum markup problem.

direct cost + project overhead; or 3) a % of the direct cost + project overhead + general overhead. On the other hand, markup may be perceived to include an allowance for: 1) profit only; 2) profit + contingency; 3) profit + general overhead; 3) profit + general overhead + contingency; or 4) profit + general overhead + project overhead + contingency. As shown in Fig. 5.1, the seven output attributes clearly represent important factors that directly impact the bidding decision process. Not only a markup value need to be estimated but it is also required to give an indication about the implications of the qualitative factors and the estimated markup on: the chances of win/loose; difference in ($) between the winner and second bid; project potential for change orders as high, medium, or low; project potential for claims as high, medium, or low; duration extension; and actual project profitability as high, medium, low, or loss.

In order to structure the hierarchical model, a series of interviews were conducted among a number of Montreal area contractors in building and heavy civil works. The interviews led to the formation and grouping of factors into four independent categories: (1) job uncertainty; (2) job complexity; (3) market conditions; and (4) company capabilities. Those categories represent the contractors' abstract-level rationale in assessing risks in general and markup estimation in particular. Each category with its constituent factors was then modelled in a micro-level neural network (sub-network) that is highly specialised and knowledgeable about the output (sub-decision) it provides (see Fig. 5.2). The outputs of the four sub-

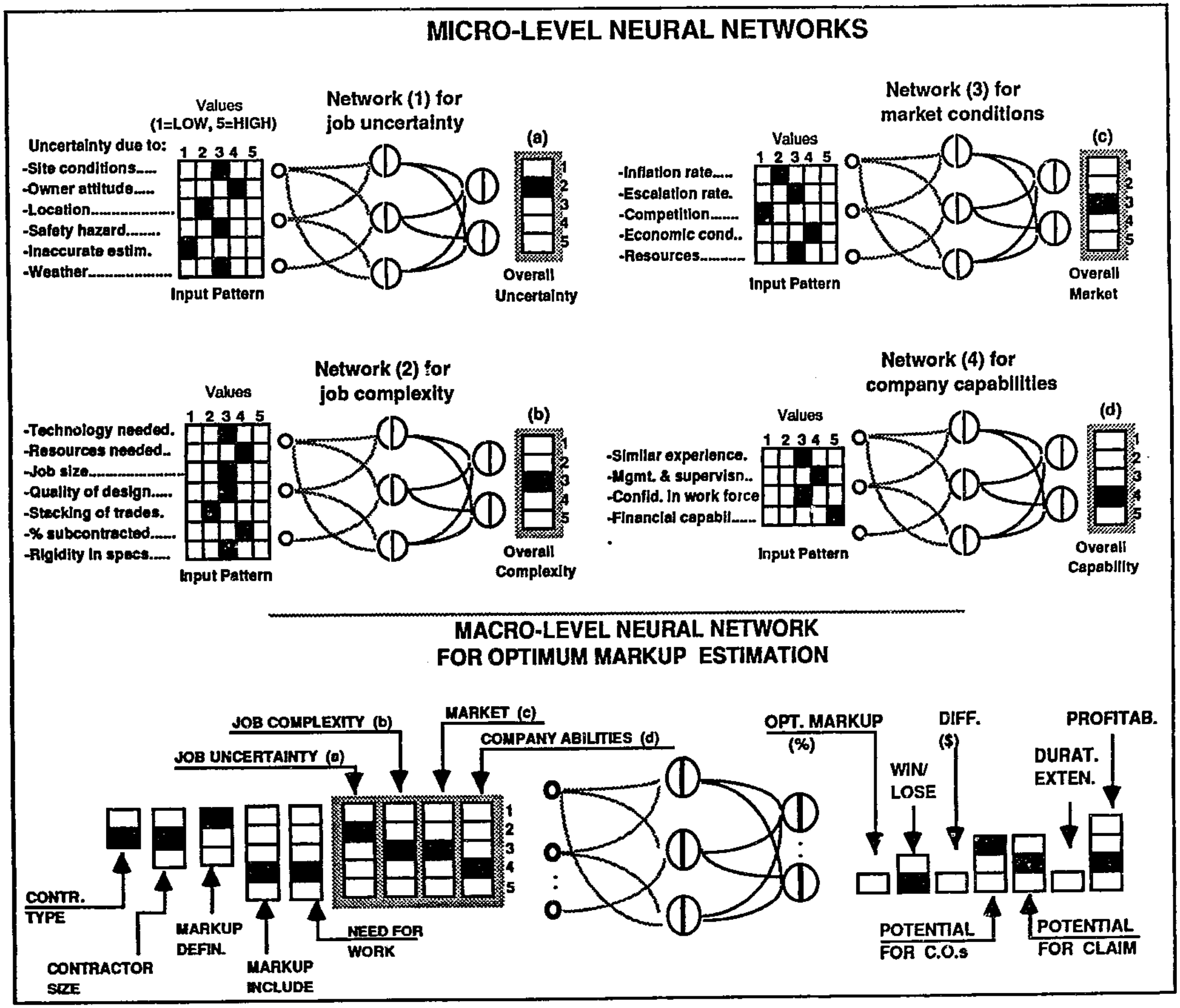


Fig. 5.2: Hierarchical neural networks for the optimum markup problem.

networks, in addition to five other attributes, are consequently used as an input to a global (macro-level) neural network that produces the same seven output attributes used in the single-network.

In all the neural networks presented in Figs. 5.1 and 5.2, an input pattern is constructed from a chess-like grid of the network attributes and the possible values of each. The possible values and their associated literal meanings are listed in Table 5.1, pertaining to 30 input attributes of the single-network model; the overall ranks of the 4 sub-networks in the hierarchical model; and the 7 output attributes of the two models. As shown in Figs. 5.1 and 5.2, the black spots on the grid represent the attributes' values in a particular training example.

5.3 Model Implementation

As described in chapter 4, the implementation of a neural network constitutes three main aspects: 1) acquiring the necessary knowledge; 2) coding the neural network paradigm selected for the problem; and 3) training and testing (validating) the network. For backpropagation-based applications, a detailed implementation procedure is illustrated in Fig. 4.21. The follow up of such procedure for the problem in hand is discussed in the next sub-sections.

Table 5.1: Description of markup attributes.

Attribute	Possible values	Scaling operation
INPUTS:		
1- Owner type	- **1**: Public; & **2**: Private.	- (Value - 1).
2- Contract type	- **1**: Lump Sum; **2**: Unit Price;	- (Value - 1).
3- Total bid value	- Real Number ($ millions).	- (Value / 20).
4- Contract duration	- Real number (months).	- (Value / 10).
5- Need for work	- **1**: Low to **5**: High.	- (Value / 5).
6- Contractor size	- **0**: Small; **1**: Medium; **2**: Large.	- (Value / 2).
7- Markup definition	- **0**: a % of (DIR*); **1**: a % of (DIR+P.O.**); **2**: a % of (DIR+P.O.+G.O.***).	- (Value / 2).
8- Markup components	- **0**: Profit only; **1**: Profit+Cont.****; **2**: Profit+G.O.; **3**: Profit+G.O.+Cont.; **4**: Profit+P.O.+G.O.+Cont.	- (Value / 4).
9 to 30	- **1**: Low to **5**: High.	- (Value / 5).
- Rank of Job Uncertainty	- (**1**: Low to **5**: High).	- (Value / 5).
- Rank of Job Complexity	- (**1**: Low to **5**: High).	- (Value / 5).
- Rank of Market	- (**1**: Low to **5**: High).	- (Value / 5).
- Rank of Ability	- (**1**: Low to **5**: High).	- (Value / 5).
OUTPUTS:		
1- Markup	- Real number (%)	- (Value / 50).
2- Win/lose	- **0**: Win; **1**: Lose.	- (Value).
3- Diff. between winner and second bidder	- Real number ($ * 10E-5).	- (Value / 10).
4- Potential for change orders	- **1**: High; **2**: Medium; **3**: Low.	- (Value / 5).
5- Potential for claims	- **1**: High; **2**: Medium; **3**: Low.	- (Value / 5).
6- Duration extension	- Real number (Ratio = Actual duration / contract duration).	- (Value / 2).
7- Actual profitability	- **1**: High; **2**: Medium; **3**: Low; **4**: loss.	- (Value / 5).

* DIR = Direct cost.
** P.O. = Project Overhead cost.
*** G.O. = General Overhead cost.
**** Cont. = Contingency.

5.3.1 Knowledge acquisition: a questionnaire survey

Based on the initial interviews conducted among a number of Montreal area contractors, a questionnaire survey was prepared and mailed to 427 general contractors in Canada and the U.S.: 211 from Canada (Duns & Bradstreet 1990); 168 from the top 400 U.S. contractors (ENR 1990); and 48 from the top 200 contractors in the province of Québec (Québec Construction 1990). In an effort to maximize homogeneity and minimize variability, questionnaires were sent to contractors who appeared to satisfy the following criteria:

- are general contractors.
- specialize mainly in building construction.
- Obtain a large percentage of their work based on competitive bidding.

These criteria were used later to qualify the respondents, ensuring some commonalities among the qualified participants and limiting the markup estimation model to a well defined domain.

The questionnaire, see Appendix I, is divided into three sections. The first elicits general information about the participating firm, including: type of contractor, annual sales, average job size, and percentage of work obtained through competitive bidding. This information was used to identify contractors who generally satisfy the criteria set above. The second section focuses on the firm's policy with regard to bidding strategy. This section is designed to depict the cost estimation and bid preparation practices and their respective components (direct cost, project overhead, general overhead, and markup). This helps in making necessary adjustments to the training examples obtained form different contractors, accounting for the different markup definitions adopted by the participants. This ensures uniform and consistent representation in the development of the model.

The third section of the survey elicits a number of complete bidding examples from the firm's past records. These examples may include successful and unsuccessful bidding situations. Respondents were asked to provide complete data about:

- project general information (e.g., project type and contract type);
- assessment of the various factors affecting markup and overall category ranks, using a score from 1 to 5;
- contractor's bid data (total bid price and percent markup decided);
- bid outcomes (win/loose and difference in ($) between the winner and second lowest bidder); and
- if successful, after construction outcomes (intensity of change orders experienced; intensity of claims experienced; duration extension depicted; and level of actual project profitability attained).

5.3.2 Analysis of Survey Responses

The survey was mailed to contractors in the Spring of 1991. A total of 90 contractors have responded, a response rate of 21%. Out of the 90 responses, 78 were qualified as they meet the criteria mentioned above. Analysis of the 78 qualified responses are compiled in Figs. 5.3 to 5.14. Contractors' General information are illustrated in Figs. 5.3 to 5.8, pertaining to their annual sales, value of construction equipment owned, percentage of work based on competitive bidding, average job size, percentage of work subcontracted on average job, and percentage of equipment leased, respectively. These figures show a high degree of homogeneity among the participants and exhibit a typical environment of a building contractor with low percentage of equipment owned, and a high percentage of work subcontracted. Figures 5.9 to 5.14, on the other hand, depict the cost estimation practices of the survey respondents with respect to direct cost, project overhead cost, general overhead cost, adopted markup definitions, and perceived markup components, respectively. These figures show that:

- Majority of contractors (92%, see Fig. 5.9) estimate direct cost in a detailed manner.
- Majority of contractors (83%, see Fig. 5.10) estimate project overhead costs in a detailed manner, while minority (14%) take it as a percentage of direct cost.

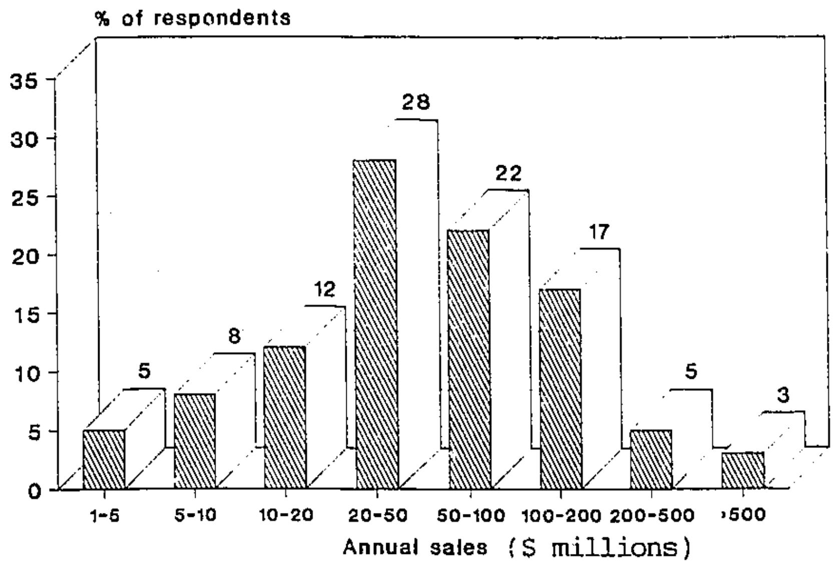


Fig. 5.3: Annual sales of survey respondents.

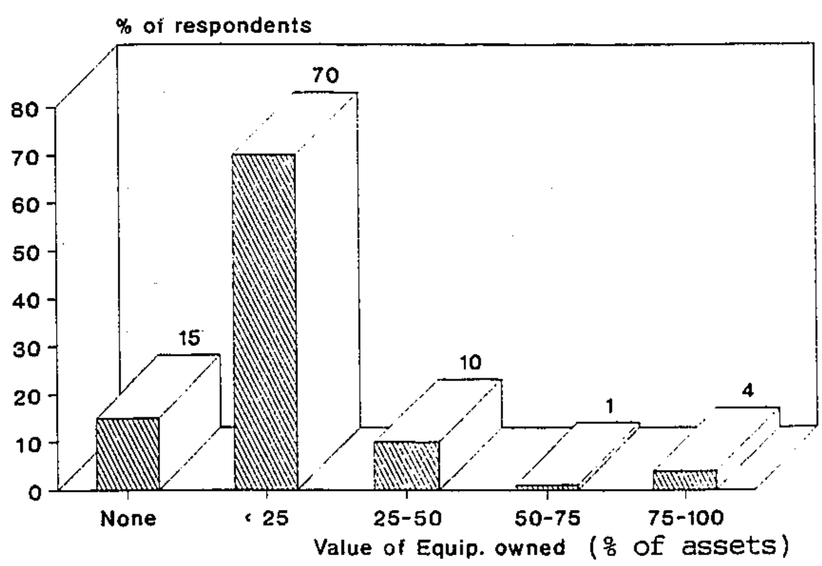


Fig. 5.4: Value of construction equipment owned.

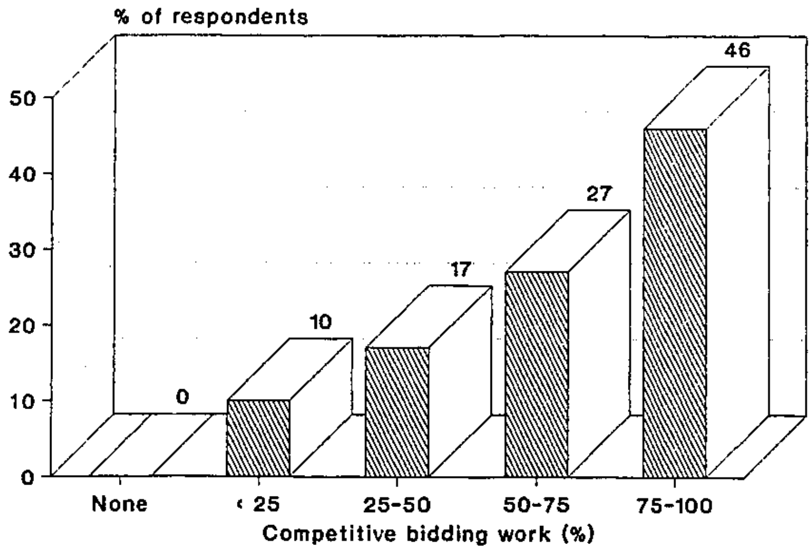


Fig. 5.5: % of work based on competitive bidding.

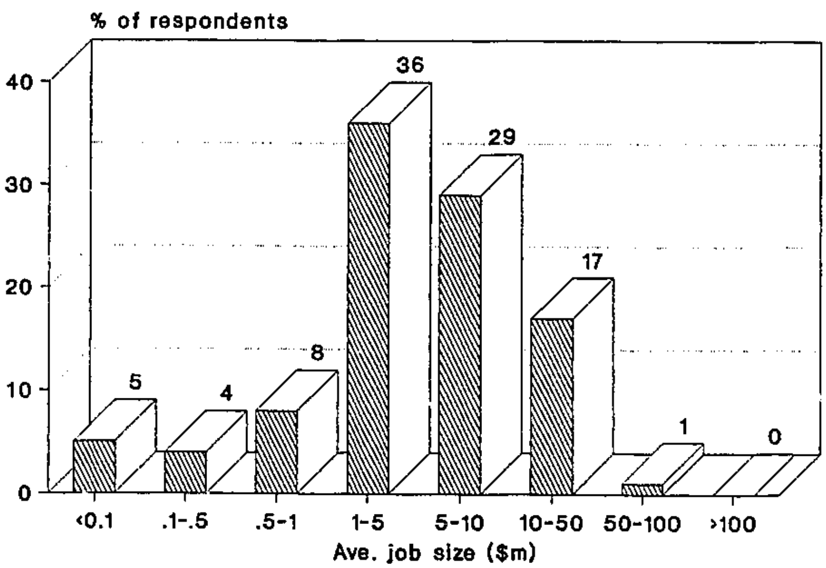


Fig. 5.6: Average job size ($millions).

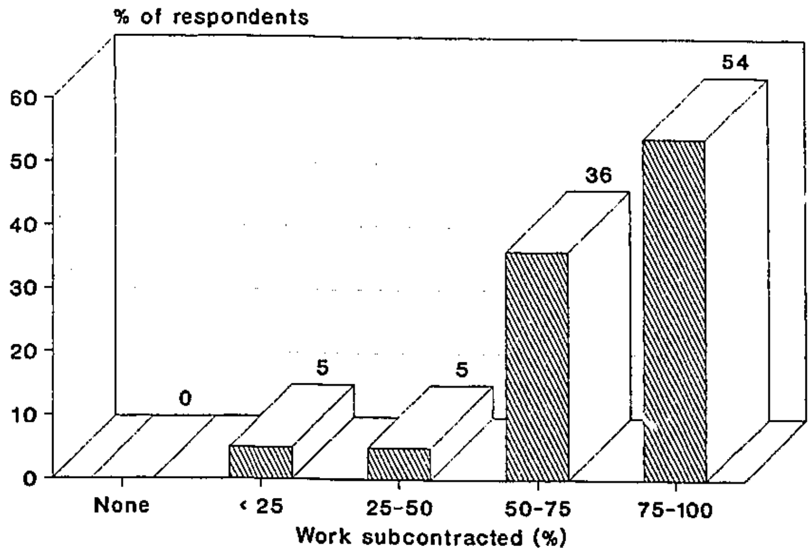


Fig. 5.7: % of work subcontracted on average job.

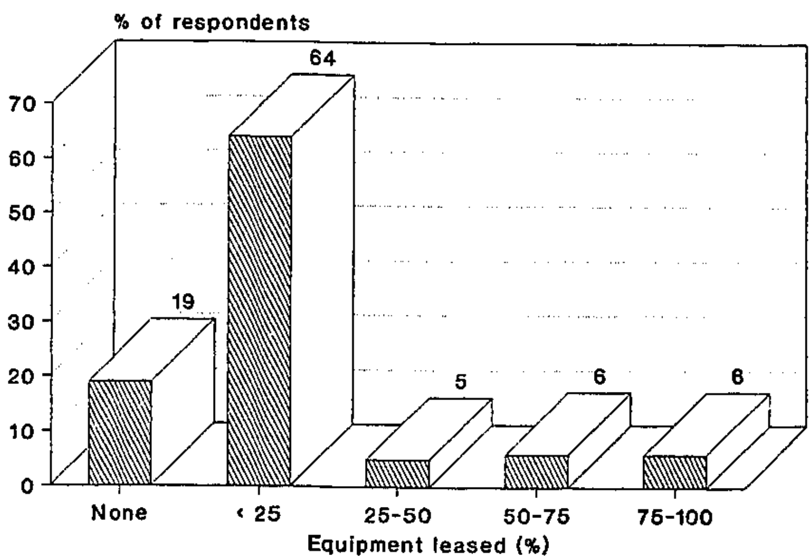


Fig. 5.8: % of construction equipment leased.

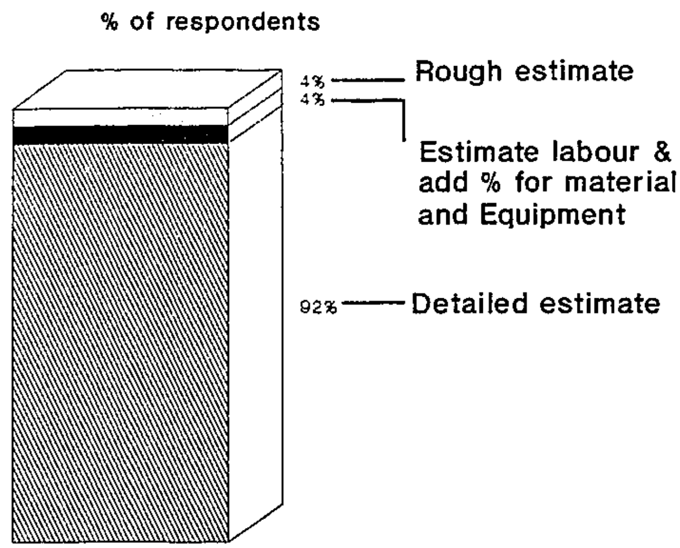


Fig. 5.9:Contractor's direct cost estimation.

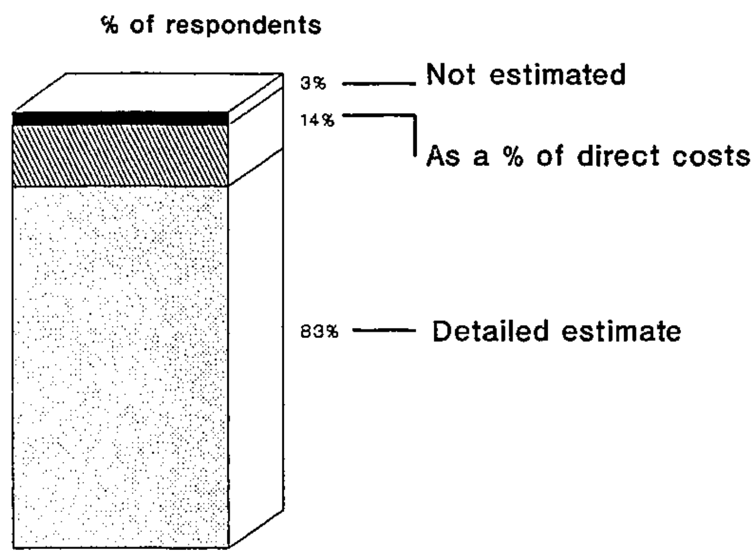


Fig. 5.10: Estimation of project overhead costs.

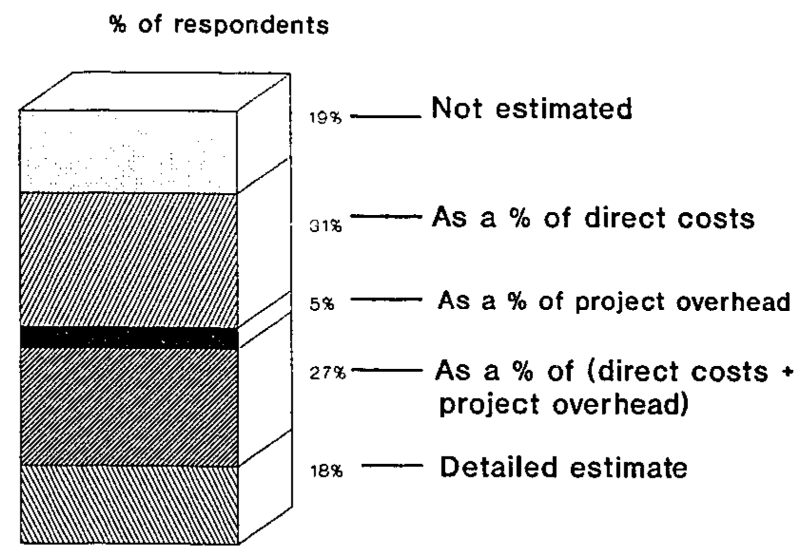


Fig. 5.11: Estimation of general overhead costs.

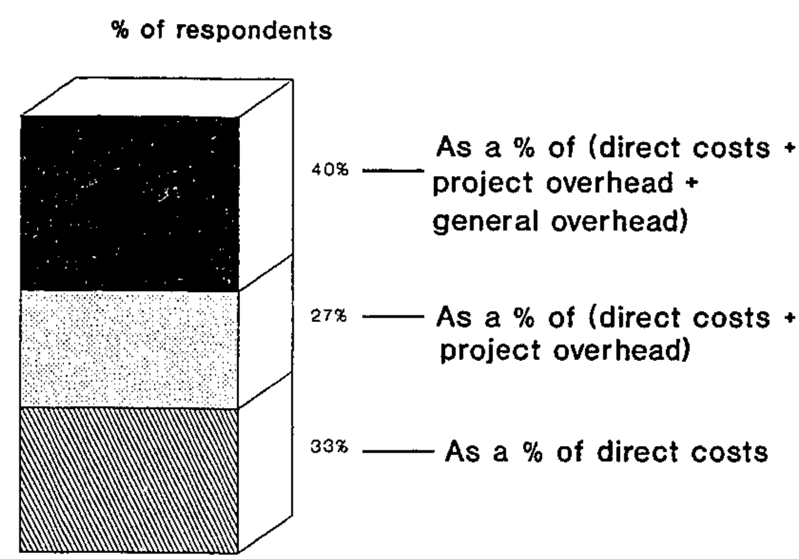


Fig. 5.12: Contractor's markup definitions.

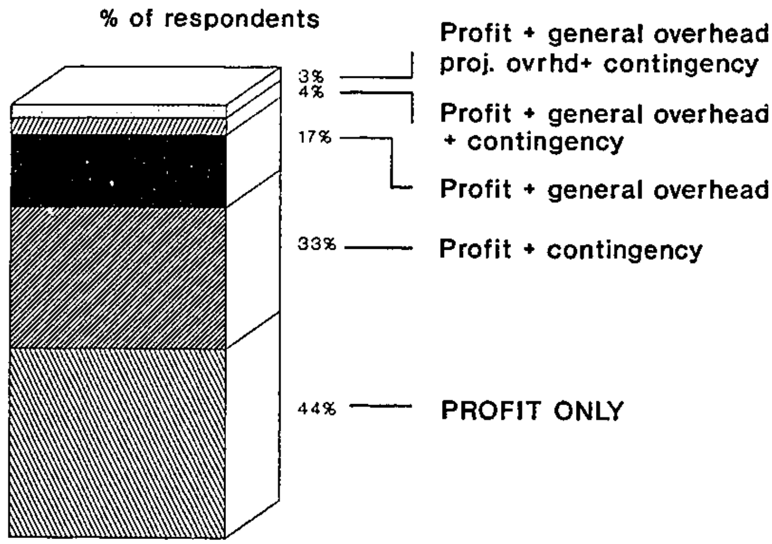


Fig. 5.13: Markup components.

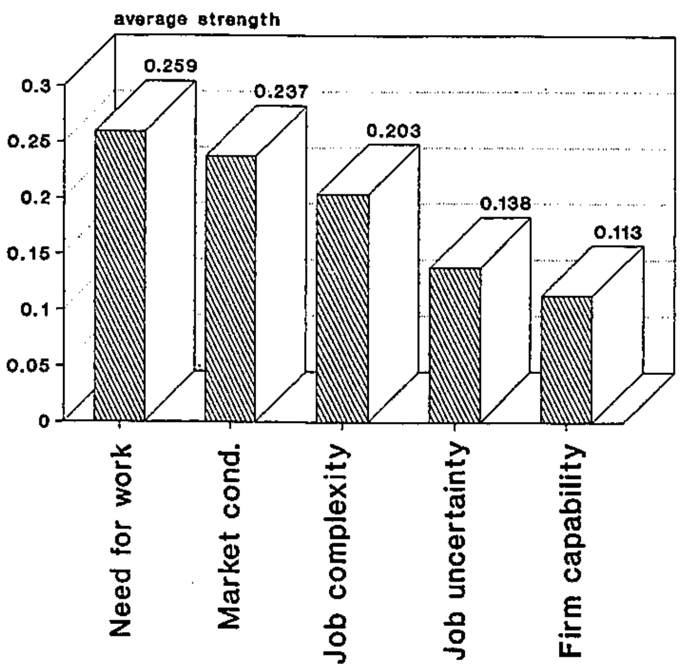


Fig. 5.14: Factors affecting % markup.

- Contractors estimate the project general overhead costs in mainly three ways: 1) as a percent of the direct cost (31% of respondents); 2) as a percent of direct cost + overheads (27%); or 3) detailed estimates (18%). Also, 19% of contractors do not estimate these costs as an isolated cost item (see Fig. 5.11). This may be combined in a markup %.
- Contractors adopt three, almost likely, definitions for the percent markup: 1) a % of total costs (direct + project overheads + general overheads) - 40% of respondents; 2) a percent of direct cost (33%); or 3) a percent of (direct cost + project overheads) - 27%), see Fig. 5.12. These differences exemplify the lack of standardized definitions and the difference in contractors' practices with respect to markup estimation.
- Contractors perceive the percent markup to include allowance for: 1) profit only (44% of respondents); 2) profit + contingency (33%); or 3) profit + general overhead (17%), see Fig. 5.13.

With respect to the factors affecting markup estimation, Fig. 5.14 shows the average strength of the five main factors provided in the questionnaire for contractors to address their relative importance. It can be seen that, on average, "Need for work" has the highest relative importance (26%) followed by "Market conditions" (24%), "Job complexity" (20%), "Job uncertainty" (14%), and "Firm capability" (11%). The majority of the survey respondents have not added additional factors to those provided in the questionnaire. However, some contractors have added factors such as "Competition" and "project duration" which although were implicit in the five main factors, contractors may have perceived them as important and their relationship with the markup need to be explicit.

With respect to the bidding examples elicited from the contractors' past records, the total number of example projects received is 103; some contractors have provided more than one example projects. These 103 example projects were not all suitable for use in training the neural network models. Contractors, occasionally, did not provide some necessary information for the example project to be suitable as a learning example. For instance, some contractors did not provide an overall rank of one or more sub-network. This may be due to the contractors' different practice regarding the assessment of the factors. In such cases, the example is not suitable for training the respective sub-network or the global network in the hierarchical model. Also, some information, including the markup value, may have been considered as confidential, and thus was not provided for some example projects. Thus, in order to identify suitable training and testing (validation) examples for the individual networks, the 103 examples were checked for completeness and suitability. Based on this scrutiny, the number of examples that were suitable for training and testing the neural networks are listed in Table 5.2. The data pertaining to these training and testing examples are provided in Appendix II. It should be noted that these data are not presented to the neural networks in the raw form given in the appendix, rather suitably scaled (Table 5.1

shows the scaling operation suitable for the different attributes). These scaling operations transform the raw data into continuous, rather than binary, values suitable for neural network representation (see Fig. 4.11 part b).

Table 5.2: Number of training and testing examples.

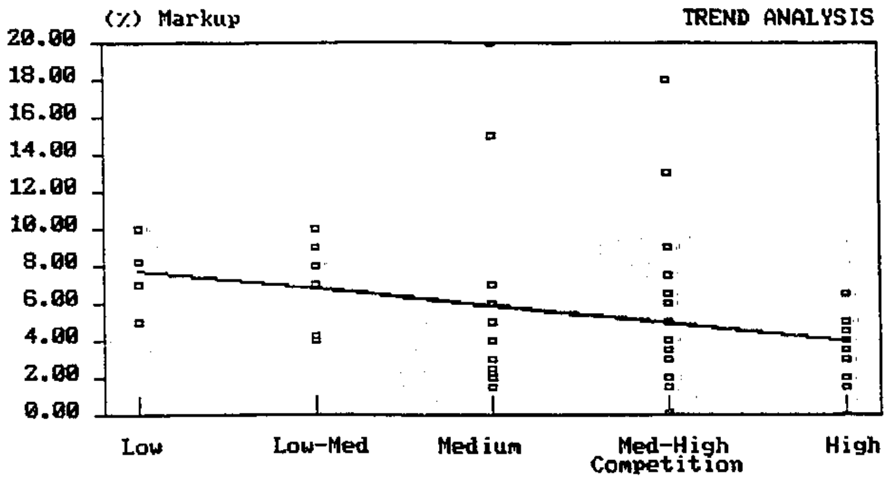
Model	Training Examples	Testing Examples
Single-Network Model	65	7
Hierarchical Model:		
Sub-net 1 (Uncertainty)	58	——
Sub-net 2 (Job Comlex.)	61	——
Sub-net 3 (Market)	48	——
Sub-net 4 (Ability)	38	——
Global Network	51	25

As shown in Table 5.2, no examples were retained for testing the sub-networks in the hierarchical model, allowing these small size networks to be trained on larger number of training examples and thus have better performance. Rather than testing the ability of individual sub-networks to produce less-important sub-goals, testing the performance of the global network shows the collective functioning of the hierarchical model to learn and generalize the system outputs. It should be noted, however, that although the learning examples used are complete in terms of information content, it is assumed that they are not free of biased judgements and inconsistencies. The contractor providing the historical example may be biased towards his recent experience and his retained knowledge pertaining to that example. Such types of errors represent noisy training data to the neural network models.

5.3.3 Validation of Training Data

Despite the noise inherent in the data, the training examples used, in every respect, provide a well training environment that reflect current industry practices and accounts for many quantitative and qualitative factors used by contractors in this domain. A simple test is conducted on the data used for training the neural networks. The test is to examine the relationship between the data pertaining to an input attribute and those of an output attribute, depicted in all the training examples. These relationships (or general trends) are established through simple regression analysis and then compared with known industry heuristics to test the information content of the training examples and their compatibility with current industry practice. Figures 5.15 to 5.30 illustrate the results of such analysis on the 65 training examples of the single network model. It is clear that the training examples, despite the presence of noisy data, exhibit trends that are in good agreement with industry heuristics. This includes:

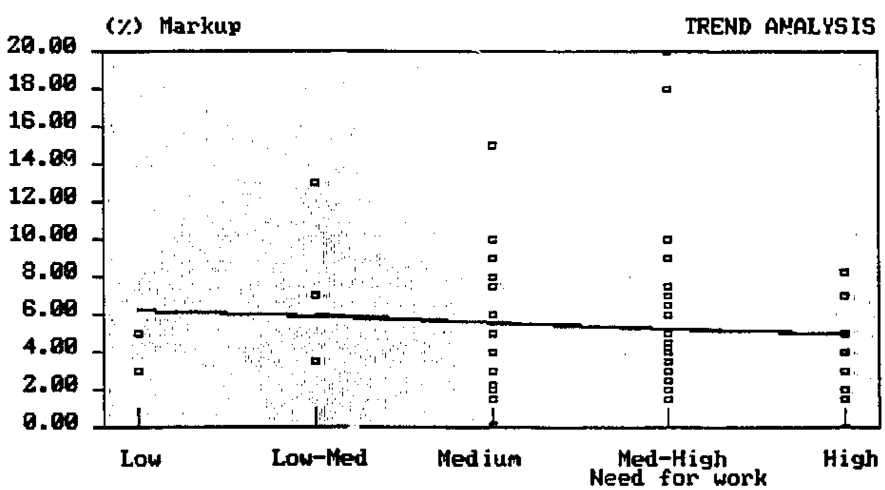
- Percent markup decreases with increase in competition (Fig. 5.15).
- Percent markup decreases with increase in need for work (Fig. 5.16).
- Percent markup is higher for public than private work (Fig. 5.17).
- Percent markup decreases with increase in contract duration (Fig. 5.18).
- Percent markup decreases with increase in company size (Fig. 5.19).



REG. FUNCTION: Y = 8.669022 -.9267812 * X

No. of data points: 65 Note: Some points may superimpose.

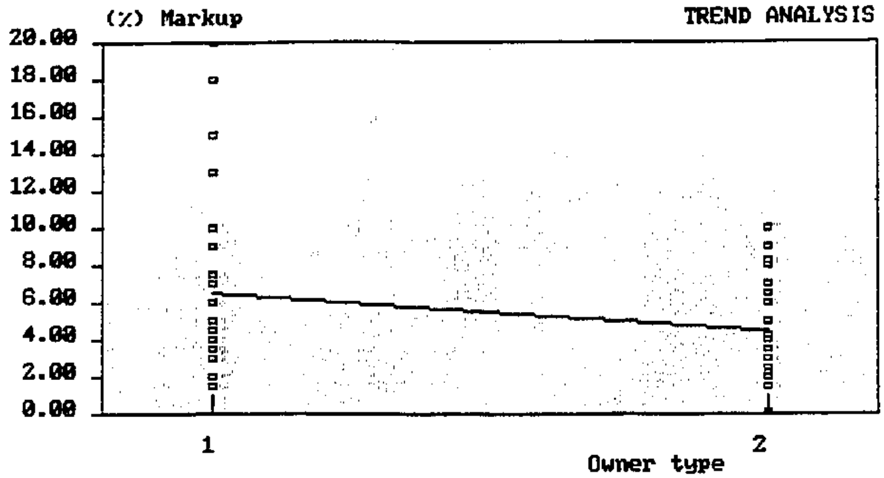
Fig. 5.15: Relationship between markup and competition.



REG. FUNCTION: Y = 6.617006 -.334879 * X

No. of data points: 65 Note: Some points may superimpose.

Fig. 5.16: Relationship between markup and need for work.



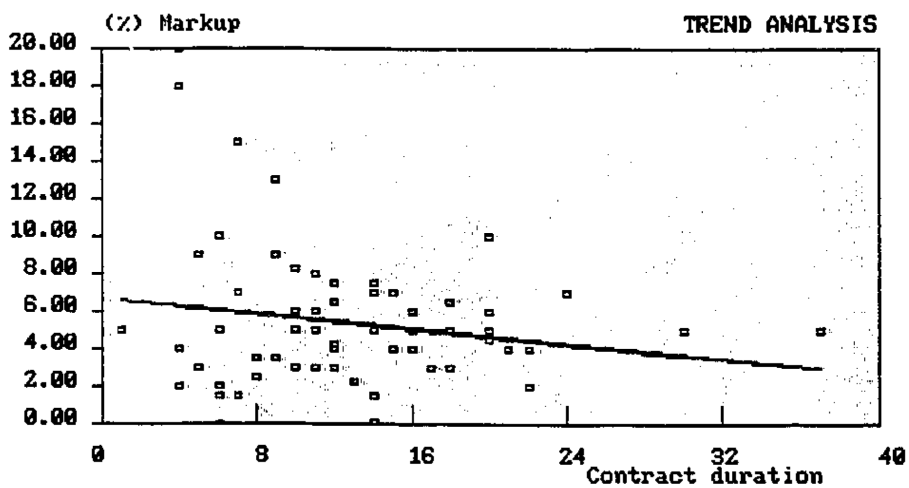
X axis :

1 = Public
2 = Private

REG. FUNCTION: Y = 8.627294 -2.092813 * X

No. of data points: 65 Note: Some points may superimpose.

Fig. 5.17: Relationship between markup and owner type.

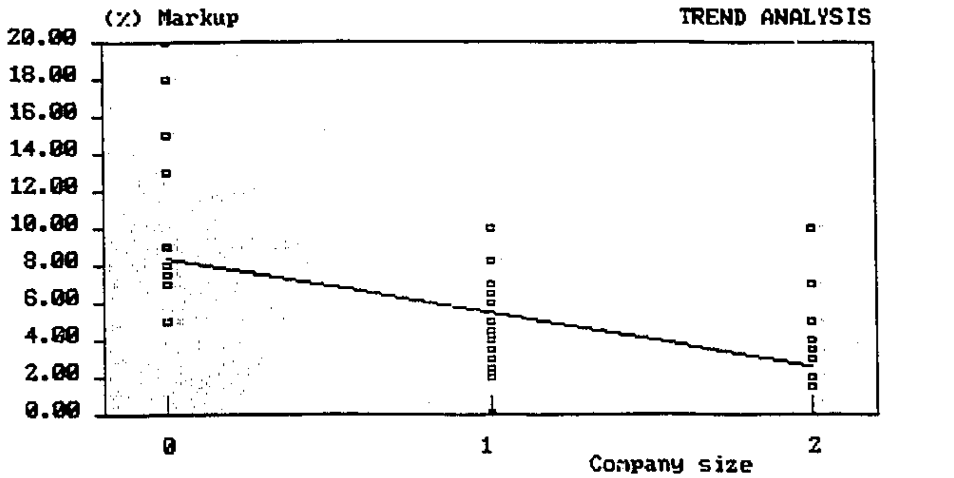


X axis : (Months)

REG. FUNCTION: Y = 6.704558 -.1008124 * X

No. of data points: 65 Note: Some points may superimpose.

Fig. 5.18: Relationship between markup and contract duration.



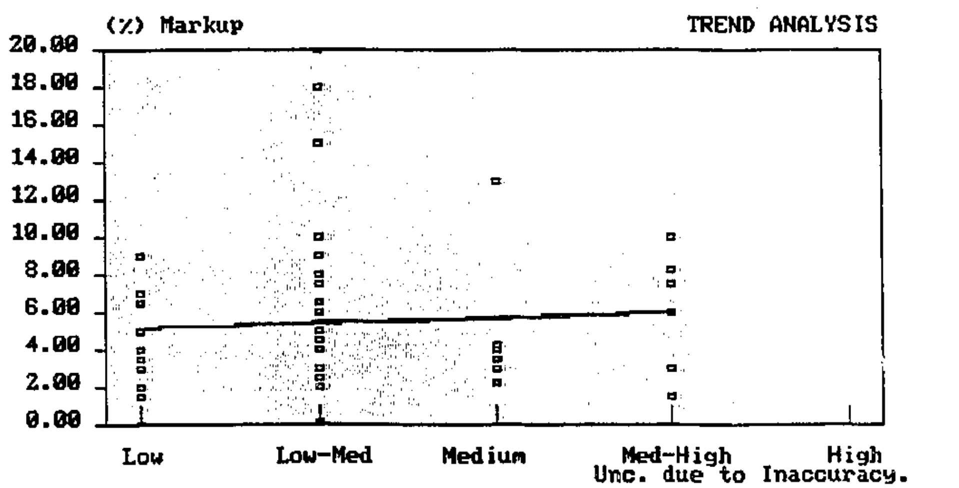
X axis :

0 = Small
1 = Medium
2 = Large

REG. FUNCTION: Y = 8.31773 -2.854516 * X

No. of data points: 65 Note: Some points may superimpose.

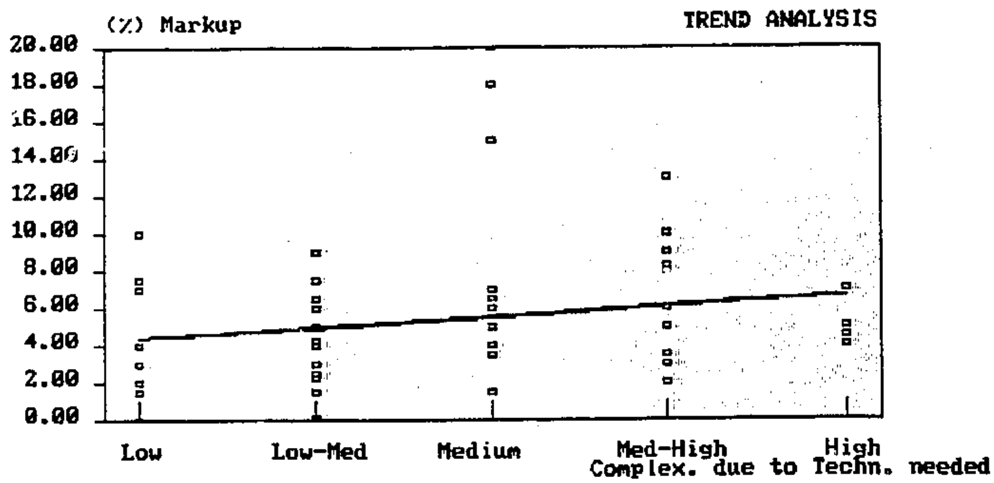
Fig. 5.19: Relationship between markup and company size.



REG. FUNCTION: Y = 4.782426 + .2919864 * X

No. of data points: 65 Note: Some points may superimpose.

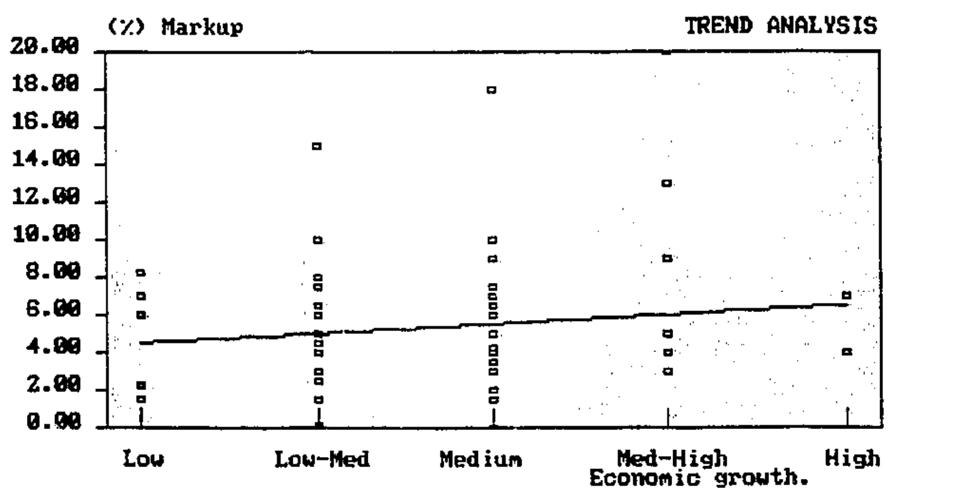
Fig. 5.20: Relationship between markup and uncertainty due to inaccurate estimate.



REG. FUNCTION: Y = 3.740144 + .5872412 * X

No. of data points: 65 Note: Some points may superimpose.

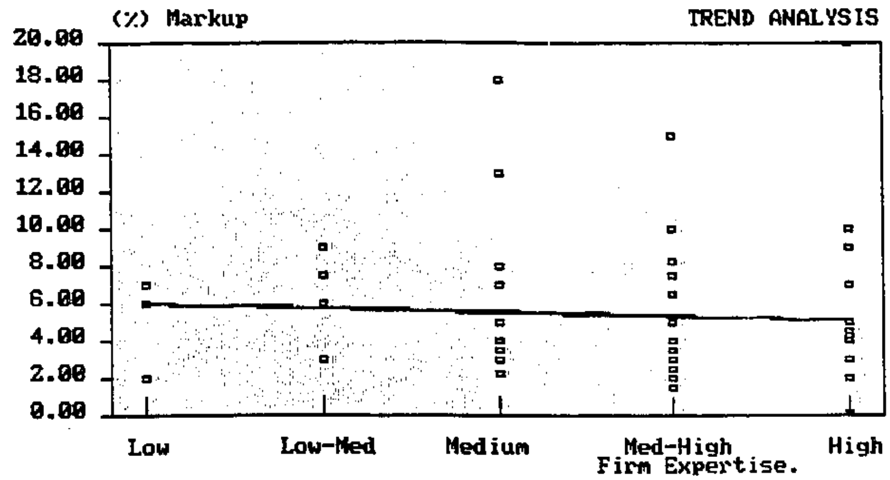
Fig. 5.21: Relationship between markup and complexity due to technology needed.



REG. FUNCTION: Y = 3.930489 + .5122485 * X

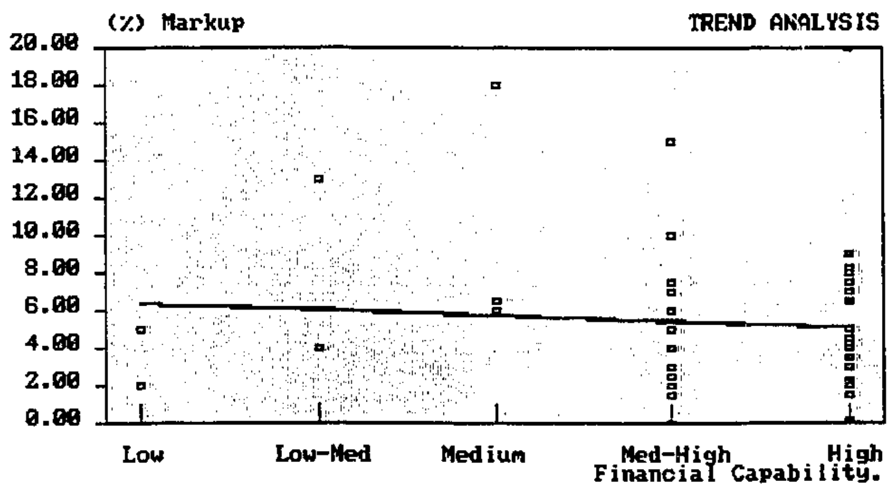
No. of data points: 65 Note: Some points may superimpose.

Fig. 5.22: Relationship between markup and economic growth.



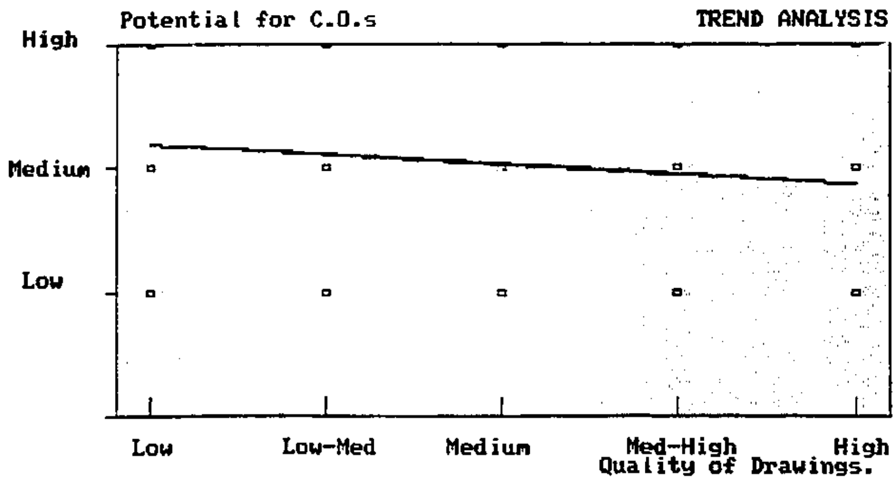
REG. FUNCTION: Y = 6.271564 -.2330058 * X
No. of data points: 65 Note: Some points may superimpose.

Fig. 5.23: Relationship between markup and firm expertise.



REG. FUNCTION: Y = 6.748116 -.3198125 * X
No. of data points: 65 Note: Some points may superimpose.

Fig. 5.24: Relationship between markup and firm financial capability.

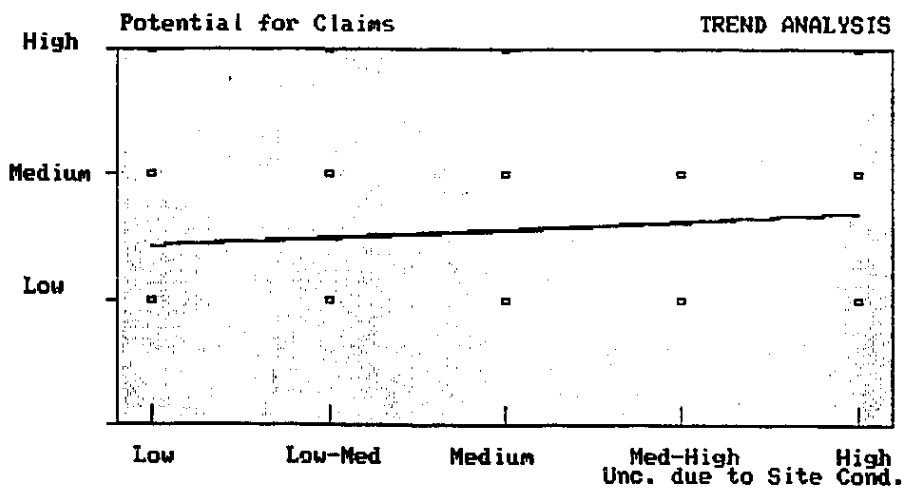


Low : 6 points
Low-Medium : 21 points
Medium : 20 points
Medium-High : 12 points
High : 6 points

REG. FUNCTION: Y = 2.282407 -8.256164E-02 * X

No. of data points: 65 Note: Some points may superimpose.

Fig. 5.25: Relationship between potential for change orders and quality of design drawings.

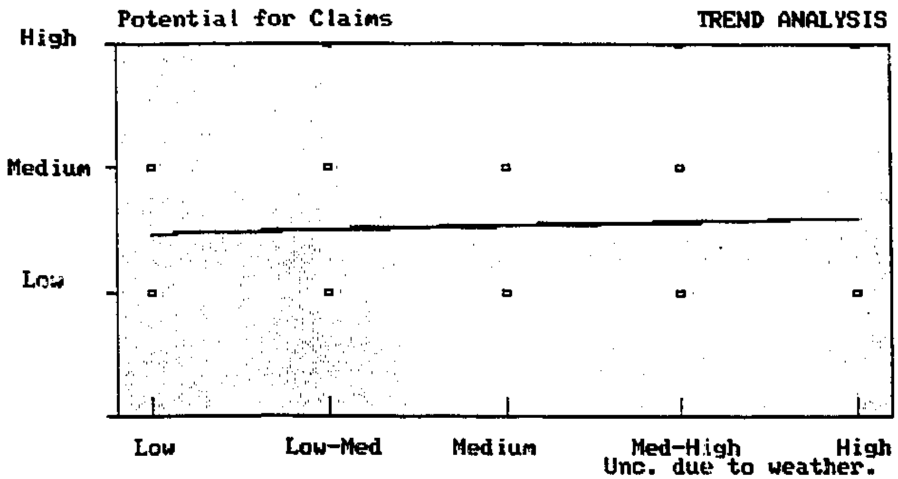


Low : 15 points
Low-Medium : 16 points
Medium : 20 points
Medium-High : 9 points
High : 5 points

REG. FUNCTION: Y = 1.354331 + 6.528865E-02 * X

No. of data points: 65 Note: Some points may superimpose.

Fig. 5.26: Relationship between potential for claims and uncertainty due to site conditions.

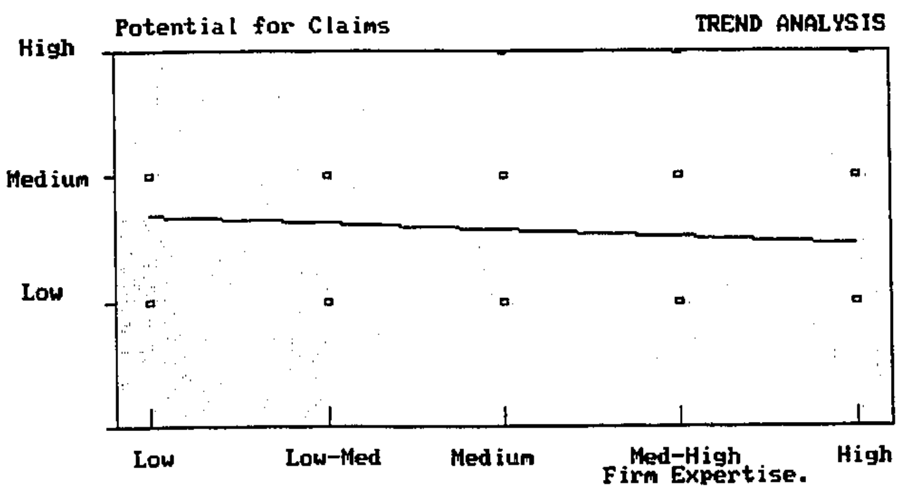


Low : 12 points
Low-Medium : 17 points
Medium : 16 points
Medium-High : 13 points
High : 7 points

REG. FUNCTION: Y = 1.433223 + 3.226778E-02 * X

No. of data points: 65 Note: Some points may superimpose.

Fig. 5.27: Relationship between potential for claims and uncertainty due to weather.

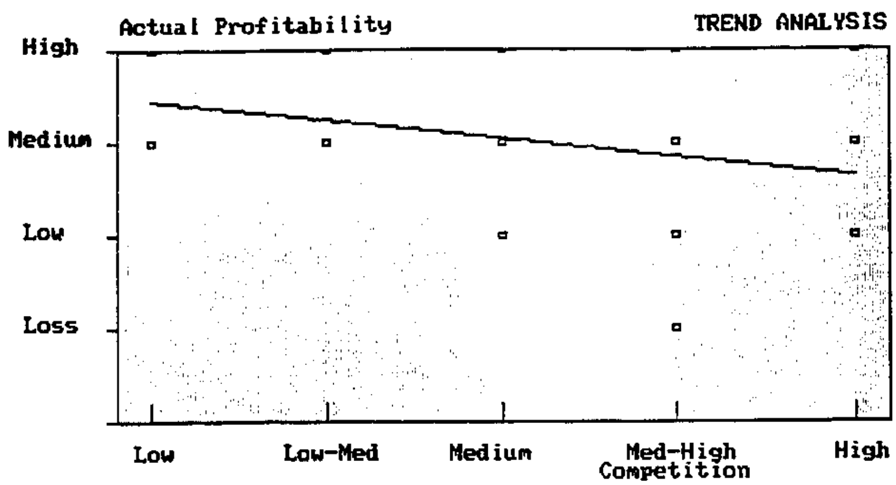


Low : 3 points
Low-Medium : 7 points
Medium : 10 points
Medium-High : 22 points
High : 23 points

REG. FUNCTION: Y = 1.73523 -5.515968E-02 * X

No. of data points: 65 Note: Some points may superimpose.

Fig. 5.28: Relationship between potential for claims and firm expertise.

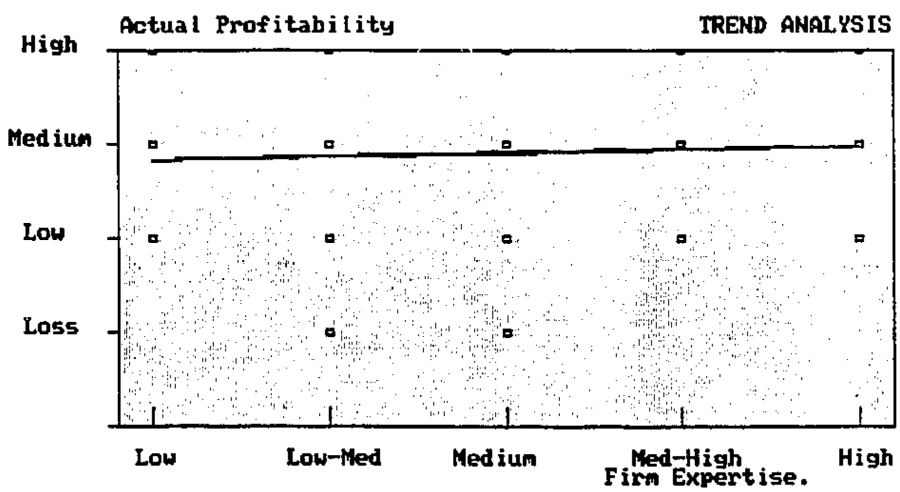


Low	: 5 points
Low-Medium	: 6 points
Medium	: 17 points
Medium-High	: 22 points
High	: 15 points

REG. FUNCTION: Y = 2.656184 -.2019566 * X

No. of data points: 65 Note: Some points may superimpose.

Fig. 5.29: Relationship between actual profitability and competition.



Low	: 3 points
Low-Medium	: 7 points
Medium	: 10 points
Medium-High	: 22 points
High	: 23 points

REG. FUNCTION: Y = 1.787899 + 3.914621E-02 * X

No. of data points: 65 Note: Some points may superimpose.

Fig. 5.30: Relationship between actual profitability and firm expertise.

- Percent markup increases with increase in estimate inaccuracy (Fig. 5.20).
- Percent markup increases with increase in job complexity due to technology needed (Fig. 5.21).
- Percent markup increases with increase in economic growth (Fig. 5.22).
- Percent markup decreases with increase in firm expertise (Fig. 5.23).
- Percent markup decreases with increase in financial capability (Fig. 5.24).
- Potential for change orders decrease with increase in the quality of design and drawings (Fig. 5.25).
- Potential for claims increase with increase in uncertainty due to site conditions (Fig. 5.26).
- Potential for claims increase with increase in uncertainty due to weather conditions (Fig. 5.27).
- Potential for claims decrease with increase in firm expertise (Fig. 5.28).
- Actual profitability decrease with increase in competition (Fig. 5.29).
- Actual profitability increase with increase in firm expertise (Fig. 5.30).

5.3.4 Hand crafting of neural networks

Since the backpropagation NN paradigm is selected as the most suitable development environment, a backpropagation algorithm (Pao 1989) is coded and utilized in the present development. The algorithm, included in Appendix III, is written in "C" language and incorporates several desirable features, including: 1) a training procedure based on the generalized delta rule; 2) a sigmoid transfer function that suits the continuous nature of the data; 3) biases for the network P.E.s; and 4) a "momentum term" to the weight adjustment computations which, as described in chapter 4, speeds the training process. The transparency of the code, as compared to commercial software packages, allows simple modification and enhancements of the algorithm, similar to those described in Table 4.5, and further integration with other routines used to facilitate data entry and analysis of the results.

Once the network development paradigm is coded, the neural networks pertaining to the two alternative models are configured, trained, and tested according to the methodology of Fig. 4.21. These implementation steps could be carried out using one of 3 approaches (hand crafting, pruning, or genetic algorithms), see Table 4.6. The simplest of the three is hand crafting of a neural network configuration. Since the process is not straight forward, the heuristics described in Table 4.3 of chapter 4 are utilized to find suitable configuration for the different neural networks. Based on these heuristics, networks are generally configured as follows:

1 - Inputs and outputs are continuous values (real numbers).
2 - Transfer function is sigmoid.
3 - Learning algorithm is based on the generalized delta rule.
4 - Learning rate coefficient is to 0.7 and momentum coefficient is 0.9.
5 - Networks are fully connected.

6 - Number of hidden layers is 1.

7 - Number of P.E.s in input buffer are equal to the number of input attributes.

8 - Number of P.E.s in output layer are equal to the number of output attributes.

9 - Number of P.E.s in the hidden layer are determined based on a trial and error manner.

Having the first 6 parameters as constant for all the networks, a network configuration can be abbreviated as (a b c), indicating the number of P.E.s for the input buffer (a), hidden layer (b), and output layer (c) respectively. Parameters (a) and (c) can be easily established based on the problem formulation, while (b) determined by a process of trial and error in which the number of P.E.s in the hidden layer is varied and the corresponding network error monitored for a constant number of training iterations (e.g., 1000 at this early network configuration stage).

As an example, the sub-network for the assessment of job uncertainty produces an output of one attribute representing an overall ranking of the job uncertainty in response to an input pattern representing the six governing attributes of: site conditions, owner attitude, project location, safety hazard, inaccuracy in cost estimate, and work sensitivity to weather (see Fig. 5.2, sub-network 1). A training set consisting of the 58 examples (Table 5.2) was then prepared by scaling the row data (see Table 5.1). The sub-network is configured to have an input buffer of 6 P.E.s, each corresponds to one of the problem attributes and an output layer of 1 P.E., representing the overall uncertainty level. The optimum number of P.E.s in the hidden layer was then determined by varying the number of P.E.s in that layer and monitoring the network training error for a constant number of training iterations (cycles), as shown in Fig. 5.31. The network training error generally decreases with the increase in the number of P.E.s, however, this implies more processing time. Since the sub-network is of small size, 30 P.E.s were selected for the hidden layer, without much influence on processing speed. As such, the resulting network configuration is (6 30 1), indicating the number of P.E.s for the input, hidden, and output layers respectively.

Once the network configuration was determined, the network was subjected to a longer training session targeted at 30,000 training cycles (in general, this is problem-dependent and the number of training iterations can be established only by testing). The training session lasted almost 7 hours on a high speed IBM compatible personal computer PC (386 processor, 25 Mhz speed with a math co-processor), during which, the network error reduced significantly with the training cycles (see Fig. 5.32). In a fashion similar to the uncertainty sub-network, the three remaining sub-networks were configured as: (7 20 1) for the job complexity sub-network; (5 25 1) for the market conditions sub-network; and (4 15 1) for the firm ability sub-network. These sub-networks were then trained on their respective

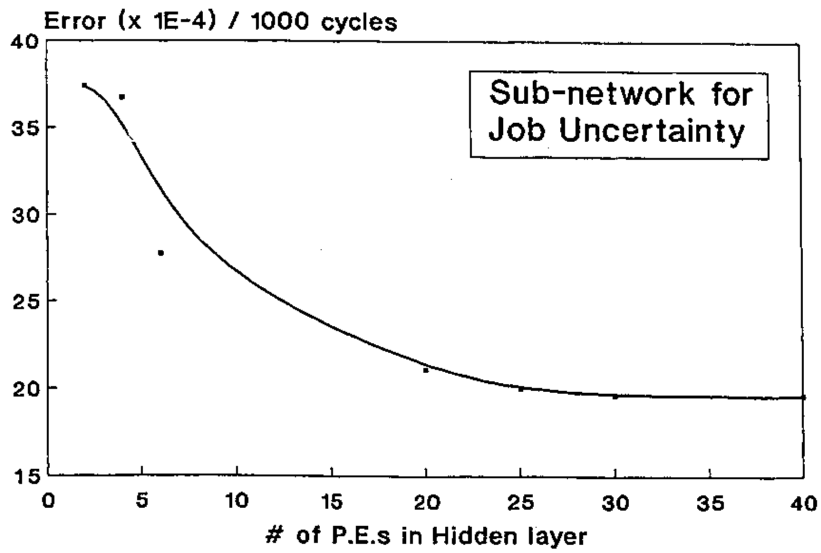


Fig. 5.31: Selecting the number of hidden layer P.E.s.

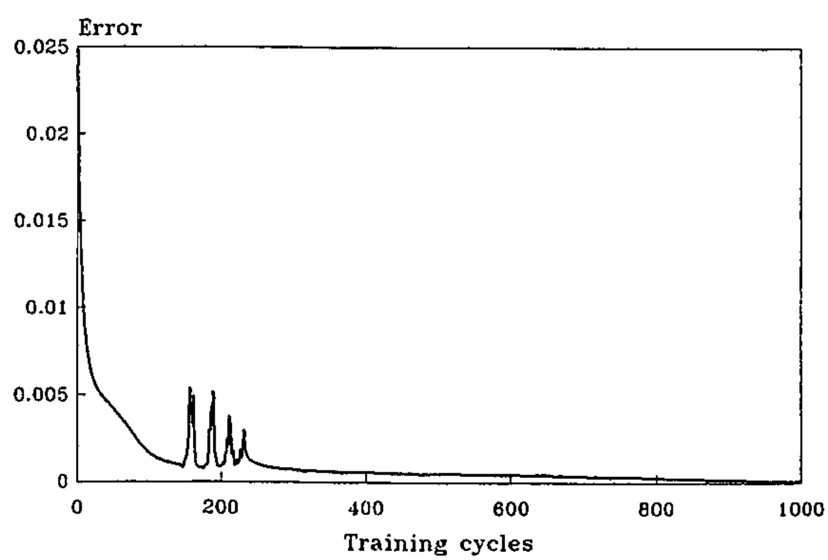


Fig. 5.32: Training the sub-network for uncertainty assessment.

training example. Training sessions took comparable processing times on the PC machine, resulting in almost zero errors on learning the training examples.

After the sub-networks have been successfully trained, the global network of the hierarchical model as well as the single-network were trained in a similar manner. Since the two networks have a larger number of input and output attributes, they involve more P.E.s and interconnections, making them relatively large in size and require a longer processing time to design and train. Thus, network configuration and training were conducted on a powerful Unix-based IBM workstation (RS/6000 Model 530 using AIX 3.1 Operating System). On one hand, the single-network was configured as (30 30 7) and trained on its 65 training examples for 100,000 training cycles. Training took almost 19 hours on the IBM workstation and the least mean square error of the network, accumulated over all the training examples, reached 0.000089 and could not decrease further. On the other hand, the global network of the hierarchical model was configured as (9 35 7) and trained on its 51 training examples for an identical number of training cycles (100,000). Training took less time (almost 10 hours on the IBM workstation), however, the least mean square error of the network stopped at a higher level (0.00634) and could not decrease further.

The next implementation step upon completion of required training is testing. The single-network and the global network of the hierarchical model were tested by introducing two sets of test examples: 1) the same training examples, however without their desired outputs; and 2) the test examples that were not used in the training process and the networks have never been exposed to. For the single-network, values pertaining to the 30 input attributes were fed to the network without their desired outputs. For the hierarchical model, inputs from the test examples pertaining to the sub-networks were fed to each sub-network, allowing them to produce respective overall ranks. These overall ranks, in addition to the other five attributes, were introduced to the global network without desired outputs. Based on the inputs, the single network and the global network produced actual outputs pertaining to the 7 output attributes which were compared to the desired outputs of the test examples used. Table 5.3 shows the results of two categories of examples (training and testing) on the networks' estimation errors (difference between the network actual outputs and the desired ones) pertaining to the 7 output attributes, averaged over all the examples used. Details of the errors pertaining to the individual training and testing examples are included in Appendix IV (parts a and b) with respect to the single-network and the global network respectively.

It can be seen from the results that the neural networks of the two models examined could learn the training examples with practically insignificant errors, despite the high level of noise inherent in the data of these examples. In the case of unseen data, the single-network model exhibits better generalization capabilities

Table 5.3: Training and testing errors of hand crafting trained networks.

Attribute No.	Attribute Name	Mean Error (%)	Standard Deviation	Error Range(%)
Model:	**Single-Network (30 30 7)**	**- Training.**		
1	Markup (%)	-0.25	0.58	3.67
2	Win/Lose	0.00	0.00	0.00
3	Difference ($)	-1.69	8.58	62.2
4	Potential for C.O.s	0.00	0.00	0.00
5	Potential for Claim	0.00	0.00	0.00
6	Duration Extension	0.04	0.07	0.35
7	Actual profitability	0.00	0.00	0.00
Model:	**Single-Network (30 30 7)**	**- Testing.**		
1	Markup (%)	15.1	67.19	182.4
2	Win/Lose	0.00	0.00	0.000
3	Difference ($)	-34.0	48.19	125.5
4	Potential for C.O.s	-4.76	29.16	100.0
5	Potential for Claim	19.1	75.29	233.3
6	Duration Extension	-3.80	11.62	36.46
7	Actual profitability	-28.6	38.54	116.6
Model:	**Hierarchical (9 35 7)**	**- Training.**		
1	Markup (%)	0.09	4.20	37.4
2	Win/Lose	0.00	0.00	0.00
3	Difference ($)	-18.7	39.6	148.5
4	Potential for C.O.s	0.00	0.00	0.00
5	Potential for Claim	0.00	0.00	0.00
6	Duration Extension	0.02	0.06	0.32
7	Actual profitability	0.00	0.00	0.00
Model:	**Hierarchical (9 35 7)**	**- Testing.**		
1	Markup (%)	-48.6	72.0	345.4
2	Win/Lose	87.6	132.4	500.0
3	Difference ($)	-78.0	40.3	119.8
4	Potential for C.O.s	-8.71	74.4	280.0
5	Potential for Claim	78.4	476.5	2475.0
6	Duration Extension	76.0	222.8	1037.2
7	Actual profitability	27.3	118.5	580.0

than the hierarchical model. This may stem from the larger number of training examples that were used for its training. Also, the less accurate generalization of the hierarchical model may stem from the difficulty that might have faced contractors in grouping the factors and imposing an overall rank associated with each group, thus, reducing the sensitivity of those factors and making their direct impact on the markup implicit (Moselhi and Hegazy 1993a). Despite its better results, the generalization errors of the single-network model are still relatively high (i.e., although average markup estimation error is reasonable at 15.11%, standard deviation is high at 67.1). This represents a generalization problem that is inherent in the development of neural network applications, particularly using the Backpropagation paradigm. In order to improve the network generalization capabilities, several measures are examined (Fig. 4.21). These include: 1) modifying the network configuration obtained by hand crafting; and 2) rather than network hand crafting, using an automated network optimization technique such as genetic algorithms.

5.3.5 Generalization improvement using hand crafting

Three experiments were conducted to test their impact on generalization: 1) varying the number hidden layers in the single-network model and their respective number of P.E.s; 2) varying the number of output attributes used; and 3) modifying the structure of the hierarchical model. The first aspect examined is the effect of using a larger network configuration (i.e., increasing the number of hidden layers) on the generalization performance of the single-network (as the model with higher potential for improvement). Two networks were then constructed by re-configuring the single-network to have 2 and 3 hidden layers respectively. The number of hidden layers' P.E.s pertaining to the two networks were identified in an iterative manner similar to the process of identifying the optimum number of P.E.s for a single hidden layer. Accordingly, the configuration is (30 35 30 7) and (30 35 30 20 7) for the networks with 2 and 3 hidden layers respectively. The two networks were then trained on the 65 examples for 30 and 48 hours respectively per the 100,000 training iterations on the IBM workstation. After training, the networks were tested as previously described. The resulting estimation errors, pertaining to the 7 output attributes, are shown Table 5.4. Similar to the one-hidden-layer configuration, the results of Fig. 5.4 show that the two networks with 2 and 3 hidden layers could learn the training examples with high degree of accuracy. However, upon testing the networks on the unseen (test) examples, the network with 2 hidden layers showed generalization performance close to that of the one hidden layer, while the network with 3 hidden layers was worst. These results indicate that, for the problem in hand, changing the number of hidden layers and their respective P.E.s have not led to improvement in generalization capabilities, rather substantially increased the training time (from 19 hours to 30 and 48). The lack of improvement with increase in number of hidden layers may be attributed to adequate representation of the input attributes used in the present application. In situations where the inputs are not clearly defined as a result of inherent problem characteristic, the additional hidden layers coupled with a higher level of abstraction may prove beneficial (Caudill 1990; Dutta and Shekhar 1988).

Table 5.4: Effect of the number of hidden layers on generalization performance.

Attribute No.	Attribute Name	Mean Error (%)	Standard Deviation	Error Range(%)
Model:	**Single-Network 2 Hidden layers (30 35 30 7)**	**- Training.**		
1	Markup (%)	-0.01	0.136	0.80
2	Win/Lose	0.00	0.00	0.00
3	Difference ($)	-9.12	27.9	102.9
4	Potential for C.O.s	0.00	0.00	0.00
5	Potential for Claim	0.00	0.00	0.00
6	Duration Extension	-0.04	0.04	0.14
7	Actual profitability	0.00	0.00	0.00
Model:	**Single-Network 2 Hidden layers (30 35 30 7)**	**- Testing.**		
1	Markup (%)	15.6	65.81	202.7
2	Win/Lose	0.00	0.000	0.000
3	Difference ($)	-41.1	50.94	109.7
4	Potential for C.O.s	7.14	41.65	150.0
5	Potential for Claim	16.7	40.83	133.3
6	Duration Extension	-2.14	6.400	20.82
7	Actual profitability	-26.2	24.97	66.60
Model:	**Single-Network 3 Hidden layers (30 35 30 20 7)**	**- Training.**		
1	Markup (%)	-0.23	0.54	4.32
2	Win/Lose	0.00	0.00	0.00
3	Difference ($)	-1.79	11.4	93.7
4	Potential for C.O.s	0.00	0.00	0.00
5	Potential for Claim	0.00	0.00	0.00
6	Duration Extension	-0.08	0.10	0.43
7	Actual profitability	0.00	0.00	0.00
Model:	**Single-Network 3 Hidden layers (30 35 30 20 7)**	**- Testing.**		
1	Markup (%)	40.2	110.3	357.6
2	Win/Lose	0.00	0.00	0.000
3	Difference ($)	-46.2	45.0	103.8
4	Potential for C.O.s	-7.14	17.5	50.00
5	Potential for Claim	33.3	85.91	266.6
6	Duration Extension	-1.58	6.59	18.51
7	Actual profitability	-19.1	44.03	116.66

The second aspect examined in developing networks with better generalization performance is the effect of the number of output attributes used in the single-network model. A network with less number of output attributes may become more specialized and less confused about the output(s) it produces. For the problem in hand, since the percent markup is the main output desired to be produced by the model, a network with one output attribute was constructed and the training and testing examples prepared. The network was configured as (30 10 1), having one hidden layer with 10 P.E.s as the optimum number obtained through the iterative process. The network was then trained on the 65 training examples and tested. Results of training and testing errors (Table 5.5) reveal that although adequate training errors are produced, generalization performance is not improved. This may stem from the nature of data used in training and the method by which the number of P.E.s for the hidden layer are obtained. The training data inherently have considerable amount of noise, requiring a network that is large enough (i.e., have large number of hidden P.E.s and interconnections) to enable the filtering of such noise and the extraction of the characteristic features of the data, leading to better generalization. Since the heuristics used to determine the number of hidden P.E.s produced only 10 P.E.s based on partial training, the network may have been smaller than what is needed for good generalization. The difficulty of the problem is then attributed to the lack of a more reliable and problem-independent heuristics for determining the optimum number of hidden P.E.s. For the network with 7 output attributes, larger number of interconnections are involved, making the network large in size which increases its likelihood of good generalization, particularly in the presence of noisy data.

Table 5.5: Effect of reducing the number of output attributes on generalization performance.

Attribute No.	Attribute Name	Mean Error (%)	Standard Deviation	Error Range(%)
Model:	**Single-Network 1 Hidden layer (30 10 1)**	**- Training.**		
1	Markup (%)	0.06	0.99	1.10
Model:	**Single-Network 1 Hidden layers (30 10 1)**	**- Testing.**		
1	Markup (%)	73.06	171.4	560.2

In order to investigate improvements to the hierarchical model and its generalization performance, a new structure for the model is proposed. Similar to the original structure, the system contains the same 4 sub-networks, pertaining to the assessment of job uncertainty, job complexity, market conditions, and firm capabilities. However, the global network is made to have 13 input attributes, as opposed to only 9 in the original structure. The input attributes are:

a) 9 attributes of original hierarchical model:

1 - Contract type.

2 - Contractor size.

3 - Markup definition.

4 - Markup components.

5 - Need for work

6 - Overall rank of job uncertainty.

7 - Overall rank of job complexity.

8 - Overall rank of market conditions.

9 - Overall rank of company abilities.

b) 4 additional attributes:

10- Owner type.

11- Competition.

12- Total bid value.

13- Contract duration.

The four additional attributes correspond to factors that the contractors participating in the questionnaire survey have viewed as important, although may have been implied by the overall ranks of the 4 sub-networks. Adding these factors as external to the sub-networks may strengthen their role in the model formulation and make their impact on markup explicit. The modified model was configured as (13 25 7) and trained on the 51 examples pertaining to the hierarchical model. Upon training and subsequent testing, the resulting estimation errors were tabulated as shown in Table 5.6. Similar to the original hierarchical model with 9 input attributes, the network could learn the training examples adequately, however, did not exhibit improvement to the generalization performance.

Table 5.6: Performance of the modified hierarchical model.

Attribute No.	Attribute Name	Mean Error (%)	Standard Deviation	Error Range(%)
Model:	**Hierarchical 1 Hidden layer (13 25 7)**	**- Training.**		
1	Markup (%)	-0.20	0.40	2.29
2	Win/Lose	0.00	0.00	0.00
3	Difference ($)	-0.04	1.03	8.44
4	Potential for C.O.s	0.00	0.00	0.00
5	Potential for Claim	0.00	0.00	0.00
6	Duration Extension	-0.10	0.06	0.30
7	Actual profitability	0.00	0.00	0.00
Model:	**Hierarchical 1 Hidden layer (13 25 7)**	**- Testing.**		
1	Markup (%)	156.8	255.4	851.4
2	Win/Lose	-28.0	44.90	100.0
3	Difference ($)	-2.34	78.57	409.1
4	Potential for C.O.s	16.0	87.37	366.7
5	Potential for Claim	1.33	70.07	266.7
6	Duration Extension	2.66	16.74	74.59
7	Actual profitability	14.0	72.56	275.0

5.3.6 Generalization Improvement using Genetic Algorithms

It is clear from the above results that the modifications suggested based on hand crafting did not lead to improvement in the networks' generalization capabilities. Yet, the best network available is a single network, hand-crafted to have 30 input attributes and 7 output attributes. Further improvement of the network is then experimented with the genetic algorithms (GA) technique as an automated and problem-independent network optimization tool. It should be noted that network pruning, as an alternative to the GA technique, was not found useful in the present study. This is because the absolute values for the majority of the network weights and biases are substantially higher than zero, indicating no redundancy in the

P.E.s and the interconnection weights, which pruning could have eliminated.

The procedure of evolving an optimum neural network for a particular problem is previously described in chapter 4 (see Fig. 4.15). For the markup estimation problem in hand, an effort is made to guide the highly randomized nature of the technique utilizing the results obtained during network hand crafting. Thus, a network configuration of (30 30 7), as obtained by hand crafting, is used as a constant in the optimization, with the values for the network weights and biases as the parameters to be optimized. With 30 inputs, 30 hidden P.E.s, and 7 outputs, the number of optimization parameters are calculated as follows:

- Number of weights connecting input and hidden layers are 30 * 30 = 900.
- Number of weights connecting hidden and output layers are 30 * 7 = 210.
- Number of neuron biases for the hidden layer P.E.s = 30.
- Number of neuron biases for the output layer P.E.s = 7.

Total parameters = 1147.

These 1147 parameters are considered elements of a network "gene". A population of 8 networks and their associated genes were generated and utilized in the GA optimization, following the procedure outlined in Fig. 4.15. Four of these network genes were generated completely from random numbers in the range from (-5.0 to +5.0, traditionally assumed by researchers as good starting values). The other 4 genes were generated as variations from the weights and biases of the **"trained"** network produced by hand crafting. The weights and biases of the single-network were "shaken" to produce the 4 genes as follows:

- weights and biases are kept the same as the trained network.
- weights and biases ± random values in the range (0 to 5%) of their values.
- weights and biases ± random values in the range (0 to 10%) of their values.
- weights and biases ± random values in the range (0 to 15%) of their values.

Having the 8 networks with a (30 30 7) configuration and initial values for their weights and biases represented in their respective genes, the optimization was conducted in a process of training, testing, and offspring generation. First, each of the 8 networks were trained on the 65 examples for 1000 iterations. Training and testing on the unseen examples produced errors that are used to calculate a weighted error for each network. This weighted error represents the objective function which according to the network degree of minimizing it, the network and its associated gene is considered fit or unfit to survive. Since generalization over the % markup estimate is a main criteria for optimization, the weighted error for the network is formulated as a composite of the errors in training and testing, as follows (Hegazy and Moselhi 1993):

Error		Weight
Average error, in testing examples, of % markup estimation	*	0.4 +
Error range, in testing examples, of % markup estimation	*	0.4 +
Average error in testing, over all attributes and examples	*	0.1 +
Average error in training, over all attributes and examples	*	0.1

where, 90% of the network weighted error are attributed to errors in testing on unseen examples. In order to set a termination limits for the GA procedure, a desirable weighted error level for a network is calculated based on desirable limits of (10%, 30%, 5%, 1%) for the 4 error categories respectively. Then, a satisfactory level for the weighted error is = 0.1 * 0.4 + 0.3 * 0.4 + 0.05 * 0.1 + 0.01 * 0.1 = 0.166.

Once the weighted errors are calculated for the 8 networks, a merit of each network is calculated as (1 / weighted error). Based on the merit values, relative merits for the networks are calculated as:

$$\text{Relative merit}_i = \frac{merit_i}{\sum_{i-1}^{No.\ of\ nets} merit_i}$$

Using these relative merits, each gene is assigned a probability of being selected to produce an offspring proportional to its relative merit. After initial training, the weighted errors and respective relative merits of the 8 parent networks are shown in Table 5.7(a). Since none of the weighted errors satisfies the desirability limits,

Table 5.7: Genetic Algorithm population for the (30 30 7) network.

Population	Weighted Error	Relative Merit
(a) Before Offspring generation:		
Network 1	1.897377	7
Network 2	1.510249	9
Network 3	1.328633	11
Network 4	1.179697	12
Network 5	0.795535	19
Network 6	1.040386	14
Network 7	1.417171	10
Network 8	1.124485	13
(b) After GA optimization (189 generations):		
Network 1	0.360881	15
Network 2	0.440975	11
Network 3	0.387351	13
Network 4	0.370503	14
Network 5	0.442332	11
Network 6	0.440100	12
Network 7	0.418913	12
Network 8	0.440932	11

an offspring is generated with its gene formed by a crossover operation between 2 parent genes, selected according to their probabilities (see Fig. 4.16). The offspring is then trained for a random number of training iterations in the range (100 to 500) and subsequently tested and its weighted error calculated. If the weighted error of the offspring is lower than that of the worst parent network (i.e., offspring is more fit), the offspring becomes a parent gene (i.e., survives), replacing the worst gene. Accordingly, the new set of 8 parent networks are then checked for their relative merits. The process of generating new offsprings, training, and testing continued until all the parent networks almost became the same and their performance levelled off with no further improvement, even if the desirability limit is not achieved. During the optimization process, the weighted error, averaged over the 8 networks, decreased with the number of offsprings generated, as shown in Fig. 5.33. It can be seen that the performance levelled off after 189 offspring generations (this process took almost three days on a PC 486 processor, 33 MHZ speed) with the weighted errors and the relative merits pertaining to the 8 networks as shown in Table 5.7(b). The results of Table 5.7 show an improvement in the weighted error from 0.796 for the fifth parent network (the network trained based on hand crafting) before GA optimization to 0.361 for the first parent network after optimization, referred to later as the GA optimized network. The performance of the optimized network on the training and testing examples is presented table 5.8, with the details included in Appendix IV (part c). In comparison with the results of the hand crafting shown in Table 5.3, the testing errors of the optimized network exhibit improvement in markup estimation both on the average value (-6.11% as opposed to 15.11%) and in the standard deviation (25.87% as opposed to 67.19%). Testing errors pertaining to the other output attributes show comparable results on the average and slight improvement in the standard deviation. The reason for a higher rate of improvement in markup estimation than other attributes is eventually due to the formulation of the weighted error used. It is possible, however, to change the weighted error formulation

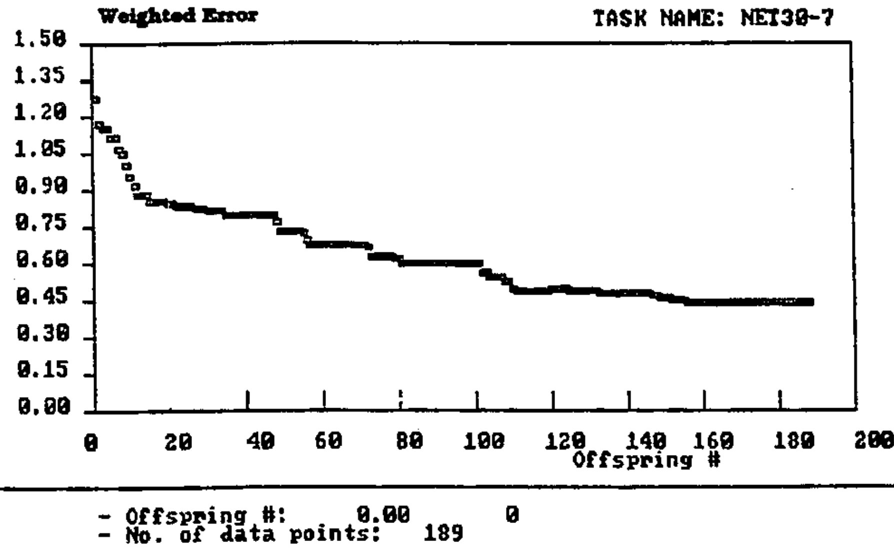


Fig. 5.33: Weighted error reduction during Genetic Algorithm optimization for the (30 30 7) network.

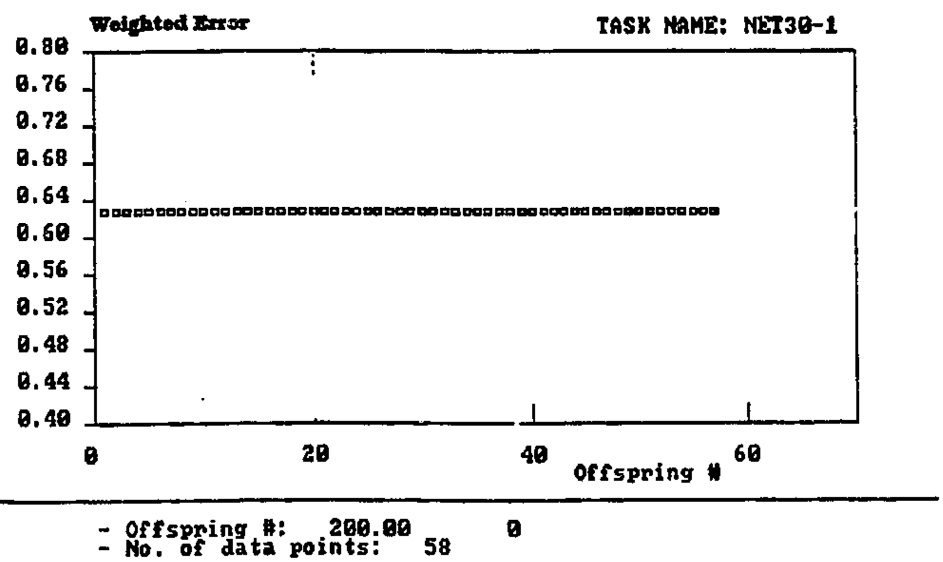


Fig. 5.34: Weighted error reduction during Genetic Algorithm optimization for the (30 10 1) network.

Table 5.8: Training and testing errors of the GA-trained network.

Attribute No.	Attribute Name	Mean Error (%)	Standard Deviation	Error Range(%)
Model:	**Network (30 30 7) after GA optimization.**	**- Training.**		
1	Markup (%)	-24.3	32.8	193.5
2	Win/Lose	0.00	0.00	0.000
3	Difference ($)	-25.8	44.1	163.2
4	Potential for C.O.s	6.15	27.0	150.0
5	Potential for Claim	-28.7	18.8	66.67
6	Duration Extension	-10.5	7.43	39.30
7	Actual profitability	35.6	41.5	150.0
Model:	**Network (30 30 7) after GA optimization.**	**- Testing.**		
1	Markup (%)	-6.11	25.87	77.49
2	Win/Lose	0.00	0.000	0.000
3	Difference ($)	-54.9	49.72	107.0
4	Potential for C.O.s	-21.4	24.74	50.00
5	Potential for Claim	-19.1	52.27	166.7
6	Duration Extension	-12.5	6.559	23.45
7	Actual profitability	-23.8	47.02	116.7

and repeat the GA optimization with an objective to generalize better on other attributes, thus possibly developing seven networks that suit the seven attributes. It is noted, however, that the improvement achieved in generalization on unseen examples comes with an expense. The GA optimization have resulted in higher errors on the training examples than those resulting from hand crafting. The performance of the optimized network is no more best fitting the training data (as exhibited by the hand crafted network), rather generalizes their inherent trends and performs acceptably, with respect to the heuristic and unstructured nature of the problem in hand, on unseen data. In this sense, the performance of the two networks could be comparable to those illustrated in chapter 4, Fig. 4.14.

In an effort to take advantage of the capabilities of the two networks, an experiment was conducted utilizing both networks. The two networks were fed by the same test examples pertaining to the two categories (those used in training and the unseen test examples), without the desired outputs. The actual outputs (i.e., responses) of both networks were then averaged and compared with the examples' desired outputs. The estimation errors are tabulated in Table 5.9 and their details included in Appendix IV (part d). It can be seen that the resulting errors exhibit some improvement over the GA optimized network with respect to the training examples with no loss in performance with respect to the unseen data.

Table 5.9: Results of averaging the hand-crafted and the GA-trained networks.

Attribute No.	Attribute Name	Mean Error (%)	Standard Deviation	Error Range(%)
Model:	**Network (30 30 7).**	**- Training.**		
1	Markup (%)	-11.6	19.6	137.6
2	Win/Lose	0.00	0.00	0.000
3	Difference ($)	-13.8	23.5	111.2
4	Potential for C.O.s	1.54	8.64	50.00
5	Potential for Claim	-13.1	16.7	50.00
6	Duration Extension	-5.21	3.72	19.72
7	Actual profitability	16.15	25.5	100.0
Model:	**Network (30 30 7).**	**- Testing.**		
1	Markup (%)	4.50	37.04	113.1
2	Win/Lose	0.00	0.000	0.000
3	Difference ($)	-44.5	44.51	103.2
4	Potential for C.O.s	-14.3	22.59	50.00
5	Potential for Claim	7.14	80.60	250.0
6	Duration Extension	-8.13	6.541	22.07
7	Actual profitability	-23.8	39.69	116.7

Using the genetic algorithms technique, two additional experiments were conducted on networks which seemed to have potential for improvement in generalization. These are: 1) the single network with 30 inputs and one output attribute for % markup estimation (i.e., 30 10 1); and 2) the global network of the hierarchical model, having 9 inputs and 7 outputs (i.e., 9 35 7). For each of the two experiments, a population of 5 neural networks were generated with their genes constructed from random values. The same optimization procedure was conducted for 58 and 81 offspring generations respectively(almost 24 hours of processing on the PC 486 machine). The networks, however, stabilized at high error levels (average weighted error for the 5 networks as 0.63 and 0.84 respectively, as opposed to the 0.361 value achieved by the 30 30 7 network discussed previously), see Fig. 5.34 and 5.35.

Based on the results obtained, the proposed model is made to utilize the two (30 30 7) networks of hand crafting and GA optimization, with the model output as the average of their responses. This, in addition to reducing the possibility of irrational outcomes, is likely to produce acceptable errors for projects that are close to the training examples and still generalizes adequately on others.

5.4 Interpretation of the Network Weights

Having the neural networks of the markup model being trained, values of the weights and biases, see Appendix V, encode the networks state of knowledge

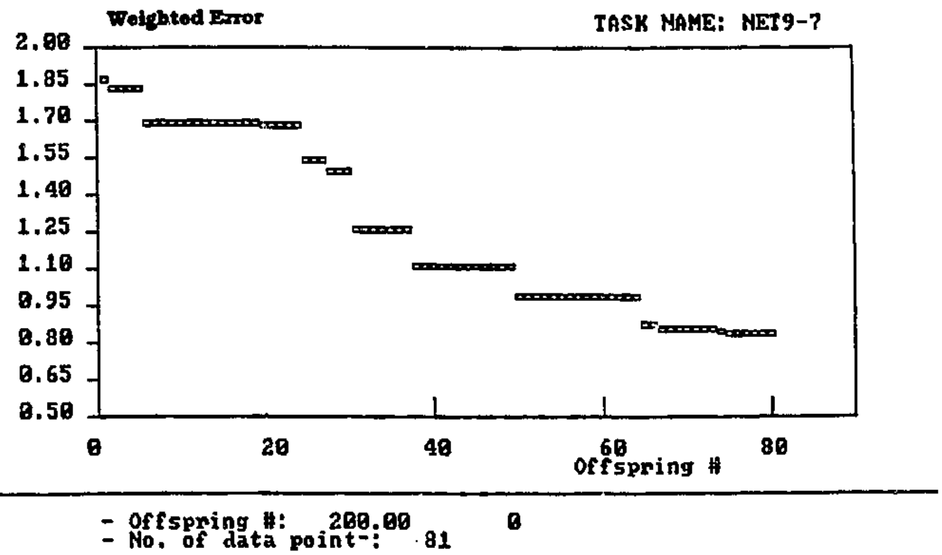


Fig. 5.35: Weighted error reduction during Genetic Algorithm optimization for the (9 35 7) Hierarchical network.

gained from the training examples used. Utilizing the neural network model for markup predictions on new projects is a matter of mathematical manipulation of these values with the responses produced almost instantaneously. Usually, no logical inferences or explanations are involved in providing the solution, making the solution process non-transparent and non logically traceably. This, as discussed in chapter 4, has contributed to "Black Box" image of neural networks which researchers have been trying to demystify.

In an effort to interpret the network weights and their underlying logic, the technique proposed by Garson (1991), see Eq. 4.17 and Fig. 4.19 in chapter 4, is utilized. Based on the technique, the relative importance of the 30 input attributes to the individual outputs are calculated, pertaining to the 2 neural networks that constitute the markup model, see Appendix VI. From the results obtained, it can be seen that:

- All the input attributes are highly relevant to the model formulation. The values for the relative importance vary between 1.2 for a least important to 5.0 for the most important input attribute. Thus, none of the input attributes could have been eliminated in the model formulation.

- The values for the relative weights determine the input attributes that govern the conclusions derived by the individual outputs. For instance, the network trained based on hand crafting showed that "Competition", "Contract duration", and "Uncertainty due to owner attitude" are the most influential factors on the % markup estimate.

- A certain input attribute impacts the individual system outputs with varying degrees. For instance, "Need for work" influences more the win/loose possibility, % markup, duration extension, potential for change orders, potential for claims, actual profitability, and the ($) difference between the winner and second bidder.

Therefore, the relative importance values obtained could be later utilized to detect causal rules and explanation facilities for the markup model.

5.5 Conclusion

This chapter presents the development of a markup estimation model using the backpropagation NN paradigm. The markup estimation process is analyzed and attributes governing the decision process identified. Two approaches for designing the neural network model have been examined and their results compared: 1) a single-network model; and 2) a hierarchical model. A questionnaire survey is developed to elicit knowledge pertaining to current bid preparation practices of general contractors in Canada and the U.S. The survey results are utilized to structure, design, train, and test the two alternative models. It became vivid from training the two models based on hand crafting that both were capable of learning efficiently from the set of example projects used in training. Testing both models on unseen project examples, however, showed that the single-network model exhibits better generalization capabilities than the hierarchical model, although the errors are still relatively high. Accordingly, the single-network was selected for the present application. Two methods were employed to improve the generalization capabilities of the selected model: the hand crafting technique and the genetic algorithms technique. While hand crafting was instrumental for the configuration of the network and its efficient training, it was less successful than the GA technique in improving the generalization capabilities. In order to guide the random search of the GA procedure, the results of hand crafting were used as initial inputs. Using the GA technique, the network weight and bias values are optimized having the optimization constraint geared towards better generalization. Accordingly, two NNs pertaining to the single-network structure are used to formulate the markup estimation model: 1) the network trained based on hand crafting; and 2) the network optimized based on the GA procedure. The output of the model being the average of the two networks' responses. Analysis of the two networks' weights were further conducted in order to identify the relative importance of the input attributes to the individual system outputs. These could then be used to interpret important causal rules that might be used to provide explanation capabilities during recall.

The present model is developed utilizing the knowledge of non-specific contractors. However, the contractor, user of the model, has to have the utility to store his own bidding experiences and utilize them to adapt the general model to his particular environment. This adds flexibility to the model use and enables it to learn and modify its performance based on new experiences. Developing such a utility is described in chapter 6. The developments made in this study with respect to problem structuring, knowledge acquisition, data validation, network hand crafting, generalization improvement, and GA optimization provide general guidelines in developing practical neural network applications. The NN approach described could be applied to other construction management problems that are solved in practice based mainly on analogy with previous situations and for which traditional algorithmic tools are inadequate.

CHAPTER 6

AN ANALOGY-BASED D.S.S. FOR BIDDING IN CONSTRUCTION

6.1 Introduction

This chapter describes the adaptation of the markup estimation model and its use as a decision support system (D.S.S.) for bidding in construction. The model has been coded in a user friendly software that facilitates storage of previous bid encounters, data input, model adaptation to the user's environment, computations, sensitivity analyses, assessment of the probability of winning, and further integration with other modules for planning and scheduling and cost estimation. The contractor (user of the model) is provided with a utility to store his own bid encounters in a database. The model can be re-trained to generalize on those experiences and adapt its performance accordingly. In estimating optimum markup for a new project, the uncertainty in the user assessment of the project risks (used as model inputs) is accounted for by a sensitivity analysis conducted using the Monte Carlo simulation technique. Such analysis produces a measure of the probability of winning at any desired level of markup. The capabilities of the software developed are demonstrated through an example application.

6.2 D.S.S. and Bidding Data

A computer program **"DBID"** has been developed encoding the NN markup estimation model described in chapter 5 as a decision support system (D.S.S.) for bidding in construction. The core of the program is two neural networks, both are trained on the industry general knowledge elicited from contractors in Canada and the U.S.: 1) the hand-crafted network; and 2); the GA-optimized network. Both networks accept as inputs data regarding the contractor assessment of various risks associated with a project, and produce as outputs 7 data elements pertaining to: optimum markup estimate (%), win/lose possibility; difference in ($) between the winner and second lowest bidder; anticipated actual profitability as high, medium, low or loss; project potential for change orders as high, medium, or low; project potential for claims as high, medium, or low; and expected duration extension.

A schematic diagram of program **DBID** is shown in Fig. 6.1, showing the system components. The menu system of program **DBID** is also shown in Fig. 6.2. Three categories of data are allowed to be stored in the system for use by the solution processing mechanisms. These three data categories, accessed through the main menu, pertain to: 1) company fixed data; 2) data pertaining to a set of projects for which the contractor is currently preparing bids; and 3) data pertaining to historical bid encounters (past projects). The first category represents organization-dependent data that is fixed irrespective of the projects being estimated or executed. These data (Fig. 6.3) relate to: a) firm size (small, medium, or large); b) contractor definition of markup, and thus his method of adjusting the bid based on a given % markup value; and c) contractor anticipation of the cost components

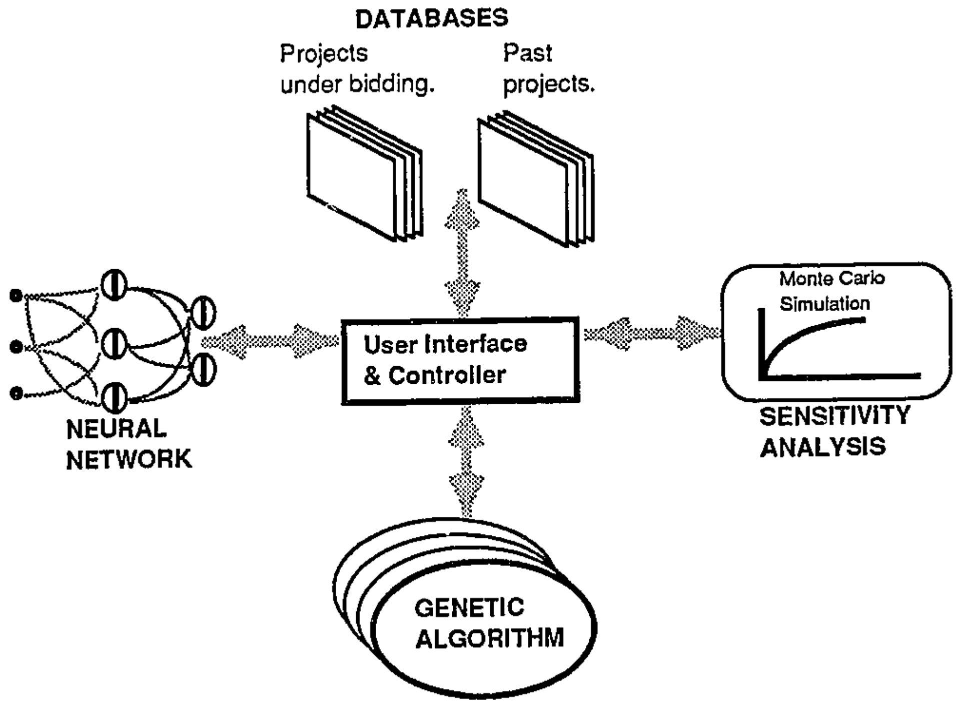


Fig. 6.1: A schematic diagram of program "DBID".

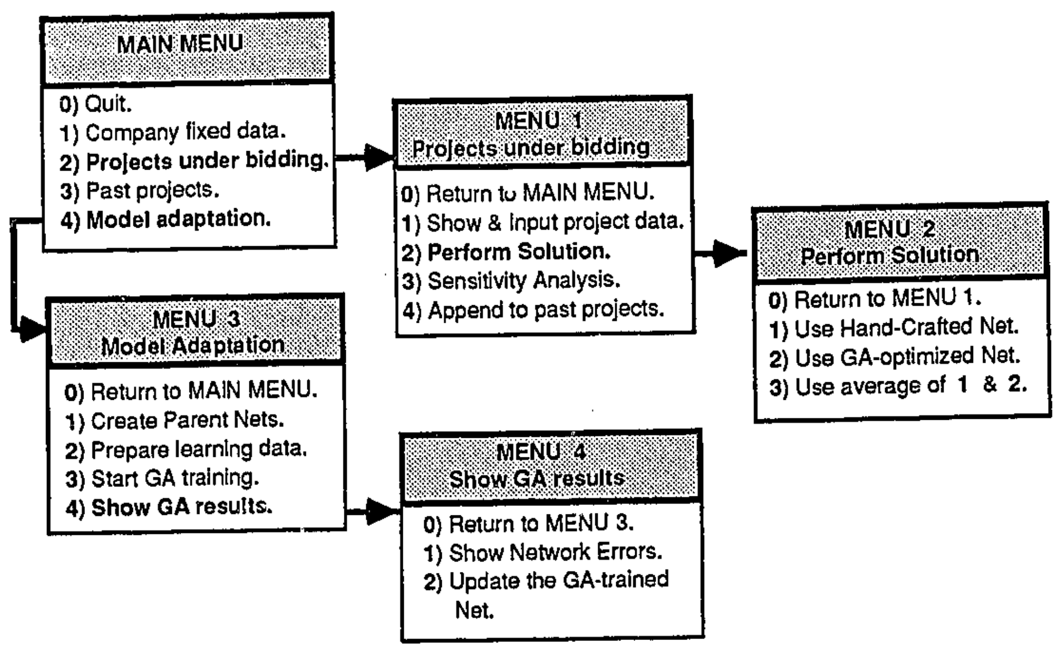


Fig. 6.2: Menu system of program "DBID".

covered by a given % markup value. These data are used as part of the input attributes for the neural networks of the system.

The second category of data used by the system pertains to a set of projects for which the contractor is currently preparing bids. A set of five screens (Figs 6.4 to 6.8) are designed to facilitate user input of these data, forming a database of these projects. The user is able to sail through the database of projects, allowing editing and printing of their data. For each project, Figs. 6.4 to 6.8 for example, the user enters some general information in addition to the contractor's assessment of the several factors that influence markup decisions and represent the system input attributes. Using these data, the neural network model can provide bidding decision assistance for the set of projects under bidding.

The third category of data pertains to past projects on which the contractor had bidding experience, either successful or unsuccessful. Similar to the database of projects under bidding, a database for the contractor's past projects is constructed. The data pertaining to each project is entered through a set of six screens. The first five screens are the same as those for the projects under bidding (Figs. 6.4 to 6.8) which provide a description of the project environment. The sixth screen (Fig. 6.9) is designed for user input of 7 data elements that relate to the contractor's markup decision made under the project environment described and to the actual outcomes of the project after being constructed. These 7

FILE: FIX.DAT

```
                                          ESC to exit
|FIXED DATA|
                    FIRM SIZE,
        MARKUP DEFINITION & COMPONENTS

- Size of firm:
    Small: N       Medium: Y       Large: N

- Markup definition:   % of dir. cost: N
        % of dir.cost + Proj.Overhead: N
   % of dir.cost + P.O. + Gen.Overhd: Y

- Markup components:      Profit only: N
                   Profit+contingency: Y
                         Profit + G.O.: N
               Profit + G.O. + Contin.: N
        Profit + P.O. + G.O. + Contin.: N
```

Fig. 6.3: Screen for the contractor's general data.

FILE: CURRENT.DAT   PROJECT #:   2

```
                                    First screen
GENERAL INFORMATION
- PROJECT:     MULTI-FAMILY HOUSING
- LOCATION:    MONTREAL, CANADA

- TYPE:        1) Building
               2) Heavy Civil
               3) Industrial      --> 1

- OWNER:       1) Public
               2) Private         --> 2

- CONTRACT:    1) Lump Sum
               2) Unit Price
               3) Other           --> 1

                               mm-dd-year
- EXECUTION START DATE:        05-01-1990
- CONTRACT DURATION      :     11     (months)
- TOTAL BID PRICE ($)    :      4.92 (million)
```

<ESC-quit> <F1-GOTO> <F2-BackWD> <F3-FWard> <F6-PRINT>

Fig. 6.4: Input screen 1 for a project under bidding.

```
FILE: CURRENT.DAT            PROJECT #:  2 (cont.)
|JOB UNCERTAINTY|                                  <---
  DUE TO:
                       L=1, H=5
 - Site conditions---------> 4
 - Owner attitude----------> 2
 - Project location--------> 3
 - Safety hazard-----------> 1
 - Inaccuracy in estimate-> 2
 - Sensitivity to weather-> 3
```

```
<ESC-quit> <F1-GOTO> <F2-BackWD> <F3-FWard> <F4-EDIT> <F5-NEW> <F6-PRINT>
```

Fig. 6.5: Input screen 2 for a project under bidding.

```
FILE: CURRENT.DAT            PROJECT #:  2 (cont.)
|JOB UNCERTAINTY|                  ---> |JOB COMPLEXITY|
  DUE TO:                                                 L=1, H=5
                       L=1, H=5          - Technology needed------> 2
 - Site conditions---------> 4           - Resources needed-------> 2
 - Owner attitude----------> 2           - Job size---------------> 3
 - Project location--------> 3           - Quality of design------> 4
 - Safety hazard-----------> 1           - Stacking of trades-----> 3
 - Inaccuracy in estimate-> 2            - % Subcontracted--------> 1
 - Sensitivity to weather-> 3            - Rigidity in Specs------> 2
```

```
<ESC-quit> <F1-GOTO> <F2-BackWD> <F3-FWard> <F4-EDIT> <F5-NEW> <F6-PRINT>
```

Fig. 6.6: Input screen 3 for a project under bidding.

```
FILE: CURRENT.DAT              PROJECT #:  2 (cont.)
|JOB UNCERTAINTY|                         |JOB COMPLEXITY|
    DUE TO:                                                     L=1, H=5
                         L=1, H=5            - Technology needed------> 2
 - Site conditions--------> 4                - Resources needed-------> 2
 - Owner attitude---------> 2                - Job size---------------> 3
 - Project location-------> 3                - Quality of design------> 4
 - Safety hazard----------> 1                - Stacking of trades-----> 3
 - Inaccuracy in estimate-> 2                - % Subcontracted--------> 1
 - Sensitivity to weather-> 3                - Rigidity in Specs------> 2

|[illegible]|                      <---
                         L=1, H=5
 - Inflation rate---------> 1
 - Escalation rate--------> 1
 - Competition------------> 1
       (No. OF COMPETITORS:   1)
 - Economy growth---------> 3
 - Resources availability-> 3

<ESC-quit> <F1-GOTO> <F2-BackWD> <F3-FWard> <F4-EDIT> <F5-NEW> <F6-PRINT>
```

Fig. 6.7: Input screen 4 for a project under bidding.

```
FILE: CURRENT.DAT              PROJECT #:  2 (cont.)
|JOB UNCERTAINTY|                         |JOB COMPLEXITY|
    DUE TO:                                                     L=1, H=5
                         L=1, H=5            - Technology needed------> 2
 - Site conditions--------> 4                - Resources needed-------> 2
 - Owner attitude---------> 2                - Job size---------------> 3
 - Project location-------> 3                - Quality of design------> 4
 - Safety hazard----------> 1                - Stacking of trades-----> 3
 - Inaccuracy in estimate-> 2                - % Subcontracted--------> 1
 - Sensitivity to weather-> 3                - Rigidity in Specs------> 2

|MARKET CONDITIONS|                   --> |[illegible]|
                         L=1, H=5                                       L=1, H=5
 - Inflation rate---------> 1                - Similar experience-----> 5
 - Escalation rate--------> 1                - Mgmt & supervision-----> 4
 - Competition------------> 1                - Confidence in wrk force> 3
       (No. OF COMPETITORS:   1)             - Financial capability---> 5
 - Economy growth---------> 3
 - Resources availability-> 3                -     NEED FOR WORK IS --> 4

<ESC-quit> <F1-GOTO> <F2-BackWD> <F3-FWard> <F4-EDIT> <F5-NEW> <F6-PRINT>
```

Fig. 6.8: Input screen 5 for a project under bidding.

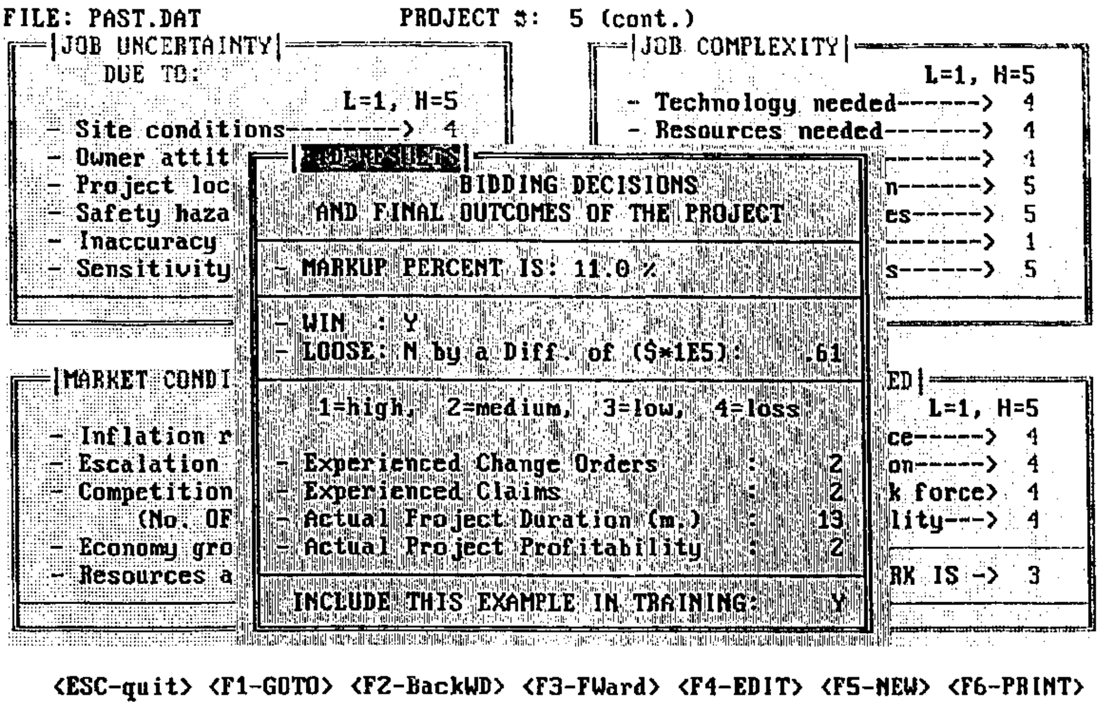


Fig. 6.9: Input screen 6 for a past project.

data elements are the same as the output attributes of neural networks, however, are input by the user. The database of past projects, as such, represent the contractor's bidding environment since it encodes his intuitive bidding decisions made in response to various project situations, and the impact of such decisions on the projects' outcomes. Past projects database is, therefore, used by the system to adapt the performance of the GA-optimized net to the user's environment.

6.3 Adapting the Markup Model

Based on the NN developments described in Chapter 5, the proposed D.S.S. retains two neural networks: 1) the hand-crafted network; and 2) the GA optimized network. The first is well trained on domain but not contractor-specific knowledge and is utilized to generate parent networks for the GA optimization. The second network, on the other hand, is trained on the domain knowledge using the GA technique to generalize well on project examples not used in training, and is utilized for problem solving in new bid situations. The second network, thus, represents an adaptation of the general model (network 1) to some additional specific knowledge. As such, to adapt the markup model to the experiences (i.e., the environment) of the user, his database of past projects could be used to re-train a neural network using the genetic algorithms technique to generalize on the data of such database. Once such network is constructed, it represents an adaptive network that could be used to assist in making decisions on future bids.

The model adaptation process is fully automated through **DBID** program. The contractor has the flexibility to select from the database of past projects a set of projects that best represents his bidding environment. For examples, the contractor may restrict his selection to projects executed within the past five years. Menu 3 of the program (Fig. 6.2) outlines the GA-based adaptation process. First, a population of parent networks are automatically generated by DBID (default is 5 networks) with their weights and biases as variations of those pertaining to the hand-crafted network (i.e., variation ranges between ± 0% to ± 20%). These variations, as opposed to using totally random numbers, provide moderate "shaking" of the network in order not to severely disrupt the knowledge of the network and to allow fast adaptation. Once parent networks are generated, the projects representing the contractor's environment are prepared for use by the GA procedure. The GA process then conducts an optimization search, utilizing the contractor's past projects as test examples to control the optimization process. The procedure is fully described in Chapter 5 where the objective function in the optimization is formulated so as to minimize the errors on the test examples. The GA optimization session may take a long time (about 15 minutes per offspring generation on a PC 486 processor, 33 MHz speed). The user, however, is able to stop the process and review the network errors depicted in each of the network attributes. Accordingly, the user may: continue the process; update the GA-optimized network; or cancel the adaptation process. Once the adaptation process is finished, program **DBID** can then be used in a new bid situation, with 3 possible options in producing the outputs: 1) utilize the hand-crafted network; 2) utilize the GA optimized network that is adapted to the contractor's environment; or 3) use an average of both networks.

It should be noted that the adaptation process described is used to incorporate additional knowledge (bid encounters) to the model and accordingly adapt the model performance. However, the number of input and output attributes and the NN configuration used (number of hidden layers and the number of P.E.s in each) are fixed as those found optimum in the NN development stage. The user has no access to these data and cannot adjust the model for other purposes. This limitation is mainly attributed to the highly problem-dependent nature of neural network models.

6.4 Sensitivity Analysis Using Monte Carlo Simulation

In estimating optimum markup for a new project, the contractor may often be uncertain about an accurate assessment of one or more of the project risks used as model inputs. This is because the time permitted for bidding is usually short which does not allow the contractor to collect as much information regarding all the risk-related factors as needed. In the **DBID** program, such uncertainty in the contractor's assessment is accounted for by a sensitivity analysis conducted using the Monte Carlo simulation technique (Moselhi et al. 1993). The objective is to study the sensitivity of the model outputs, particularly optimum markup, to changes in the assessment of the risk-related factors.

Monte Carlo simulation is a technique used for risk analysis and uncertainty modeling. The method has been reported as appropriate when the final outcomes to a decision problem depends of the outcomes of a number of different uncertain events and on the manner in which they might combine (Birnie and Yates 1991). For the problem in hand, the contractor is required to provide an assessment of several factors on a scale from 1 (for low) to 5 (for high); such as the contractor's need for work. Using the **DBID** program, the Monte Carlo process conducted to account for the uncertainty in the scores can be summarized as follows:

1 - The user selects a project from those under bidding and enters his assessment of most likely **scores** (1 to 5) for the different **factors** describing the project (e.g., need for work, owner attitude, etc).

2 - The user enters the number of simulations.

3 - A counter for the simulation number is incremented by 1.

4 - A random number, in the range form 0 to 1.0, is generated for each project factor in each simulation.

5 - A project factor is selected by its attribute number. The factor's score is adjusted in three possible ways: a) if the random number is less than 0.25, the score is decreased by 1; b) if the random number is equal to or larger than 0.25 and less than 0.75, the score is left as is; c) and if the random number is equal to or higher than 0.75, the score is increased by 1. In this

manner, scores are taken as discrete and the initial score has a larger probability of being left as is, thus limiting the amount of distortion to the initial project scores. Also, the discrete values allowed for the score are limited to: the initial score; a one increment up; and a one increment down.

6 - Steps 4 and 5 are repeated for all the project factors that require contractor assessment (considered as independent). This step completes one simulation. The simulation data are stored for further processing.

7 - Steps 3, 4, 5, and 6 are repeated for the number of simulations specified.

8 - The data of the simulations are processed using the neural network model, producing for each simulation a set of conclusions regarding the 7 output attributes of the model.

9 - The mean and standard deviation of each output attribute (e.g., optimum markup) in all simulations are then calculated. The mean represents the expected value of the attribute considering changes in the assessment. The standard deviation also represents the sensitivity of such decision to changes in the assessment.

6.5 Assessment of the Probability of Winning

In addition to providing an estimate of the likely decision and its variance, the benefit of applying the Monte Carlo simulation technique is that it can provide an assessment of the probability of winning at a given markup and conversely the markup value that corresponds to a desired probability of winning. From the simulation results, a markup histogram can be constructed and its discrete values accumulated to estimate the probability of winning at a given markup. The markups produced could also be approximated as having a normal distribution (the method adopted in program **DBID**). Therefore, the expected value and the standard deviation may be utilized to predict the probability of winning at any given markup.

6.6 Example Application

In order to demonstrate the operation of program **DBID** as a decision support system, an example application is presented. The example represents a project on which the contractor is preparing a bid. First, the contractor has to input the organization fixed data (main menu, option 1, see Fig. 6.2). Data pertaining to the project in hand are then input and added to the database of projects under bidding. The user selects menu 1, option 1 (projects under bidding, show and input data). The data are entered through a set of five screens pertaining to the general information and the contractor assessment of the various factors describing the project environment, Figs. 6.4 to 6.8. Once the project is added to the database as project #2, the contractor may select menu 1, option 2 to select the neural network to be used in producing the outputs (menu 2, Fig. 6.2). Using the GA optimized network for the present example, the neural network model is then activated and accordingly the user is prompted with a set of 3 screens (Figs. 6.10 to 6.12) showing the input data and the corresponding conclusions produced by the NN model. The conclusions reached (Fig. 6.12) can be read as follows:

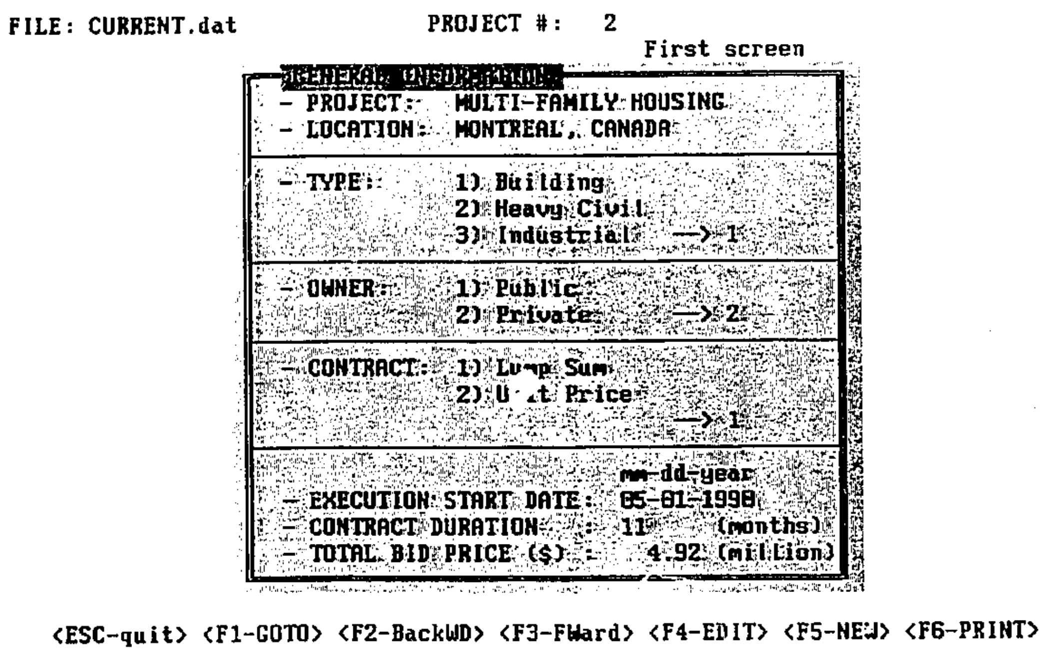


Fig. 6.10: Output screen 1 for the NN results.

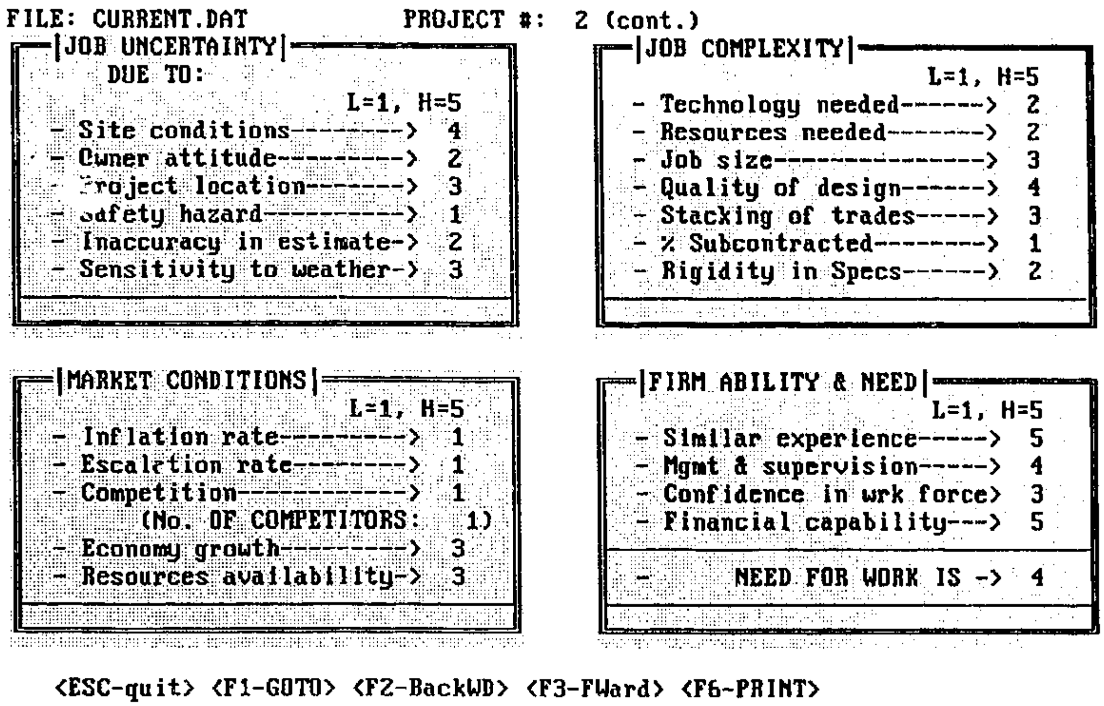


Fig. 6.11: Output screen 2 for the NN results.

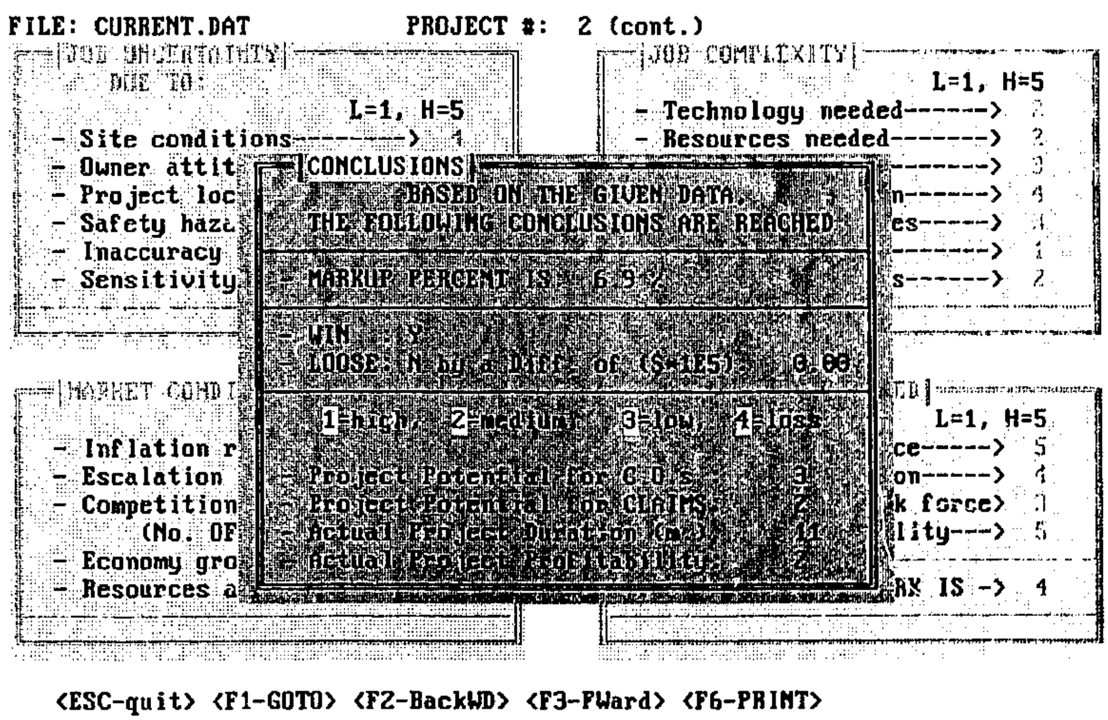


Fig. 6.12: Output screen 3 for the NN results.

"out of the knowledge currently incorporated in the system pertaining to some past projects, the data provided for the present project leads to the following conclusions:

- At 6.9% markup, the contractor is likely to win the bid with almost no difference between his bid and the second lowest bidder.
- The project has low potential for change orders; medium potential for claims; no expected delays in duration; and may end up as a medium profitability."

In order to investigate the effect of uncertainty or variations in the initial assessment to the risk-related factors of the project, sensitivity analysis is conducted (menu 1, option 3 of Fig. 6.2). The number of simulations in the present example is set to 100. The Monte Carlo simulation process is then conducted and the results are presented in Fig. 6.13, showing the mean and standard deviation values for each output attribute. For the present example, the mean value of the optimum markup, produced by the simulation process, is 7.09% (as compare to 6.9% before conducting the simulation (Fig. 6.12). Although the standard deviation is slightly high (1.9%), the mean value of the markup is close to the original estimate. This might give the contractor some confidence about the markup estimate produced.

In order to assess the probability of winning and its variation with percent markup, Simulation Results

Total simulations = 100

		Mean(μ)	ST.Dev.(δ)
-Percent Markup (%)	:	7.09	1.90
-Win/Lose (1=win; 0=lose)	:	0.99	0.10
-Difference ($ * 100,000)	:	0.33	1.59
-Potential C.O.s (1=H to 3=L)	:	2.68	0.47
-Potential Claims(1=H to 3=L)	:	1.19	0.42
-Actual Duration (months)	:	10.62	1.55
-Actual Profit(1=H to 4=Loss)	:	2.00	0.00

..Hit <RETURN> to see PLOT 1..

Fig. 6.13: Simulation results.

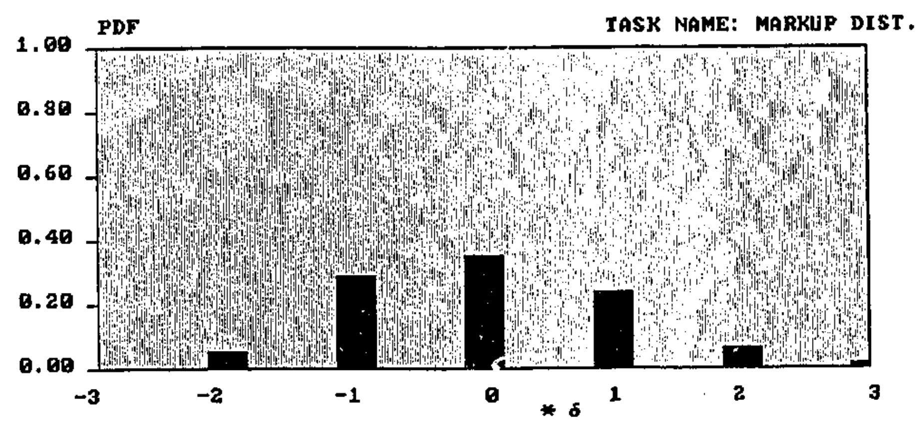
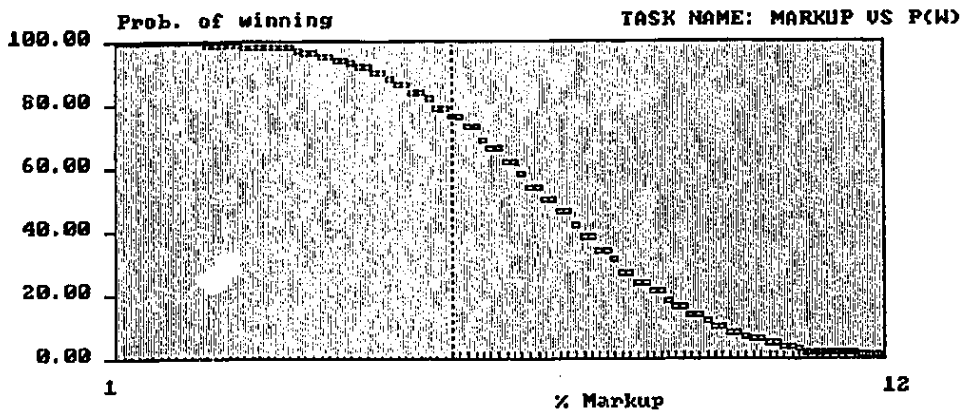


Fig. 6.14: Historgam of the % markup in all simulations.



- % Markup : 5.84
- P(w) : 76.00

<ESC-QUIT> <-->-RIGHT> <<---LEFT> <END, HOME>

Fig. 6.15: Cummulative density function for % markup.

the markup histogram is constructed from the simulation results and presented to the user (Fig. 6.14). It can be seen from the shape of the histogram that it can be reasonably approximated by a normal distribution. Accordingly, the markup expected value and the standard deviation are used to calculate the cumulative density function (see Fig. 6.15). The cumulative density function (CDF) is then presented on the screen and the user is allowed to move a vertical cursor (see Fig. 6.15) in small intervals and its corresponding coordinates on the CDF function (% markup; probability of winning) are instantaneously calculated and presented on the screen.

6.7 Conclusion

A decision support system (**DBID**) for construction bidding is developed using neural networks for markup estimation. The D.S.S. retains two neural networks. The first network is well trained on domain but not contractor-specific knowledge and is utilized for GA optimization and model adaptation. The second network, on the other hand, is trained on the domain knowledge and can be adapted to the user environment, and is utilized for problem solving in new bid situations. The two networks are complementary and there are several benefits for having both networks as model components. First, the contractor is not dependent on his experiences only which might not reflect a proper bidding process. Second, the model is adaptable to users' environments; novice contractors can utilize the provided model while prominent contractors can adapt the model based on their experiences. Third, comparing the results produced by the two networks may give the contractor some confidence in the conclusions reached.

CHAPTER 7

BID PREPARATION PROTOTYPE - A HYPOTHETICAL EXAMPLE

7.1 Introduction

This Chapter describes the development of a prototype for the integrated bid preparation system proposed in this study. The capabilities and limitations of the prototype are described with respect to the system's four main modules: 1) WBS and cost estimation; 2) planning and scheduling; 3) risk assessment and markup estimation; and 4) bid unbalancing and cash flow optimization. The performance of the prototype is demonstrated through an example hypothetical project.

7.2 Prototype Description and Limitations

Based on the integrated bid preparation methodology proposed in this study and described in Chapter 3 (see Figs. 3.11 and 3.12), a prototype microcomputer system (**ESTIMATOR**) is developed. The computerized system is designed in a modular architecture, incorporating four main modules, as shown earlier in Fig. 3.10: a WBS and cost estimation module, a planning and scheduling module, a risk assessment and markup estimation module, and a bid unbalancing and cash flow optimization module. These modules are linked through a core of databases structured based on input from Montreal area contractors in building and civil engineering works (Fig. 3.7).

The system contains basic resource data in six databases: labour database, equipment database, construction material database, permanent material database, crews database, and subs database. Another database includes data regarding various work tasks, reference to the resources needed for the tasks, and the respective production rates of those resources. One last database contains data regarding the competitors and their performance in past bids. The record structure of the system databases is shown in Fig. 3.7 of Chapter 3. The present system databases are implemented in Lotus 1-2-3 for its database, spreadsheet, and graphics capabilities. Each of the system's eight databases is a separate lotus file, and the database manager is a macro-driven program that customizes the user interface, allowing user-interactive maintenance of the databases and simultaneous data viewing as needed. The decentralized design adopted allows large databases to be maintained without much effect on the processing time. When a database is saved, a standard ASCII version of the database is produced and can be read by an interfacing program designed to link the databases with the different system modules.

7.2.1 Cost estimation module

The cost estimation module incorporates three programs: one is for performing the WBS and the other two are for estimating the project direct and indirect cost, respectively. The WBS program is implemented in a user friendly lotus environment. Currently, the WBS allows the user to specify only three levels: area level (1 to 7 elements), major-activity level (1 to 7 elements), sub-activity or task level (1 to 7 elements). The total number of elements permitted at the lowest WBS level is therefore 7 * 7 * 7 = 343. This restriction, however, is not expected to affect the practicality of the system since WBS elements (currently at the area level only) could be assigned number of repetitions. In a high-rise project, for example, a project area may be called "Floor Construction" and made to incorporate the activities in a typical floor. This project area could then be assigned a number of repetitions equal to the total number of floors. Accordingly, all activities and tasks of that area are assigned the same number of repetitions.

Each element in the WBS is assigned a unique code of accounts. The code is designed to be a sequential number from 1 to 400: 7 elements at the area level, 49 elements at the second level, and 343 elements at the third level, in addition to one code for the entire project. As such, the code allows the data pertaining to the WBS elements to be included in a database of 400 records. At any WBS level, the elements' codes are function of their parent elements at upper levels, and as such the aggregation and/or detailing of the WBS data is facilitated. The WBS is performed in an interactive manner, with input templates and menus that allow direct access to the tasks database and the contract items database. To facilitate simple development and review of an estimate, the WBS menus dynamically incorporate the selected WBS elements into the user interface, hiding the internally calculated code of accounts.

An algorithm for direct cost estimation is developed using QuickBasic 4.5. To facilitate direct cost calculations, the algorithm is tied to the WBS program and to the system databases (Moselhi et al. 1991d). In estimating the direct cost, the algorithm reads the project WBS database and the tasks associated with its elements. Tasks' data, including the codes for the resources used, are then retrieved from the tasks database. Resources codes are, in turn, used to retrieve the appropriate resource database and accordingly resources' cost as well as the task duration are calculated for each element at the lowest WBS level. The direct cost is then aggregated upwards to the upper WBS levels, considering the number of repetitions specified. In performing direct cost calculations, the user is allowed to specify an overtime strategy for the labour and select among 3 options for the hourly rate of an equipment: 1)operational rate; 2) operational + internal rental rate; or 3) external rental rate. These options account for the type of equipment resource assigned to a task (i.e., owned versus externally acquired) and, accordingly, accurate direct cost can be calculated.

The third procedure incorporated in the cost estimation module is a lotus spreadsheet designed with different indirect cost items. The contractor is expected to estimate the costs pertaining to the items applicable to the project in hand. Indirect costs include the project overhead costs and a part of the firm general overhead. The optimal distribution of the total indirects, in addition to the direct cost and the markup, among the contract items will be dealt in section 7.2.4.

7.2.2 Planning and Scheduling module

In order to facilitate planning and scheduling, it is required that the WBS be logic-driven. The default logical relationships are set as parallel project areas (level 1 of the WBS) with sequential elements for levels 2 and 3 (activities and tasks). Such default relationships, however, can be changed by the user. The Planning and scheduling module incorporates the following:

- An algorithm for calendar date/working day conversion that allows the user to specify the project start date, non-working days in the week, and other non-working dates.
- An algorithm for specifying logical relationships among the WBS elements. Relations could be of any type: Finish-to-Start, Finish-to-Finish, Start-to-Start, or Start-to-Finish. Lag times in days could be specified. Relationships between a repetitive and a non-repetitive activity are permitted, however, currently not to a particular unit in the repetitive activity. For flexibility, logical relationships could be specified at any WBS level, for example, two project areas could be assigned predecessor-successor relationship, as seen later in the example application.
- An integrated algorithm for scheduling projects with repetitive and non-repetitive activities (Hegazy et al. 1993a). The algorithm is basically a PDM procedure generalized to accommodate repetitive activities, following the scheduling procedure described by Russell and Caselton (1984) for

sequential repetitive activities. The algorithm considers the weather impact on activity duration through user specified, monthly productivity factors. To account for resource constraints, overtime could be specified for user-selected activities and the impact of such strategy on the project cost and time is estimated. In such algorithm, first, sequential step numbers are assigned to the WBS tasks based on their successor and predecessor relations. Forward path calculations are then performed by ordering the tasks according to their sequence step numbers. When a task is selected, its early start time is calculated as the largest of its predecessors' finish times. If the task is repetitive, early start times for all the units are calculated considering work continuity along the repetitive units and the logical relationships with predecessors. Once start time is calculated, the task's duration is then modified by an average of the productivity factors of the months along the activity duration. Early finish time for the task can then be calculated based on its early start time and the modified duration. The procedure is then continued, selecting another task which has equal or higher sequence step number, leading to the determination of start and finish times as well as the modified durations for all activities.

7.2.3 Risk assessment module

The risk assessment module incorporates the neural network decision support system for bidding (**DBID**) discussed in Chapter 6. This module is activated through the interfacing algorithm of the system.

7.2.4 Bid unbalancing module

The bid unbalancing module, as described in Chapter 3, performs optimum bid unbalancing and cash flow optimization. A Quickbasic algorithm is prepared using the Simplex method for optimization. The algorithm reads the cost and schedule data for all the WBS elements and establishes the quantity and cost pertaining to each contract article in every month along the project duration. The optimization constraints are then formulated in a user interactive manner. As mentioned in Chapter 3, section 3.4.5, default values for the upper and lower limit constraints of the "all-in" unit prices (items costs/items quantities) are set to 1.2 and 1.0 of only the **direct** unit prices, respectively. These default values are made to safeguard against unbalancing risks. A lower limit of 1.0 * (item direct cost/item quantity), for example, ensures that the contract item will be priced below its direct cost. The default value for the contractor's monthly cash-out-of-flow limit is also set to the first month's direct cost. These default values can also be changed by the user.

Once the optimization constraints are set, the optimization can be performed and optimum unit prices could be assigned to the contract items. It should be noted that the performance of the algorithm used is limited by the memory limitations of the hardware used and the problem size. For large projects with hundreds of contract items, memory limitations could be exceeded and the user is prompted.

To assist the user in such situations, the system is designed to incorporate a manual bid unbalancing option where the user can specify the portion of the indirect costs and profit assigned to each contract item, and accordingly, the unit prices are calculated. Such option permits contractors to confirm and possibly change the determined unit prices as well as perform bid unbalancing as they would naturally do.

7.3 Example Application - A Hypothetical Project

In order to demonstrate the capabilities of **ESTIMATOR** and its performance, a hypothetical project example is selected and its unit-prices prepared using the system. The process of preparing a bid follows the methodology of Chapter 3 (Fig. 3.11). The menu system of **ESTIMATOR** is outlined in Fig. 7.1, showing the prototype's available options. After selecting the example project, the options given under PROJECT MENU (Fig. 7.2) have to be followed in sequence and the user is prompted if not. The first option is to specify the project calendar: project start date, non-working days, and non-working dates (Fig. 7.3). The second option is filling the computerized contract articles form, and performing the WBS. Based on the bid documents, a database for the contract items is filled, having the following fields: contract item number, item description, unit of measure, and the quantity (Fig. 7.4). Following that, work breakdown structure (WBS) is performed, based on study of the project drawings and specifications. Each element in the WBS is automatically assigned a unique code of accounts (Fig. 7.5). The elements in the

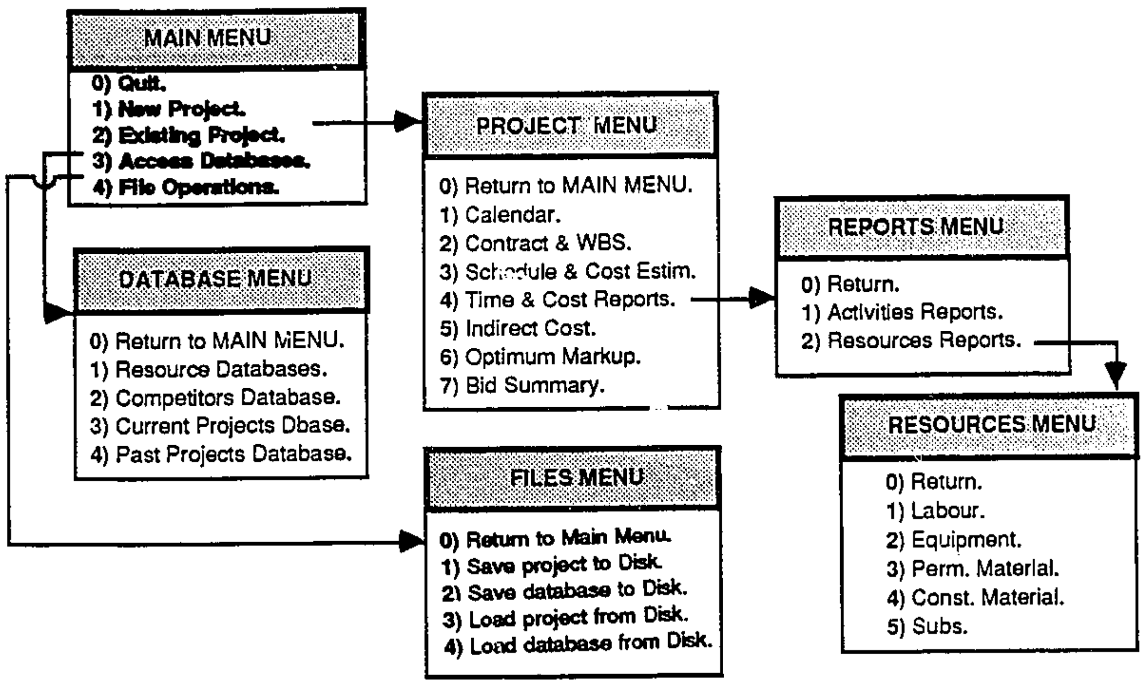


Fig. 7.1 : ESTIMATOR Menu system.

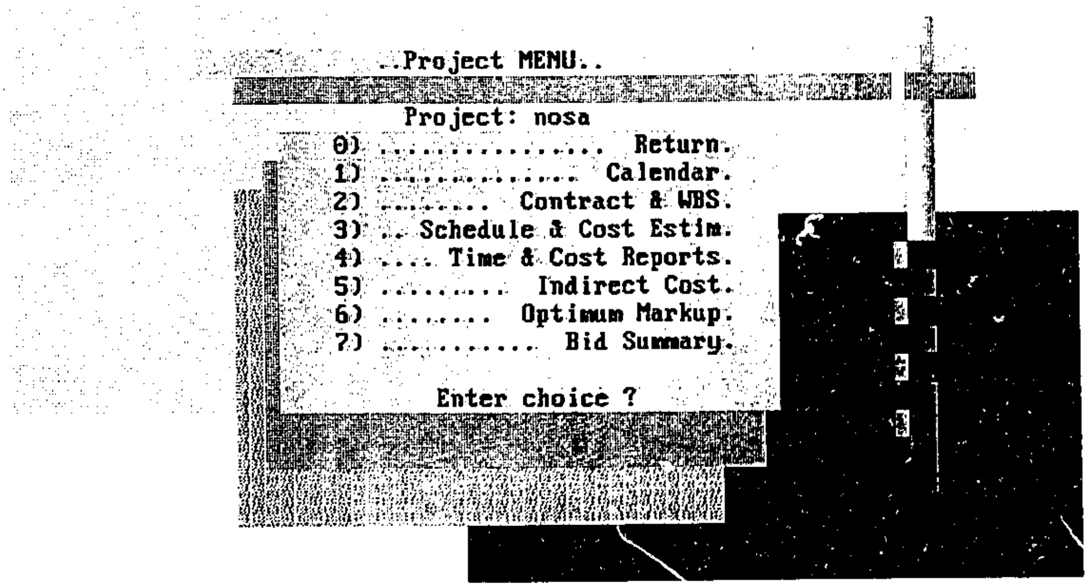


Fig. 7.2: Screen for PROJECT MENU.

```
CALENDAR

NAME:          NOSA
DESCRIPTION:   A TEST AND DEMONSTRATION PROJECT

START DATE:    05-04-1991  (MM-DD-YYYY)

WORKING DAYS:                  SATURDAY:N    SUNDAY   :N
               MONDAY   :Y     TUESDAY  :Y   WEDNESDAY:Y
               THURSDAY:Y      FRIDAY   :Y

NON-WORKING DATES: 1:12-25-1991   2:01-07-1992   3: - -
      4: - -       5: - -         6: - -         7: - -
      8: - -       9: - -        10: - -        11: - -
     12: - -      13: - -        14: - -        15: - -
```

Fig. 7.3: Calendar screen.

```
MODIFY_Articles  SAVE_Contract  PRINT_Contract  Worksheet  QUIT
Modify existing contract articles
```

CONTRACT ARTICLES DATA BASE

ART. #	ART. DESCRIPTION	ART.QUANTITY	ART.UNIT
1	Excavation	1000	Cuyd
2.1	Piled foundations	20	units
2.2	Isolated Footings	25	units
3	Superstructure	20	units
4	Brick work	2000	sqft
5	Electrical work	1	lsum
6	Mechanical work	1	lsum
7	Interior Decoration	20	units
8	landscaping	1	lsum

Fig. 7.4: Contract articles spreadsheet.

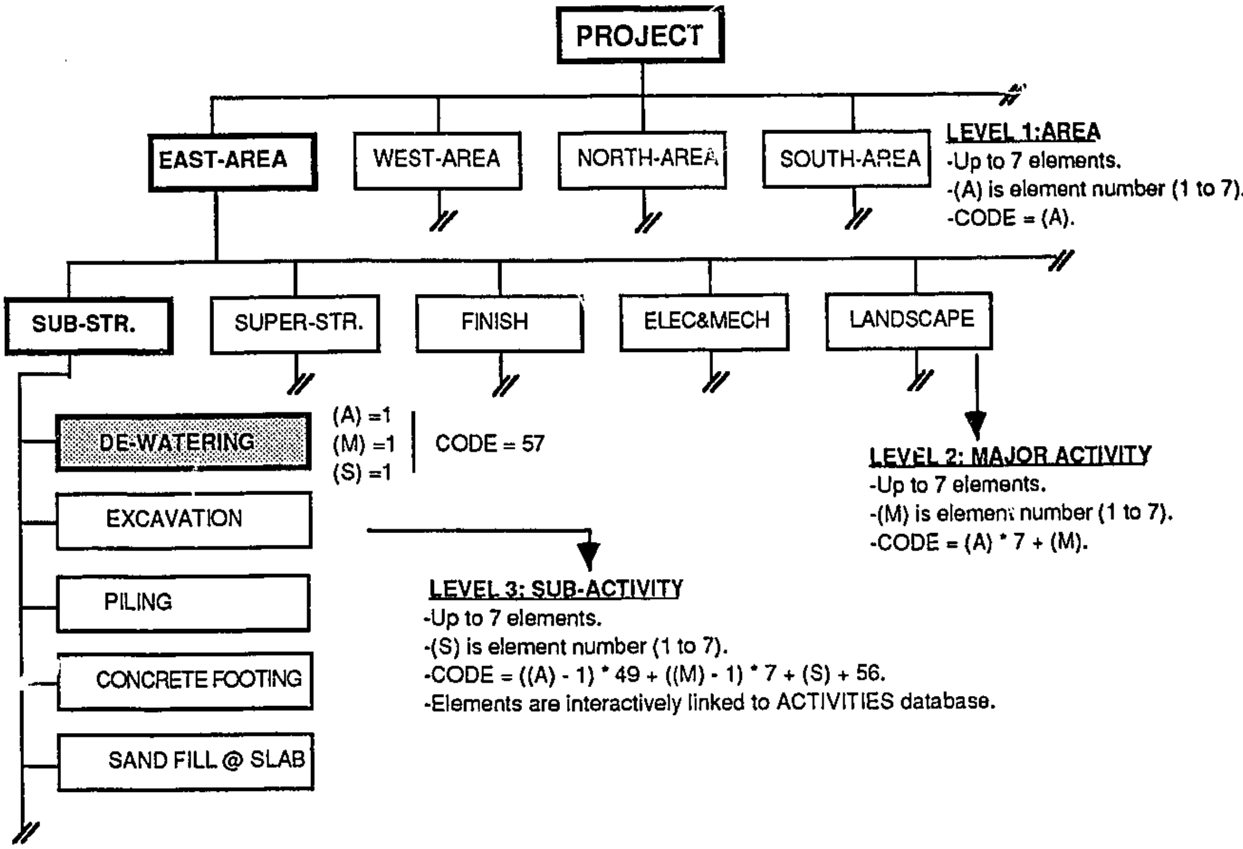


Fig. 7.5: Work Breakdown Structure (WBS) elements and the code of accounts used - Example Project.

lowest level of the WBS are interactively assigned a task from the tasks database and the user is prompted for the quantity of work estimated and the appropriate contract item to which this activity is linked (Fig. 7.6a). The WBS elements' data are stored in a database of 400 records, with the following fields: a reference indicating the position of this element on the WBS, the contract article that is linked to the element, the quantity of work, the unit of measure, the number of work repetitions, and the task code that links that element to the tasks database (Fig. 7.6b).

Once the logic-driven WBS is performed (i.e., parallel areas with sequential tasks), the third option in the project menu can be used to calculate the direct costs and the scheduled times associated with all the elements of the WBS. The interface algorithm activates the cost estimation module to calculate initial costs. The algorithm reads the tasks in the WBS elements and retrieve their resource type, cost, and average productivity data from the appropriate databases. The user is then prompted to input data regarding the overtime strategy and the method of calculating the equipment cost (Fig. 7.7). The algorithm calculates the labour costs, equipment hours and costs, material quantities and costs, subs costs, and total manhours used. These data are used to calculate initial direct costs and initial durations for the lowest WBS elements.

Once initial costs are calculated, the interface program switches control to the

```
                    MODIFY_WBS  SAVE_WBS  PRINT_WBS  Worksheet  Rename_Project  QUIT
               EAST-AREA  WEST-AREA  NORTH-ARE  SOUTH-ARE
          SUBSTRUC.  SUPER-STR  FININSH  ELEC.&MEC  LANDSCAPE
     DEWATERIN  Excavatio  Piling  Concrete  Sand Fill
Change_Sub.Act.  Modify_Quantity_Ref.Art.  Return

       BE       BF       BG       BH       BI       BJ       BK
483          ++==========================================++
484          ::                                          ::
485          ::       W_B_S  Level 3 : SUB_ACTIVITIES    ::
486          ::                                          ::
487          ::      LEVEL 1: EAST-AREA                  ::
488          ::      LEVEL 2: SUBSTRUC.                  ::
489          ::               Sub. Act.  ART. #   QUANTITY ::
490          ::               _________  ______   ________ ::
491          ::      1  :     DEWATERIN      1       1000 ::   to change
492          ::      2  :     Excavatio      1       4800 ::   Art.# or
493          ::      3  :     Piling       2.1         50 ::   Quantity,
494          ::      4  :     Concrete     2.2        510 ::
495          ::      5  :     Sand Fill    2.2        160 ::   use arrows
496          ::      6  :                                 ::   to move &
497          ::      7  :                                 ::   Press RET.
498          ::         move using DOWN arrow,            ::   when done.
499          ::  MUST type <*> to select from the d.base  ::
500          ::    LEAVE NO GAPS _ MUST FILL IN SEQUENCE  ::
501          ++==========================================++
```

7.6 (a)

PROJECT :nosa.WBS W_B_S

REF.	ACT_DESCRIPTION	ART. #	QUANTITY	U.O.M.	Repeated	ACT_CODE
A1-M1-S1	DEWATERING	1	1000	sfca		2.22
A1-M1-S2	Excavation	1	4800	cuyd		2.1
A1-M1-S3	Piling	2.1	50	each		2.355
A1-M1-S4	Concrete Footing	2.2	510	cuyd		3.1
A1-M1-S5	Sand Fill @ Slab	2.2	160	cuyd		2.223
A1-M1-S6	0	0	0	0		0
A1-M1-S7	0	0	0	0		0
A1-M2-S1	Form Work	3	1200	sft		3.751
A1-M2-S2	Rebars	3	1200	sqft		3.5
A1-M2-S3	Concrete Walls	3	150	cuyd		3.12
A1-M2-S4	Structrl Framing	3	1200	sqft		6.1

7.6(b)

Fig. 7.6: Spreadsheet for WBS; a) WBS template and menues; b) WBS database.

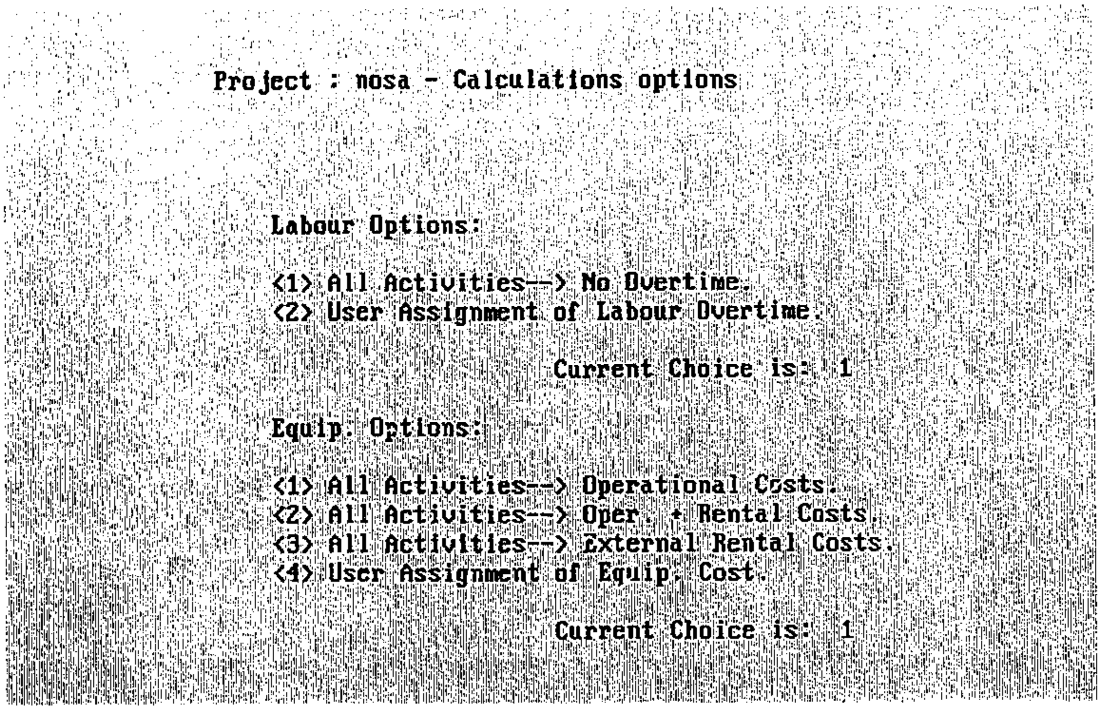


Fig. 7.7: Options for overtime strategy and equipment cost.

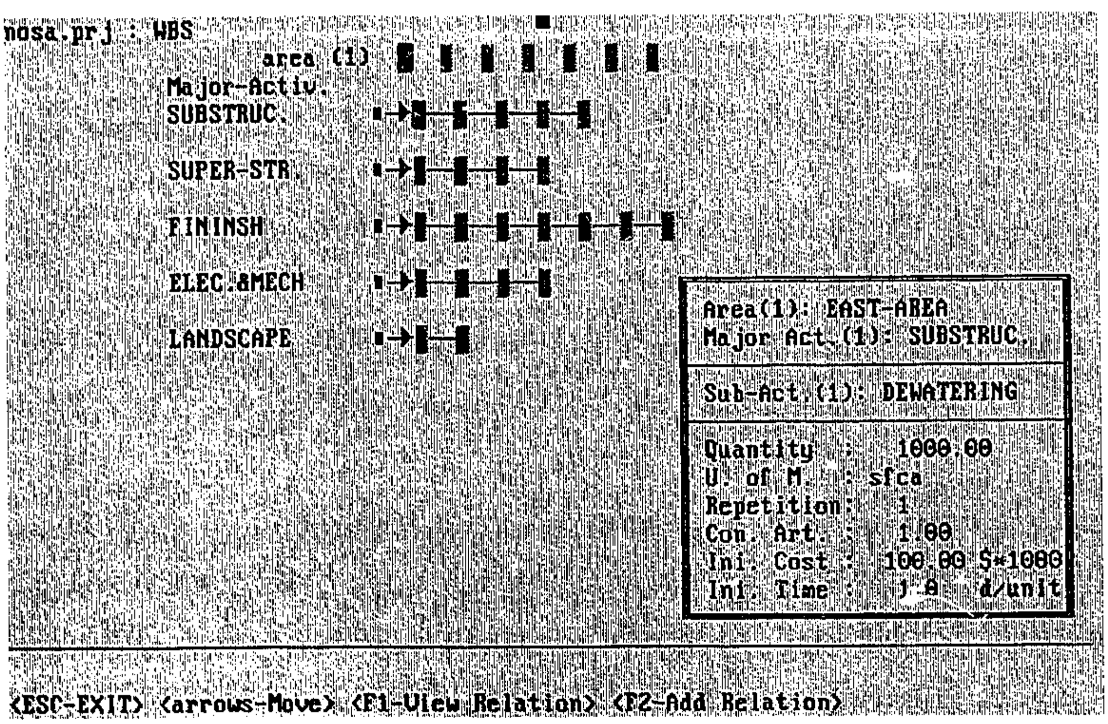


Fig. 7.8: Screen for WBS and logical relationships.

planning and scheduling algorithm that calculates the scheduled times for the tasks and refines the durations based on the monthly productivity factors (generated considering only the impact of weather on productivity). The algorithm starts by viewing the default logical relations among the WBS elements, allowing the user to sail through the project activities, interactively viewing the position of each element on the WBS and its initially calculated data (Fig. 7.8). The user can also view, add, or delete a relation, and accordingly the logical relationships are modified. Fig. (7.9) demonstrates an added relation in which area 4 is set as a predecessor to area 2 (finish-to-finish relation with a lag time of 2 days), as opposed to being parallel as the default. Once the logical relationships are decided, the algorithm proceeds with the schedule calculations based on user interactive input of the monthly productivity factors (Fig. 7.10). Accordingly, schedule calculations are performed as described earlier. Afterwards, the interfacing algorithm returns control to the cost estimation module to refine the project cost based on the tasks' modified durations. Accordingly, labour cost, equipment cost, and total manhours are adjusted. These costs are then aggregated to the different WBS levels and to the corresponding contract items.

After cost and schedule calculations, option 4 in the project menu allows access to various reports pertaining to the project activities, direct costs, schedule, and resource use. Samples activity reports for the example project are shown in Figs. 7.11 - 7.13 for a task report, area report, and a total project report, respectively.

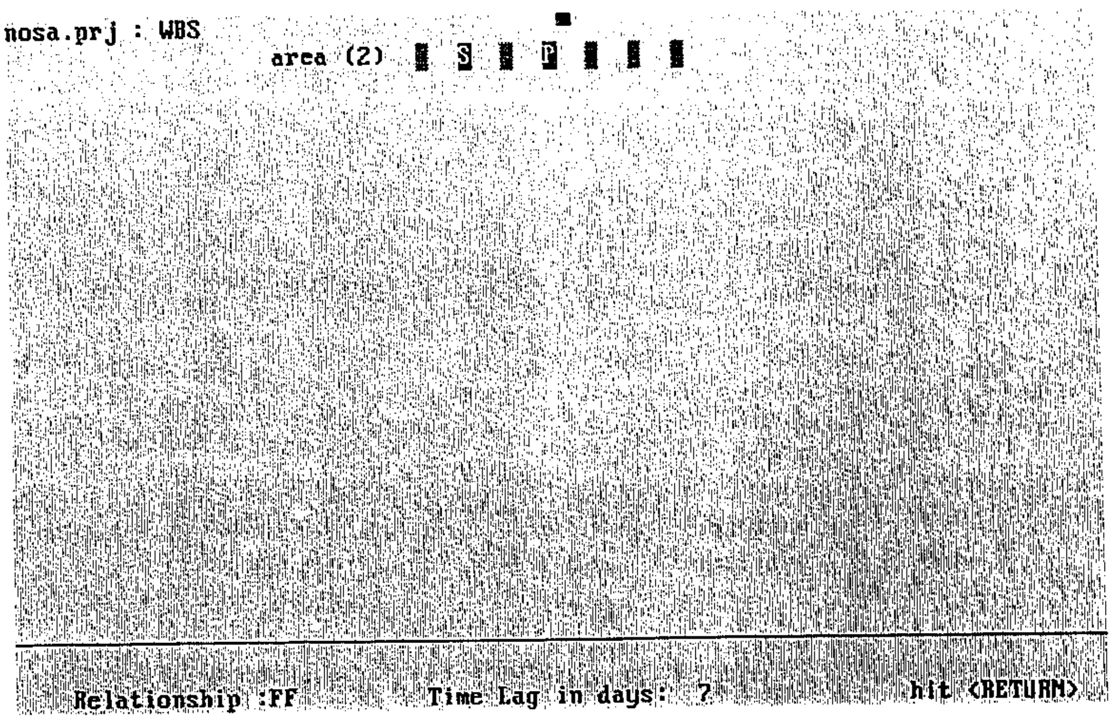


Fig. 7.9: Adding a relationship between areas 2 and 4.

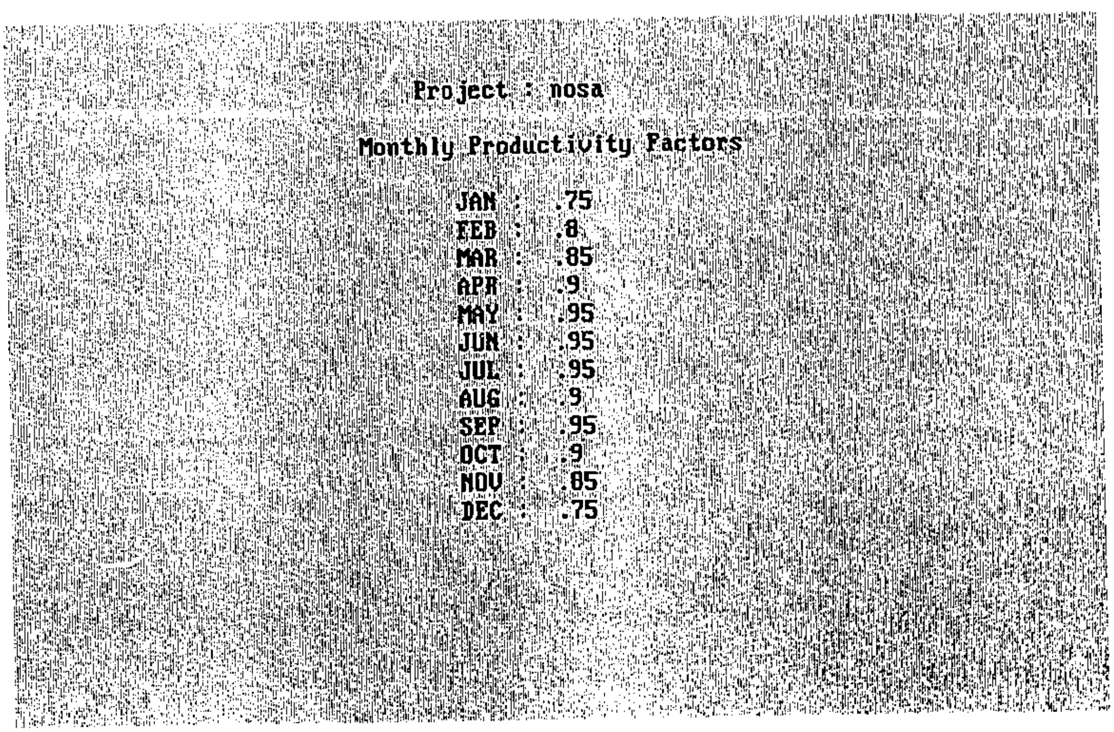


Fig. 7.10: Screen for monthly productivity factors.

```
Project: nosa.prj - Area: EAST-AREA  Repetitions:  1
--------------------------------------------------------------------------------
Major Activity: SUBSTRUC.
                  Sub_activity: Excavation   - Quantity:  4800
U.O.M.   Durat.   M.H.  Labour   Equip.      Mat.      Subs   Other      Total
         (days)  (hrs)  cost($)  cost($)   cost($)  cost($) cost($)   cost($)
--------------------------------------------------------------------------------
cuyd        2.0    109    3556      352         0        0       0     3908

Resources:            Start date:  5 7 1991    Finish data:  5 9 1991
   3 L2      ( 1 CR1    ) Hrs/week:45  O.T.: 0 ==> M.Hrs:    55 Cost:    1814
   1 E10     ( 1 CR1    ) Hrs/week:45  O.T.: 0 ==> E.Hrs:    18 Cost:       0(1)
   2 L4      ( 1 CR1    ) Hrs/week:45  O.T.: 0 ==> M.Hrs:    36 Cost:    1016
   1 E5                   Hrs/week:45  O.T.: 0 ==> E.Hrs:    18 Cost:     352(1)
   1 L8                   Hrs/week:45  O.T.: 0 ==> M.Hrs:    18 Cost:     726
--------------------------------------------------------------------------------
                     <1> NEXT        or       <E> EXIT
```

Fig. 7.11: Activity report.

```
               Project: nosa.prj - Area: EAST-AREA Report
     Start date:  5 6 1991    Finish data:  7 14 1992         Duration:  310.2762
--------------------------------------------------------------------------------
              #    M.H.   Labour   Equip.      Mat.     Subs   Other     Total
M.Activ.    Rep.  (hrs)  cost($)  cost($)   cost($)  cost($) cost($)  cost($)
--------------------------------------------------------------------------------
SUBSTRUC.    1    595.3    20777     3514     27304   122500       0   174095

SUPER-STR    1   6176.8   185926      720    180000        0       0   366645

FININSH      1   6538.8   206674        0     88710    10000       0   305384

ELEC.&MEC    1   1601.5    61770        0      4000   100000       0   165770

LANDSCAPE    1      5.7      185       18         0    50000       0    50204
--------------------------------------------------------------------------------
                  14918   475332     4252    300014   282500       0  1062098
--------------------------------------------------------------------------------
                     <1> NEXT        or       <E> EXIT
```

Fig. 7.12: Area report.

As shown in Fig. 7.13, the project total direct cost considering the impact of the schedule is reported as $2,197,489 while the total without considering the schedule impact is only $2,054,516 (an increase of 7%). Project duration is noticed to be 311 days and the start and finish dates of the project areas reflect the logical relationships specified. Sample resource report is also provided in Fig. 7.14. These reports provide elaborate data on the cost elements, manhours, and the resources used, for each task. Based on such reports, a cycle of schedule updating, changing the logical relations, and specifying an overtime strategy may become necessary in order to meet the project deadline and/or more efficiently re-arrange the resources used. Tasks' reports can, in this case, provide initial guidance concerning the tasks' cost-per-day, and thus, facilitate the selection of which tasks to crash, with minimum cost implications. The impact of such updates is directly reflected on the direct cost and schedule in a continuous cycle until the project constraints are met. Activity and resource reports could also be utilized in formulating a strategy for allocating individual indirect cost components. A spreadsheet containing a list of possible indirect cost items can be accessed through option 5 in the project menu of ESTIMATOR (Figs. 7.15 and 7.16). The user is expected to fill the costs pertaining to the project applicable categories.

After indirect cost estimation, the next option (option 6 in the project menu) could then be activated and the interfacing program prompts program (**DBID**) to provide a decision aid during optimum markup estimation. This is done based on the

Project: nosa.prj Summary Report

Start date: 5 6 1991 Finish data: 7 14 1992 Duration: 310.2762

AREA	# Rep.	M.H. (hrs)	Labour cost($)	Equip. cost($)	Mat. cost($)	Subs cost($)	Other cost($)	Total cost($)
		From:	5 6 1991	To:	7 14 1992		Duration:	310.3
EAST-AREA	1	14918	475332	4252	300014	282500	0	1062098
		From:	5 6 1991	To:	5 5 1992		Duration:	260.5
WEST-AREA	2	352	12290	2087	16139	61250	0	91766
		From:	5 6 1991	To:	7 8 1991		Duration:	46.0
NORTH-AREA	1	2260	72225	72	46235	101000	0	219532
		From:	5 6 1991	To:	4 24 1992		Duration:	253.4
SOUTH-AREA	1	10323	338040	27	115260	279000	0	732327
		28206	910176	8526	493787	785000	0	2157489

Without Considering Schedule Effect: 2054516

... Hit RETURN to CONTINUE ...

Fig. 7.13: Project report.

Project : nosa - Equipment Resource Report

Code	Desc.	#	Area	Major	Sub_act.	Cost_type	Hrs/Week	Overtime	Repetition	EQ.Hrs.	Cost/rep.	
E3	CONCRETE MIX	1	EAST-AREA	SUBSTRUC.	Concrete Foo	Operational	45	0	1	81	1557	(2 CR4)
E3	CONCRETE MIX	1	EAST-AREA	SUPER-STR	Concrete Wal	Operational	45	0	1	19	366	(2 CR4)
E3	CONCRETE MIX	1	WEST-AREA	SUB-STRUC	Concrete Foo	Operational	45	0	2	47	916	(2 CR4)
E3	CONCRETE MIX	1	NORTH-AREA	SUPER-STR	Concrete Wal	Operational	45	0	1	2	37	(2 CR4)
								Total This code:		196	3791	
E4	LOADER	1	EAST-AREA	SUBSTRUC.	Sand Fill @	Operational	45	0	1	2	20	
E4	LOADER	1	WEST-AREA	SUB-STRUC	Sand Fill @	Operational	45	0	2	1	15	
								Total This code:		4	50	
E5	GRADER	1	EAST-AREA	SUBSTRUC.	Excavation	Operational	45	0	1	18	352	
E5	GRADER	1	EAST-AREA	LANDSCAPE	Excavation	Operational	45	0	1	1	18	
E5	GRADER	1	WEST-AREA	SUB-STRUC	Excavation	Operational	45	0	2	11	212	
E5	GRADER	1	SOUTH-AREA	LANDSCAPE	Excavation	Operational	45	0	1	1	27	
								Total This code:		43	822	
										:		
E10	POWER TOOLS	2	SOUTH-AREA	FINISH	Windows	Operational	45	0	1	29	0	(1 CR7)
E10	POWER TOOLS	2	SOUTH-AREA	FINISH	Wood Floorin	Operational	45	0	1	480	0	(1 CR7)
E10	POWER TOOLS	1	SOUTH-AREA	FINISH	Insulation	Operational	45	0	1	394	0	
E10	POWER TOOLS	1	SOUTH-AREA	LANDSCAPE	Excavation	Operational	45	0	1	1	0	(1 CR1)
								GRAND TOTAL	:	646	8526	

Fig. 7.14: Resource report (equipment).

```
Modify  Save  Print  Change-Name  Worksheet  QUIT
Wages-Salaries  Gen-Expenses  Site-Install.  Oper-of-S.I.  Miscell.  Return

    A      B      C    D    E    F      G       H       I
1   =============== INDIRECT COST ESTIMATION ======================
2                   Project : NOSA.IND
3
4
5              A. WAGES  AND  SALARIES        :          103,500.0
6
7              B. GENERAL OFFICE EXPENSES     :           15,000.0
8
9              C. SITE   INSTALLATIONS        :           12,200.0
10
11             D. OPERATION OF SITE INSTALL.  :              750.0
12
13             E. OTHERS                      :            4,000.0
14                                                    ---------------
15
16                      TOTAL INDIRECT COST = $          135450.00
17
18
19
20
```

Fig. 7.15: Spreadsheet for indirect costs.

```
    A          B        C    D    E    F      G       H       I
29  =============== INDIRECT COST ESTIMATION =======================
30  A. WAGES & SALARIES
31  ==================
32
33                    CODE 9100 - SUPERVISION
34                    -----------------------
35                       No.      PERIOD         RATE/
36                                (Months)       Month
37
38  PROJECT MANAGER :       1 X       20 X      3,500.0 =      70,000.0
39  PROJ. SUPERINT. :       0 X        0 X          0.0 =           0.0
40  GEN.  SUPERINT. :       0 X        0 X          0.0 =           0.0
41  ASS.       ,,   :       0 X        0 X          0.0 =           0.0
42  TRADE SUPOERINT.:       0 X        0 X          0.0 =           0.0
43  MASTER MECHANIC :       0 X        0 X          0.0 =           0.0
44  ASS.        ,,  :       0 X        0 X          0.0 =           0.0
45  OTHER           :       0 X        0 X          0.0 =           0.0
46  OTHER           :       0 X        0 X          0.0 =           0.0
47
48
```

Fig. 7.16: Indirect cost estimation.

project expected competition and all other factors that contribute to a better classification of the project degree of risk. Accordingly, optimum markup and probability of winning the job are estimated. Upon returning from **DBID** session (similar to that described in Chapter 6), the user is prompted to confirm his optimum markup estimate (a 3.5% markup is assumed practical in the present hypothetical example) and proceed with bid unbalancing and summarization (option 7 in the project menu). Accordingly, the cash flow optimization algorithm is activated. The algorithm first reads the schedule and cost data and then prompts the user to confirm the optimization constraints (Fig. 7.17). The user can change the default values for interest rate(%), monthly retention(%), lower and upper limits for the unit prices, and monthly cash flow limit ($). Based on these data, optimization calculations are performed and a bid summary report is viewed on the screen (Fig. 7.18). The report shows detailed direct costs associated with each contract item. The total of the indirect cost and the profit are optimally distributed among the different contract items. The user can manually adjust the amount of indirects assigned to each contract item and the unit price of that item is modified and so as the total bid value. The user is always provided with the total assigned and the remaining portion yet to be. The bid summary report can also be printed with all the details as shown in Fig. 7.19 and the bid could be prepared for submission to the owner.

Project : nosa

Cash Flow Optimization Data

Interest Rate: 1 % monthly
Cash Flow Limit/m. ($ excluding revenues): 471725

Article # : 1 Excavation
Quantity : 1000 Direct Cost ($): 208629.8
Unit Price ==> Lower Limit: 208.6298 Upper Limit: 250.3557

Select Option:

<N> Move to NEXT Article.
<1> Modify Interest Rate (%).
<2> Modify Monthly Retention (%).
<3> Modify Lower Limit for Unit Price.
<4> Modify Upper Limit for Unit Price.
<5> Modify Cash Flow Monthly ($) Limit. <E> EXIT.

Fig. 7.17: Screen for optimization constraints.

Project: nosa.prj - Summary Report.
Total Directs($)= 2197489 - Total Indirects($)= 135450 - Markup= 3.5 (%)

.... Distribution of Indirects + Markup ....
Distribute($) ==> 212362 - Assigned: 212362 - Remaining: 1

Art.	Desc.	Unit	Quant.	Labor	Equip.	Mat.	Subs	Other	Total direct	Indirect	Unit price
				....	($) x 1000		....				
1.00	Excavat	cuyd	1000	8	1	0	200	0	209	+42	0
2.10	Piled f	unit	20	0	0	0	45	0	45	+9	3
2.20	Isolate	unit	25	38	7	60	0	0	104	+21	5
3.00	Superst	unit	20	205	1	201	25	0	431	+86	26
4.00	Brick w	sqft	2000	300	0	158	0	0	458	+55	0
5.00	Electri	lsum	1	26	0	0	0	0	26	+0	26
6.00	Mechani	lsum	1	114	0	9	400	0	523	+0	523
7.00	Interio	unit	20	221	0	66	0	0	286	+0	14
8.00	landsca	lsum	1	0	0	0	115	0	116	+0	116
GRAND TOTALS				918	9	494	785	0	2197	+212	

.... Use Up & Down Arrows to View all Articles ....
<ESC-QUIT> <F1-PRINT REPORT> <F2-DISTRIBUTE> <F3-% AGES> Total Bid: 2409851

Fig. 7.18: Screen for manual bid-unbalancing.

Project: nosa.prj - Bid Summary Report.

Total Directs ($)= 2197489
Total Indirects($)= 135450
.... Markup(%)= 3.5

Total to Distribute($) = 212362 - Assigned: 212362 - Remaining: 1

Art. #	Description	Unit	Quantity	Labor cost	Equipment cost	Material cost	Subs cost	Other cost	Total direct	Indirect cost	Unit price
1.00	Excavation	cuyd	1000	7853	776	0	200000	0	208630	+41726	250.4
2.10	Piled foundations	units	20	0	0	0	45000	0	45000	+9000	2700.0
2.20	Isolated Footings	units	25	37504	6912	59582	0	0	103998	+20800	4991.9
3.00	Superstructure	units	20	204518	792	201125	25000	0	431435	+86287	25886.1
4.00	Brick work	sqft	2000	299810	0	158400	0	0	458210	+54549	256.4
5.00	Electrical work	lsum	1	25636	0	0	0	0	25636	+0	25636.0
6.00	Mechanical work	lsum	1	113842	0	9040	400000	0	522882	+0	522882.4
7.00	Interior Decorati	units	20	220554	0	65640	0	0	286194	+0	14309.7
8.00	landscaping	lsum	1	459	45	0	115000	0	115504	+0	115504.4
GRAND TOTALS				910176	8526	493787	785000	0	2197489	+212362	

total Bid: 2409850.5

(%) Labour of direct cost = 41.4
(%) Equip. of direct cost = 0.4
(%) Mater. of direct cost = 22.5
(%) Subs of direct cost = 35.7
(%) Others of direct cost = 0.0

(%) Directs of total bid = 91.2
(%) Indirects of total bid= 8.8

Fig. 7.19: Bid summary report.

7.4 Conclusion

A prototype microcomputer system (**ESTIMATOR**) for bid preparation is developed based on the structured methodology proposed in this study. The prototype exhibits improved efficiency in preparing a bid estimate in a competitive bidding environment. **ESTIMATOR** is developed to automate the bid estimation process proposed and provide a user friendly interface. An example application is presented in an effort to illustrate the essential features of the system and demonstrate the operations of the prototype developed. The prototype provides an efficient quantitative as well as qualitative assessment of direct and indirect costs, incorporating the impact of the project schedule and cash flow limitations.

Based on the prototype operation, the system has several interesting features:

- Structured, modular architecture allowing future expansions and enhancements.
- Efficient, user interactive and fast processing on IBM machines and compatibles.
- Transparent, changes in the databases are simply reflected on the estimate.
- Practical, databases are saved and downloaded with the project files for records.
- Flexible, all project files could be renamed and used as templates.

Chapter 8

CONCLUSION

8.1 Conclusion

A structured methodology for cost estimation and bid preparation is developed to enable contractors prepare practical as well as competitive bid estimates. The methodology employs a work breakdown structure (WBS) that is directly linked to contract items, a unified code of accounts, and resource productivity databases. This enables efficient estimation of direct and indirect costs, incorporating the impact of the project schedule. The methodology further integrates two techniques: Neural Networks and linear programming. Neural Networks are used to develop an analogy-based decision support system (D.S.S.) for bidding and optimum markup estimation considering the project risk pattern. Linear programming is used to unbalance the bid and arrive at optimum unit prices associated with the contract items, considering the contractor's cash flow limitations. A prototype PC-based software system (**ESTIMATOR**) is developed to automate the bid preparation process and provide a user friendly interface. An example application is presented in an effort to illustrate the essential features of the system and demonstrate its effectiveness and practicality.

Based on the prototype performance, the apparent advantages of the proposed system over available tools lie in its incorporation of adequate assessment of the major elements of the bid preparation process and in the successful functioning of the integration scheme adopted. The procedures used within the system four modules, despite current limitations, exhibit practical and fairly simple developments that suit the nature of the sub-problems involved, and can produce decisions that are adequate, considering the time and information limitations of the bidding stage. The adopted cycle of cost and schedule refinements proved to be promising in developing accurate cost and schedule baselines.

It is evident from the study that neural networks have interesting problem-solving capabilities suited to a number of problems in the field of construction engineering and management. In contrast to traditional reasoning-intensive AI tools such as knowledge-based expert systems (KBES), NNs employ a learning mechanism and emulate the human ability to learn from examples and solve pattern recognition tasks similar in nature to that performed according to the current industry practice in using "gut feeling" and analogy with previous cases. Despite the difficulties generally associated with setting a neural network model for a particular problem and sometimes the large hardware requirements, NNs are worthwhile to be utilized for the class of problems described. Several of the guidelines provided in this study may, however, be used to structure the development process and help develop practical applications particularly using the backpropagation NN paradigm.

The developments made in this study with respect to the markup estimation problem demonstrates the powerful generalization capabilities of neural networks.

As opposed to the algorithmic probability-based models, developed to provide optimum markup solutions, the decision support system developed in this study (**DBID**) has several interesting features and advantages, including the following:

- It incorporates a model that is built and refined based on a representation of the risk pattern and the resulting **actual** outcomes of a large number of projects.
- It provides a decision aid in terms of the % markup, in addition to some indication about the implications of such markup on other important aspects including actual project profitability.
- It has a user friendly interface that facilitates storage of previous bid encounters, efficient data editing, model adaptation to the user's environment, and necessary computations.
- It incorporates sensitivity analyses to account for the uncertainty that may exist in the user assessments of the project risks.
- It provides a measure of the probability of winning at any desired level of markup.
- It facilitates integration with other modules for planning and scheduling and cost estimation, forming a comprehensive bid preparation framework.
- It derives solutions instantaneously, with no reasoning and/or deduction involved, rather utilizing a form of analogy and pattern recognition.

The present D.S.S. produces not only an optimum markup value but also provides the decision-maker with some indications about the implications of such markup decision. This includes: win/lose possibility; estimated difference in ($) between the winner and second lowest bidder; anticipated actual profitability; project potential for change orders; project potential for claims; and expected duration extension. These conclusions are sometimes significant not only for markup estimation but also for the analysis of bid/no-bid problem. The D.S.S. also may not be restricted for use by contractors at the bidding stage, owners and construction managers may also find these indications useful when categorizing the project's degree of risk and accordingly decisions pertaining to the type of contractual relationships with prospective contractors could be made.

8.2 Future Research

There are several potential improvements to the bid preparation prototype proposed in this study and other areas of future research related to the methodology adopted. These may include:

- A complete field trial could be conducted using the bid preparation prototype in order to identify possible areas of modifications and practical enhancements to the prototype.
- The planning and scheduling module of the prototype could be developed in a more detailed and comprehensive manner. Different units in a repetitive activity could have different quantities. Effect of learning curve could be

considered. Relationship to a specific unit in a repetitive activity need to be permitted. Weather sensitive activities could be pre-identified and the application of productivity factors is then restricted to those activities.

- For practicality, the complete system could be re-written using an object oriented programming language, making use of its class structure and inheritance capabilities, and thus making further modifications to the system easier.
- Other modules could be developed and integrated with the available prototype. These modules may pertain to: resource levelling, progress reporting, generating procurement plans for equipment and material resources, and contingency allocation.
- WBS templates could be incorporated into the system for typical types of projects. The user can retrieve the template and adjust the activities, quantities, number of repetitions, and contract items based on the actual project in hand. This approach may save bid preparation time, eliminate omission mistakes, and facilitate the identification of logical relationships among activities.
- An expert system could be developed for generating an automated WBS for projects based on user input of the heuristics used by contractors to divide the project into areas and measurable tasks of work.
- The user may be allowed to input actual start and finish dates of the activities. Control operations and progress reporting options could be added

as an additional module.

- Neural networks could be integrated with algorithmic and/or KBES modules. KBES could also be utilized to provide reasoning and explanation facilities, and to include the industry simple rules of thumb. The combination of these techniques will be more robust than either working alone, and can offer better decision aid to contractors. The role of neural networks in integrated applications is illustrated in Fig. 8.1 (Bailey and Thompson 1990b).
- The NN methodology described in this thesis could readily be applied in other domains in construction management where solutions are generated primarily based on analogy, and where traditional algorithmic tools may prove inadequate. Some potential areas of applications are outlined in Table 8.1. Future widespread of parallel distributed computer architectures may also encourage the implementation of construction problems using neural networks, and thus more speedy solutions and decisions could be made.
- Research work could be directed towards the interpretation of causal rules from the networks weights, to be used for explanation capabilities.
- More flexible and easy to use neural network development environments could be developed to facilitate the application development process.

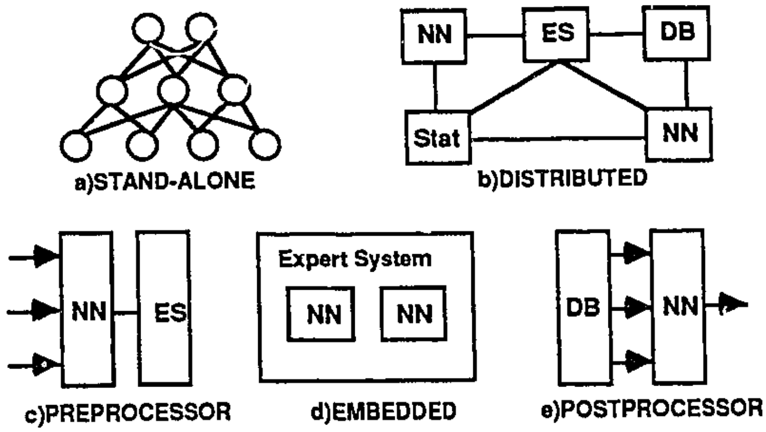


Fig. 8.1: Role of neural networks in integrated applications.

Table 8.1: Potential applications of widely utilized Neural Network paradigms in construction engineering and management.

PARADIGM	POSSIBLE APPLICATIONS
Back-propagation	Suitable for modelling problems under a static environment where, problem domain is rich in historical examples and no incremental learning or real-time performance are needed. Applications include: **a)- Selection between alternatives** (Multiple criteria decision-making, e.g. Wang and Malakooti 1990), examples: 1. Selecting prospective contractors or designers. 2. Selecting an overall contracting approach. 3. Organization design (company or project level). 4. Remidal actions for project control. 5. Equipment selection. 6. Formwork type selection. 7. Foundation type selection. 8. Dewatering method selection. **b)- Estimation and Classification**, examples: 1. Estimating the impact of project environment on the productivity levels of working trades. 2. Developing parametric time and cost models of construction projects. 3. Forcasting durations and costs of remaining activities on projects. 4. Classifying the performance of ongoing projects. 5. Estimating percent markup for bidding on projects. 6. Classifying projects in terms profitability and potential for claims and change orders. 7. Projecting cost, time or resource overruns. **c)- Function Synthesis:** Prototyping complicated functions or large time-consuming computer programs to provide approximate and apeedy solutions to very urgent problems, such as complicated stress analysis problems (e.g., Vanluchene and Sun 1990).
Counter-propagation	Suitable for modelling problems under a static environment. Fast to implement, however, accuracy is moderate (as compared to Back-propagation), examples: **- (a), (b) and (c) above.** **- Diagnosis problems**, examples: 1. Reasons for cost, time and resource overruns. 2. Claim analysis. 3. Analysis of structural defects and failures. 4. Analysis of safety related problems.
Hopfield	**- Complex Optimization problems** (e.g. Ramanujam and Sadayappan 1988), examples: 1. Time, cost and resource utilization. 2. Time-cost trade off strategy. 3. Site layout plan. **- Scheduling and Sequencing** (e.g. Flood 1989), examples: 1. Sequencing of construction operations. 2. Job-shop scheduling. **- Modelling real-time and dynamic environments,** examples: 1. Modelling the effect of escalation and inflation rates on time and cost projections.
B A M	Suitable for modelling real-time and dynamic environment similar to Hopfield. Responds to noisy, incomplete and partially incorrect data.
A R T	Suitable for applications that require incremental learning, examples: 1. Accumulation of experiences in the execution of various construction projects. 2. Accumulation and analyses of historical and short-term market trends.

REFERENCES

Adeli, H. (ed.), "Expert Systems in Construction and Structural Engineering," Chapman and Hall Ltd., 1988.

Adrian, J. "Construction Productivity Improvement," Elsevier Science Publishing Co. Inc., 1987.

Ahmad, I. Decision-Support System for Modeling Bid/No-Bid Decision problem. **Journal of Construction Engineering and Management**, ASCE, Vol. 116, No. 4, 595-608, Dec. 1990.

Ahmad, I. and Minkarah, I. "Decision Analysis and Expert System Technology: A Construction Industry Perspective," **Proceedings of the International Symposium on Building Economics and Construction Management**, CIB, Sydney, Australia, 1990, 363-372.

Ahmad, I. and Minkarah, I. "An Expert System for Selecting Bid Markups," **Proceedings of the Fifth Conference on Computing in Civil Engineering,** Alexandria, Virginia, March, 1988a, pp. 229-238.

Ahmad, I., and Minkarah, I. "Questionnaire Survey on Bidding in Construction," **Journal of Management in Engineering**, American Society of Civil Engineers, Vol. 4, No. 3, July, 1988b, pp. 229-243.

Ahmad, I., and Minkarah, I. "An Additive Utility Model for Selecting Optimum Bid Price," **Proceedings of the Eighteenth Annual Pittsburg Conference on Modeling and Simulation**, 18, Part 1, April, 1987 a, pp. 367-373.

Ahmad, I., and Minkarah, I. "Optimum Markup for Bidding: A Preference-Uncertainty Trade-off Approach," **Civil Engineering Systems**, Vol. 4, 170-174, Dec. 1987 b.

Ahuja, N. and Campbell, W.J. "Estimating, from Conception to Completion," Prentice-Hall Inc., 1988.

Al-Bahar, J., and Gerwick, B. "Setting-Up a Risk Management Policy in Contracting Firms," **Proceedings of the Project Management Institute Annual Seminar/ Symposium**, Calgary, Alberta, Canada, Oct. 1990, pp. 702-707.

Al-Bahar, J. and Crandall, K. "Systematic Risk Management Approach for Construction Projects," **Journal of Construction Engineering and management**, ASCE, Vol. 116, No. 3, Sept. 1990a, pp. 533-546.

Al-Bahar, J. and Crandall, K. "Risk Management in Construction Projects: A Systematic Approach for Contractors," **Proceedings of the International Symposium on Building Economics and Construction Management**, CIB, Sydney, Australia, 1990b, pp. 43-55.

Allwood, R.J. (1989). Techniques and Application of Expert Systems in the Construction Industry. Ellis Horwood Limited, England.

Al-Tabtabai, H. and Diekman, J. "PROCON: A Knowledge Based Approach to Construction Project Control," **Proceedings of the International Symposium on Building Economics and Construction Management**, CIB, Sydney, Australia, 1990, pp. 385-397.

Anderson, J.A. "Data Representation in Neural Networks," **AI Expert**, June 1990, pp. 30-37.

Arditi, D., and Albulak, M. Z. "Comparison of Network Analysis with Line of Balance in a Linear Repetitive Construction Project," **Proceedings of the 6th INTERNET Congress**, Vol. 2, Munchen, Germany, 1979, pp.13-25.

Arditi, D., and Riad, N., "Commercially Available Cost Estimating Software Systems," **Project Management Journal**, Vol., XIX, No., 2, 1988, pp., 65-70.

Arai, M. "Mapping Abilities of Three-Layer Neural Networks," **Proceedings of the International Joint Conference on Neural Networks**, Neural Network Society, Washington D.C., June 1991, pp. I.419-I.423.

Austin, S. "An Introduction to Genetic Algorithms," **AI Expert**, March 1990, pp. 49-53.

Badger, A. et al. "Project Control System Integration," Project Control: Needs and Solutions: **Conference Proceedings, Construction Division of the ASCE**, New York: ASCE, June 1987, pp. 88-100.

Bailey, D., and Thompson, D. "How to Develop Neural-Network Applications," **AI Expert**, June, 1990 a, pp. 38-47.

Bailey, D., and Thompson, D. "Developing Neural-Network Applications," **AI Expert**, Sept. 1990 b, pp. 34-41.

Bailey, A. "Automatic Evolution of Neural Net Architectures," **Proceedings of the International Joint Conference on Neural Networks**, Vol. 1, Washington D.C., June 1990, pp. I.589-I.592.

Barletta, R. "An Introduction to Case-Based Reasoning," **AI Expert**, Aug. 1991, pp. 235-242.

Barton, P. (ed.) "Information Systems in Construction Management: Principles and Applications," Batsford Academic and Educational London, 1985.

Battiti, R. "Optimization Methods for Back-propagation: Automated Parameter Tuning and Faster Convergence," **Proceedings of the International Joint Conference on Neural Networks**, Washington D.C., Vol. 1, June 1990, pp. I.593-I.596.

Baum, E. and Haussler, D. "What Size Net Gives Valid Generalization," **Neural Computation,** Vol. 1, 1989, pp.151-160.

Baxendale, T. "Construction Management Information Systems for Production Control," **Proceedings of the International Symposium on Building Economics and Construction Management**, CIB, Sydney, Australia, 1990, pp. 80-91.

Bechtel Power Corporation, "Synergy System Description," Norwalk, California, April, 1989.

Benjamin, D. et al. "Knowledge-Based Prototype for Improving Scheduling Productivity," **Journal of Computing in Civil Engineering**, ASCE, 4(2), April, 1990, pp. 124-133.

Benjamin, N., and Meador, R. "Comparison of Friedman and Gates Competitive Bidding Models," **Journal of the Construction Division**, ASCE, Vol. 105, No. CO1, 1979, pp. 25-40.

Birnie, J., and Yates, A. "Cost Prediction Using Decision/Risk Analysis Methodologies," **Construction Management and Economics**, Vol. 9, 1991, pp. 171-186.

Birrell, G. S. "Construction Planning-Beyond the Critical Path," **Journal of the Construction Division**, ASCE, Vol. 106, No. CO3, Sept. 1980.

Bishop, J., Bushnell, M., Usher, A., and Westland, S. "Neural Networks in the Colour Industry," in **"Applications of AI in Engineering VI"**, Rzevski and Adey (eds), Computational Mechanics Publications, Elsevier Applied Science, 1991, pp. 423-434.

Bochereau, L. and Bourgine, P. "Extraction of Semantic Features and Logical Rules From a Multilayer Neural Network," **Proceedings of the International Joint Conference on Neural Networks**, Vol. 2, 1990, pp. II.579-II.582.

Bcwen, P.A., and Erwin, G.J. "The Impact of Second Generation Expert Systems on Building Design Evaluation," **Proceedings of the International Symposium on Building Economics and Construction Management**, CIB, Australia, 1990, pp. 419-427.

Brandon, P.S. "Expert Systems - After the Hype is Over," **Proceedings of the International Symposium on Building Economics and Construction Management**, CIB, Sydney, Australia, 1990, pp. 314-346.

Brent, R. "Fast Training Algorithms for Multilayer Neural Nets," **IEEE Transactions on Neural Networks**, Vol. 2, No. 3, May 1991, pp. 346-354.

Burr,D. J., "Experiments with a Connectionist Text Reader," **Proceedings of the First International Conference on Neural Networks,** eds. M. Caudill and C. Butler, Vol. 4, pp. 717-24, 1987, San Diego, CA:SOS Printing.

Cardoso, J. "Revisiting the Towers of Hanoi," **AI Expert**, Oct. 1991, pp.49-53.

Carr, R. I., "General Bidding Model," **Journal of the Construction Division**, American Society of Civil Engineers, Vol. 108, No. CO4, Dec., 1982, pp. 639-650.

Carr, R. I., "Optimum Markup by Direct Solution," **Journal of the Construction Engineering and Management**, American Society of Civil Engineers, Vol. 113, No. 1, March, 1987, pp. 138-150.

Carr, R., "Cost-Estimating Principles," **Journal of Construction Engineering and Management**, ASCE, Vol., 115, No., 4, December, 1989, pp. 545-551.

Carpenter, G., and Grossberg, S., "A Massively Parallel Architecture for a Self-organizing Neural Pattern Recognition Machine," **Computer Vision, Graphics, and Image Processing**, 37: 54-115, 1987 a.

Carpenter, G., and Grossberg, S., "ART-2: Self-organization of Stable Category Recognition Codes for Analog input Patterns," **Applied Optics**, 26(23): 4919-30, 1987 b.

Castelaz, P., Angus, J., and Mahoney, J. "Application of Neural Networks to Expert Systems and Command and Control Systems," **IEEE Western Conference on Expert Systems**, Anaheim, CA, 1987, pp. 118-125.

Caudill, M., The View From Now. **AI Expert**, June 1992, pp. 24-31.

Caudill, M. "Neural Network Training Tips and Techniques," **AI Expert**, Jan. 1991a, pp. 56-61.

Caudill, M. "Avoiding the Great Back-Propagation Trap," **AI Expert**, July 1991b, pp. 29-35.

Caudill, M. "Evolutionary Neural Networks," **AI Expert**, March 1991c, pp. 28-33.

Caudill, M. "Using Neural Nets, Part IV, Making an Expert Network," **AI Expert**, July 1990, pp. 41-45.

Caudill, M. "Using Neural Nets, Part I: Representing Knowledge," **AI Expert**, Dec. 1989, pp. 34-41.

Caudill, M. "Neural Networks Primer, Part III," **AI Expert**, June 1988, pp. 19-23.

Caudill, M. and Butler, C "Naturally Intelligent Systems," MIT Press, Mass, 1990.

Chan, L.W., and Fallside, F. "An Adaptive Training Algorithm for Back-Propagation Networks," **Computer Speech & Language**, 2, pp. 205-218, 1987.

Chang, F. and Lipman, R. "Using Genetic Algorithms to Improve Pattern Classification Performance," in Touretsky, D. (ed.) **"Advances in Neural Information Processing Systems"**, Morgan-Kaufmann, April 1991.

Chester, D. "Why Two Hidden Layers are Better Than One," **Proceedings of the International Joint Conference on Neural Networks**, Washington D.C., Jan. 1990, Vol. 1, pp. I.265-I.268.

Cheu, R., Ritchie, S., Recker, W., and Bavarian, B. "Investigation of a Neural Network Model for Freeway Incident Detection," in "AI and Civil Engineering" B.H.V. Topping (ed.), **Proceedings of the Second International Conference on the Application of AI to Civil and Structural Engineering**, Oxford, England, Civil-Comp Press, Edinburgh, 1991, pp. 267-274.

Clough, R. 1986. "Construction Contracting," 5th Edition, John Wiley & Sons Inc.

Cohen, M.A., and Grossberg, S.G., "Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks," **IEEE Transactions on Systems, Man and Cybernetics**, 1983, 13: 815-826.

Collier, K. "Fundamentals of Construction Estimating and Cost Accounting," 2nd. ed., Englewood Cliffs, N.J., Prentice-Hall, Inc., 1987.

Copare, P.B. "Cash Flow Management Through Front-End Loading," **AACE Transactions**, 1990, pp. A.e.1-A.5.6.

Crooks, T. "Care and Feeding of Neural Networks," **AI Expert**, July 1992, pp. 36-41.

Davis, E. W. "CPM Use in Top 400 Construction Firms," **Journal of the Construction Division**, ASCE, Vol. 100, No. CO1, 1974, pp. 39-47.

Davis, G., and Olson, M. "Management Information Systems: Conceptual Foundation, Structure and Development," 2nd Edition, McGraw-Hill, 1988.

Derthick, M., "A Connectionist Architecture for Representing and Reasoning about structured knowledge," **Proceedings of the Ninth Annual Conference of the Cognitive Science Society**, 1987, Seattle, WA.

Dodd, N. "Optimization of Neural Network Structure Using Genetic Techniques," in **"Applications of AI in Engineering VI"**, Rzevski and Adey (eds), Computational Mechanics Publications, Elsevier Applied Science, 1991, pp. 939-944.

Dixie, J. "Bidding Models-the Final Resolution of a Controversy," **Journal of the Construction Division**, ASCE, Vol. 100, No. CO3, 1974, pp. 265-271.

Duns & Bradstreet. Canadian Business Directory. Duns & Bradstreet Canada Limited, P.O.Box 6200, Station "A", Mississauiga, ON, L5A 4G4, 1990.

Dutta, S. and Shekhar, S. "Bond-Rating: A non Conservative Application of Neural Networks," **Proceedings of the IEEE International Conference on Neural Networks**, Vol. II, San Diego, California, 1988, pp. 443-450.

Eberhart, R., and Dobbins, R. (eds.) "Neural Network PC Tools: A Practical Guide," Academic Press, Inc., San Diego, California, 1990.

Echeverry, D. et al. "Construction Schedule Generation Using AI Tools," **Proceedings of the 6th ASCE Conference on computing in civil Engineering,** 6th Conference, Atlanta, Georgia, Sept., 1989, pp. 44-51.

Ehrenfeucht, A., Heussler, D., Kearns, M. and Valiant, L.G. "A General Lower Bound on the Number of Examples Needed for Learning," **Information and Computation**, Vol. 82, Academic Press Inc., 1989, pp. 247-261.

Eldin, N., "Measurement of Work Progress: Quantitative Technique," **Journal of Construction Engineering and Management**, ASCE, Vol. 115, No. 3, September, 1989, pp.462-474.

Elman, J.L. and Zipser, D. "Learning the Hidden Structure of Speech," **Journal of the Acoustical Society of America**, Vol. 83, No. 4, 1988, pp. 1615-1626.

Engineering News Record (ENR). Top 400 Contractors. May 24, 1990.

Fahlman, S.E., "An Empirical Study of Learning Speed in Back-Propagation Networks," Technical report CMU-CS-88-162, Pittsburgh, PA: Carnegie Mellon University, 1988.

Fahlman, S.E., and Hinton, G.E., "Connectionist Architectures for Artificial Intelligence," **Computer**, Vol. 19, 1986, pp. 100-108.

Flanagan, R., and Norman, G., "Sealed Bid Auction: An Application to the Building Industry," **Construction Management and Economics**, Vol. 3, 1985.

Flood, I. "A Neural Network Approach to the Sequencing of Construction Tasks," **Proceedings of the 6th International Symposium on Automation and Robotics in Construction**, ISARC, San Francisco, CA, 1989, pp. 204-211.

Flood, I. "Solving Construction Operational Problems using Artificial Neural Networks and Simulated Evolution," **Proceedings of the International Symposium on Building Economics and Construction Management**, CIB, Sydney, Australia, 1990, pp. 197-208.

Frantz, W.F. "Systematic Risk Management Impacts Hybrid System Projects," **Proceedings of the Project Management Institute Annual Seminar/ Symposium**, Calgary, Alberta, Canada, Oct. 1990, pp. 275-286.

Franzini, R.A., and Almeida, L.B. "Acceleration Techniques for the Back-Propagation Algorithm," Lecture Notes in Computer Science, 412, pp. 110-119, 1987.

Friedman, L. "A Competitive Bidding Strategy," **Operations Research**, Vol. 4, 1956, pp. 104-112.

Fung Fai, NG "Knowledge Base Systems: Their Impact on the Construction Industry," **Proceedings of the International Symposium on Building Economics and Construction Management**, CIB, Sydney, Australia, 1990, pp. 549-560.

Gaarslev, A. "Neural Networks - Techniques Tested on Classical Bidding Strategy Data Set," **Proceedings of the European Symposium on Management Quality and Economics in Housing and Other Building Sector**, Portougal, 1991.

Gallant, S. "Connectionist Expert Systems," **Communications of the ACM**, 31, 1988, pp. 152-69.

Garrett Jr., J. "Neural Networks and their Applicability within Civil Engineering," **Proceedings of the 8th Conference on Computing in Civil Engineering**, ASCE, Dallas, Texas, 1992, pp. 1155-1162.

Garson, G. "Interpreting Neural-Network Connection Weights," **AI Expert**, April 1991, pp. 47-51.

Gates, M., "Bidding Strategies and Probabilities," **Journal of the Construction Division**, American Society of Civil Engineers, Vol. 93, No. CO1, March, 1967, pp. 75-107.

Goldberg, D. and Kuo, C.H. "Genetic Algorithms in Pipeline Optimization," **Journal of Computing in Civil Engineering**, ASCE, VOl. 1, No. 2, April 1987, pp. 128-141.

Goodman, D., and Burmeister, H., "A Computational Algorithm for Multi-contract Bidding Under Constraints," **Management Science**, Vol. 22, 1976.

Gorry, G., and Morton, M. "A Framework for Management Information Systems," **Sloan Management Review**, 13, No. 1, Jan. 1971.

Griffis F. H. "Bidding Strategy: Winning Over Key Competitors," **Journal of Construction Engineering and Management**, ASCE, 118(1), 1992, 151-165.

Grinyer, P.H., and Whittaker, J.D., "Managerial Judgement in a Competitive Bidding Model," **Operations Research Quarterly**, Vol. 24, 1973.

Grossberg, S., "Classical and Instrumental Learning by Neural Networks," Progress in Theoretical Biology, 1974, Vol. 3, pp. 51-141. New York: Academic Press.

Grossberg, S., "Studies of mind and brain," Boston: Reidel, 1982.

Halpin, D., "Financial & Cost Concepts for Construction Managers," John Wiley & Sons Inc., 1985.

Hamburger, D. "Uncertainty Contingency and Risk Assessment Tools for the Project Manager," **Proceedings of the Project Management Institute Annual Seminar/ Symposium**, Calgary, Alberta, Canada, Oct. 1990, pp. 267-274.

Harris, R. "Precedence and Arrow Networking Techniques for Construction," John Wiley & Sons, Inc., 1978.

Harris, F., and McCaffer, R. (eds) "Modern Construction Management (Competitive Bidding, Chapter 9)," Crosby Lockwood Staples, London, 1977.

Haussler, D. "Generalizing the PAC Model for Neural Net and Other Learning Applications," Technical Report UCSC-CRL-89-30, University of California Computer Research Laboratory, Santa Cruz, CA.

Haykin, S.S. "Adaptive Filter Theory," Prentice Hall, 1989.

Hebb, D.O., "Organization of Behaviour," New York: Science Editions, 1949.

Hecht-Nielsen, R., "Counterpropagation Networks," **Applied Optics** 26(23): 979-84, 1987.

Hecht-Nielsen, R., "Applications of Counterpropagation Networks," **Neural Networks**, 1:131-139, 1988.

Hegazy, T. and Moselhi, O. "An Analogy-Based Solution to the Markup Estimation Problem," Paper Accepted for Publication and Currently In Press, **Journal of Computing in Civil Engineering**, ASCE, 1993.

Hegazy, T., Moselhi, O. and Fazio, P. "BAL: An Algorithm for Scheduling and Control of Linear Projects," **1993 AACE Transactions**, AACE, 1993a.

Hegazy, T., Fazio, P., and Moselhi, O. "Developing Practical NN Applications Using Backpropagation," Paper Accepted for Publication and Currently In Press, **Microcomputers in Civil Engineering**, 1993b.

Hendrickson, C., and Au., T. "Project Management for Construction," Prentice Hall, Englewood Cliffes, N.J., 1989.

Hendrickson, C., et al. "Expert System for Construction Planning," **Journal of Computing in Civil Engineering**, ASCE, 1(4), 1987, pp. 253-269.

Hicks, J.C. "Heavy Construction Estimates; With and Without Computers," **Journal of Construction Engineering and Management**, ASCE, 118(3), 1992, 545-560.

Hinton, G.E., and Sejnowski, T.J., "Learning and Relearning in Boltzmann Machines," In **Parallel Distributed Processing**, 1986, Vol. 1, pp. 282-317, Cambridge, MA: MIT Press.

Hopfield, J.J., and Tank, D.W., "Neural Computation of Decisions in Optimization Problems," **Biological Cybernetics**, Vol. 52, 1985, pp. 144-152.

Hopfield, J.J., "Neurons with Graded Response have Collective Computational Properties like those of two-state Neurons," **Proceedings of the National Academy of Science**, U.S.A., 81:3088-92, 1984.

Hopfield, J.J., "Learning Algorithms and Probability Distributions in Feed-forward and Feed-back Networks", **Proceeding of the National Academy of Science,** U.S.A., 1987, Vol. 74, pp. 8429-8433.

Hopfield, J.J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities, " **Proceedings of the National Academy of Science,** U.S.A., 79:2554-58, 1982.

Hirose, Y., Yamashita, K. and Hijiya, S. "Back-Propagation Algorithm Which Varies the Number of Hidden Units," **Neural Networks,** Vol. 4, No. 1, 1991, pp. 61-66.

Ho, S. 1990 "WBS and Integrated Control," M. Eng. Major Technical Report, CBS, Concordia University, Montreal, Canada.

Holland, J. "Adaptation in Neural and Artificial Systems," University of Michigan Press, Ann Arbor, 1975.

Hornik, K. "Approximation Capabilities of Multilayer Feedforward Networks," **Neural Networks,** Vol. 4, No. 2, 1991, pp. 251-257.

Howel, J. "Inside a Neural Network," **AI Expert**, Nov. 1990, pp.29-33.

Ibbs, W., and Crandall, K. "Construction Risk: Multi-attribute Approach," **Journal of the Construction Division**, ASCE, Vol. 108, No. CO2, 187-200, 1982.

Ioannou, P. "Bidding Models: Symmetry and State of Information," **Journal of Management in Engineering,** ASCE, Vol. 114, No. 2, 1988, pp. 214-232.

Irie, B. and Miyake, S. "Capabilities of Three-Layered Perceptrons," **Proceedings of the IEEE Second Annual International Conference on Neural Networks,** June 1988, San Diego, CA, pp. I.641-I.647.

Isaac, A. "Analysis of Risks in Bidding Lump Sum Implementation Contracts," **Proceedings of the Project Management Institute Annual Seminar/ Symposium,** Calgary, Alberta, Canada, Oct. 1990, pp. 694-701.

Jaafari, A. "Criticism of CPM for Project Planning Analysis," **Journal of Construction Engineering and Management,** ASCE, 110(2), 1984, 222-233.

Jackson, P. Introduction to Expert Systems. Addison-Wesley, London, 1986.

Jacobs, R.A. "Increased Rates of Convergence Through Learning Rate Adaptation," **Neural Networks**, Vol. 1, No. 4, Elmsford, N.Y., Pergamon Press, 1988.

Jacobs, R.A. "Increased Rates of Convergence Through Learning Rate Adaptation," Technical Report COINS TR87-117, University of Massachusetts at Amherst, Amherst, MA, 1987.

Johnston, D. "Linear Scheduling Method for Highway Construction," **Journal of the Construction Division**, ASCE, Vol. 107, No. CO2, June 1981, pp. 247-261.

Kadaba, N., Nygard, E., Juell, P. and Kangas, L. "Modular Back-Propagation Neural Networks for Large Domain Pattern Classification," **Proceedings of the International Joint Conference on Neural Networks**, Washington D.C., Vol. 2, Jan. 1990, pp. II.551-II.554.

Kamarthi, S., Sanvido, V. & Kumara, S., "A Connectionist Vertical formwork Selection System," **Proceedings of the 8th Conference on Computing in Civil Engineering**, ASCE, Dallas, Texas, 1992, pp. 1171-1178.

Kangari, R. and Boyer, L. "Risk Management by Expert Systems," **Project Management Journal**, 1989.

Kangari, R. "Business Failure in Construction Industry," **Journal of Construction Engineering and Management**, ASCE, 114(2), 1988, 172-190.

Kano, N. "A Knowledge-Based System for Construction Planning and Scheduling - A Prototype System Based on the Down-from-the-Top Methodology," **Proceedings of the 7th ISARC**, Vol. 1, Bristol, England, June, 1990, pp. 303-310.

Karnin, E. "A Simple Procedure for Pruning Back-Propagation Trained Neural Networks," **IEEE Transactions on Neural Networks**, Vol. 1, No. 2, June 1990, pp. 239-242.

Karshenas, S. & Feng, X., "Application of Neural Networks in Earthmoving Equipment Production Estimating," **Proceedings of the 8th Conference on Computing in Civil Engineering**, ASCE, Dallas, Texas, 1992, pp. 841-847.

Kartam, N., and Levitt, R. "Intelligent Planning of Construction Projects," **Journal of Computing in Civil Engineering**, ASCE, 4(2), April, 1990, pp. 155-176.

Keisk, T. and Selby, K. "Automating Construction Estimating," **Proceedings of the Annual Conference and 1st Biennial Environmental Speciality Conference**, CSCE, Hamilton, ON, Canada, Vol. II-1, May, 1990, pp. 150-160.

Kerridge, A. "Contingency, Risk, and Sensitivity Analysis," In "**Engineering and Construction project Management**," Kerridge, A., and Vervalin, C. (eds), Gulf Publications Co., Houston, 1986.

Kidd, J. "Project Management Software - Are We Being Over-persuaded ?," **Journal of Project Management**, Vol. 8, No. 2, May, 1990, pp. 109-115.

Kim, J. J. "An Object Oriented Database for Project Control," **Proceedings of the 11th International Cost Engineering Congress**, 6th Annual Meeting, AFITEP, Paris, France, April, 1990.

Knaus, R. "Building Networks that Learn," **AI Expert**. March 1990, pp. 17-21.

Knaus, R. "Putting Knowledge into Nets," **AI Expert**, Sept. 1991, pp. 19-25.

Knode, C.S., and Swanson, L.A., "A Stochastic Model for Bidding," **Journal of the Operational Research Society**, Vol. 29, No. 10, 1978.

Kohonen, T., "Self-organization and Associative Memory," 2nd Ed., New York: Springer-Verlag, 1988.

Kohonen, T. "Self-organization and Associative Memory," **Series in information Sciences**, Vol. 8, Berlin, Springer Verlag, 1984.

Kosko, B., "Bi-directional Associative Memories," **IEEE Transactions on Systems, Man and Cybernetics**, 18(1): 49-60, 1987 a.

Kosko, B., "Competitive Adoptive Bi-directional Associative Memories," **Proceedings of the IEEE First International Conference on Neural Networks**, Vol. 2, San Diego, CA, 1987 b, SOS Printing, pp. 759-766.

Kolmogorov, A. "On the Presentation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition," **Doki Akad Nauk**, USSR, Vol. 144, pp. 953-956, 1957.

Laramee, J. "A Planning and Scheduling System for High-Rise Building Construction," Master's Thesis, Center for Building Studies, Concordia University, Montreal, Canada, 1983.

Lawrence, J. "Data Preparation for a Neural Network," **AI Expert**, Nov. 1991, pp. 34-41.

Le Cun, Y. "Generalization and Network Design Strategies," in **"Connectionism in Perspective,"** Pfeiffer et al. (eds), Elsevier Science Publishers B.V. (North Holland), 1989, pp. 143-155.

Levitt, R.E. "Expert Systems in Construction: State of the Art," in **"Expert Systems for Civil Engineers: Technology and Applications"**, Maher, M. L. (ed.), ASCE, NY, 1987, 85-112.

Lisboa, P. and Perantonis, S. "Complete Solution of the Local Minima in the XOR problem," Networks, in **"Computations in Neural Systems"**, Vol.2, No. 1, IOP Publishing Ltd., U.K., 1991.

Lumsden, P. "The Line of Balance Method," Pergamon Press Limited, Industrial Training Division, 1968.

Mansur, F. "The Line of Balance Method," Major Technical Report, Center for Building Studies, Concordia University, Montreal, Canada, 1990.

Manzanera, I., and Bubshait, K. "Improve Your Profits by Understanding Claim Analysis," **AACE Transactions,** 1991, pp. E.5.1-E.5.6.

Maren, A., Harston, C., and Pap, R. "Handbook of Neural Computing Applications," Academic Press, Inc., San Diego, California, 1990.

Mawdesley et al. "Time Chanage Charts for Scheduling Linear Projects," **Proceedings of the 6th ASCE Conference on computing in civil Engineering,** Atlanta, Georgia, Sept. 1989, pp. 613-620.

Means, R. J. "Mean's Cost Estimation," P.O. Box 800, 100 Construction Plaza, Kingston, Maryland 02364-0800, U.S.A., 1990.

McCaffer, R. et al. "A Site-Based Data Capture and Information System," **Proceedings of the International Symposium on Building Economics and Construction Management,** CIB, Sydney, Australia, 1990, pp. 291-300.

McCulloch, W.S. and Pitts, W.H. "A Logical Calculus for the Idea Immanent in Nervous Activity," **Bulletin of Mathematical Biophysics,** 1943, 5:115-133.

Meuller, F. "Integrated Cost and Schedule Control for Construction Projects," Van Nostrand Reinhold Company, New York, 1986.

Minkarah, I. and Ahmad, I. "Expert Systems as Construction Management Tools," **Journal of Management in Engineering,** ASCE, 5(2), 1989, 155-163.

Minsky, M.I., and Papert, S. "Perceptrons," Cambridge, MA, MIT Press, 1969.

Mohan, S. "Expert Systems Applications in Construction Management and Engineering," **Journal of Construction Engineering and Management**, ASCE, 116(1), 1990, 87-99.

Montana, D. and Davis, L. "Training Feedforward Neural Networks Using Genetic Algorithms," **Proceedings of the 11th IJCAI**, Vol. 1, 1989, pp. 762-767.

Moselhi, O. "Management of Multi-Small Projects: A Database Approach," **Proceedings of the Project Management Institute Annual Seminar/ Symposium**, Calgary, Alberta, Canada, 1990.

Moselhi, O. "BLDG 781: Project Acquisition and Control," Lecture notes, 1989.

Moselhi, O. and Lorterapong, P. "Least Impact Model for Resource Scheduling," **Canadian Journal of Civil Engineering**, April 1993.

Moselhi, O. and Hegazy, T. "Markup Estimation Using Neural Network Methodology," Paper Accepted for Publication, **Computing Systems in Engineering**, 1993a.

Moselhi, O. and Hegazy, T. Discussion of "Bidding Strategy: Winning Over Key Competitors," by F.H. (Bud) Griffis, 118(1), 1992, pp. 151-165. **Journal of Construction Engineering and Management**, ASCE, 119(3), 1993b.

Moselhi, O. and Hegazy, T. "Bidding for a Profit: A Hockey Stick Trend," **Proceedings of the Project Management Institute Annual Seminar/ Symposium**, Calgary, Alberta, Canada, Oct. 1990a, pp. 435-442.

Moselhi, O. and Hegazy, T. "Optimum Markup Estimation: A Comparative Study" **Proceedings of the 11th International Cost Engineering Congress, 6th AFITEP Annual meeting**, Paris, France, April 1990b, pp. M.2.1-8.

Moselhi, O. and Nicholas, M. "Hybrid Expert System for Construction Planning and Scheduling," **Journal of Construction Engineering and Management**, ASCE, Vol. 116, No. 2, June, 1990, pp. 221-238.

Moselhi, O. and Martinelli, A. "Analysis of Bids Using Multiattribute Utility Theory," **Proceedings of the International Symposium on Building Economics and Construction Management**, CIB, Sydney, Australia, 1990.

Moselhi, O., Hegazy T. and Fazio, P. "DBID: Analogy-Based DSS for Bidding in Construction," Paper Accepted for Publication, **Journal of Construction Engineering and Management**, ASCE, 119(3), 1993.

Moselhi, O., Hegazy, T. and Fazio, P. "Potential Applications of Neural Networks in Construction," **Canadian Journal of Civil Engineering**, Vol. 19, June 1992a, pp. 521-529.

Moselhi, O., Hegazy, T. and Lorterapong, P. "Towards the Development of an Optimum Problem-Solving Strategy," **Proceedings of the 7th International Conference on Artificial Intelligence in Engineering**, Waterloo, Ontario, Canada, July 1992b.

Moselhi, O., Hegazy T. and Fazio, P. "Neural Networks as Tools in Construction," **Journal of Construction Engineering and Management**, ASCE, Vol. 117, No. 4, Dec. 1991a, pp. 606-625.

Moselhi, O., Hegazy, T. and Fazio, P. "Markup Estimation Using AI Methodology," in "AI and Civil Engineering", B.H.V. Topping (ed.), **Proceedings of the 2nd International Conference on the Application of Artificial Intelligence Techniques to Civil and Structural Engineering**, Oxford, England, Civil-Comp Press, Sept. 1991b, pp. 257-266.

Moselhi, O., Hegazy, T. and Fazio, P. "A Hybrid Neural Network Methodology for Cost Estimation," **Proceedings of the 8th International Symposium on Automation and Robotics in Construction**, Stuttgart, Germany, June 1991c, V. 2, pp. 519-528.

Moselhi, O., Hegazy, T. and Fazio, P. "Practical Cost Estimating in a Competitive Environment," **Transactions of the AACE 35th Annual Meeting**, Seattle, Washington, U.S.A., June 1991d, pp. H.5.1-H.5.9.

Moselhi, O., Hegazy, T. and Fazio, P. "Potential Applications of Neural Networks in Construction," **Proceedings of the Annual conference of the Canadian Society of Civil Engineers** (CSCE), Vancouver, Canada, May 1991e.

Moselhi, O., Hason, S., Hegazy, T., and Nicholas, M. "Summary of Contributions to Conceptual and Definition Planning in the Control Systems and Software Sector," **A Supplementary Report to the Proceeding of the Project Management Institute Annual Seminar/ Symposium**, Calgary, Canada, 1990c.

Musen, A., and Van Der Lei, I. "Of Brittleness and Bottlenecks: Challenges in the Creation of Pattern-Recognition and Expert-System Models," In "**Pattern Recognition and Artificial Intelligence**," Elsevier Science Publishers B.V., North-Holland, 1988.

Nadel, N. "Unit Pricing and Unbalanced Bids," **Civil Engineering**, June 1991, pp. 62-63.

Navinchandra, V., Sriram, D., and Logcher, R. D. "GHOST: Project Network Generator," **Journal of Computing in Civil Engineering**, ASCE, ASCE, 2(3), 1988, pp. 239-254.

Neil, J. "Construction Cost Estimating for Project Control," Prentice Hall Inc., 1982.

NeuralWare "An Introduction to Neural Computing," Neural Networks Software User Guide, 1988.

Neuroshell, User Guide, Ward Systems Group Inc., 245 West Patrik Street, Fredrick, MD 21701, Vol. 1, 1989.

de Neufville, R., and King, D. "Risk and Need-for-Work Premiums in Contractor Bidding," **Journal of Construction Engineering and Management**, ASCE, Vol. 117, No. 4, Dec. 1991, pp. 659-673.

Ndekugri, I. and McCaffer, I. "Management Information Flow in Construction Companies," **Journal of Construction Management and Economics**, V. 6, 1988, pp. 273-294.

Nguyen, N. "Cases Impairing C/SCSC Application in Program Management," **Proceedings of the 11th International Cost Engineering Congress**, AFITEP, 6th Annual Meeting, Paris, France, April, 1990.

Nicholas, M. "Hybrid Expert System for Construction Planning and Scheduling," Master Thesis, Centre for Building Studies, Concordia University, Montreal, Canada, 1989.

O'Brien, J. "VPM Scheduling for High-rise Buildings," **Journal of the Construction Division**, ASCE, Vol. 101, No. CO4, Dec., 1975, pp. 895-905.

Paek, J. and Lavata, N. "Construction Information System: Integrating Data/Information Systems and Expert Systems," **Proceedings of the 7th ISARC**, Vol. 1, Bristol, England, June, 1990, pp. 327-333.

Pao, Y. H. "Adaptive Pattern Recognition and Neural Networks," Addison-Wesley Publishing Company Inc., 1989a.

Pao, Y. "Functional Link Nets: Removing Hidden Layers," **AI Expert**, April 1989b, pp. 60-68.

Pao, Y.H., "Autonomous Machine Learning of Effective Control Strategies with Connectionist-nets," **Journal of Intelligent and Robotic Systems**, Vol. 1, 1988, pp. 35-53.

Pao, Y.H., and Sobajic, D.J., "Autonomous Feature Discovery for Critical Clearing Time Assessment," **Proceeding of the Symposium on Expert System Applications to Power Systems**, Stockholm, Sweden, 1988.

Park, W. Construction Bidding for Profit. John Wiley and Sons, Inc., 1979.

Passanisi, P. U. "Project Management Software-What is the future, Making Project Control Systems Work," **Proceeding of the Construction Division**, ASCE, 1985.

Paulson, B.C., Jr., "Automation and Robotics for Construction," **Journal of Construction Engineering and management**, ASCE, Vol. 111, No. 3, Sept., 1985, pp. 190-207.

Peurifoy, R. and Oberlender, G. "Estimating Construction Costs," 4th Edition, McGraw Hill Inc., 1980.

Pitts, W.H. and McCulloch, W.S. "How We Know Universals: The Perceptron of Auditory and Visual Forms," **Bulletin of Mathematical Biophysics**, 9:127ff, 1947.

Pool, A. and Sriram, S. "Project Integration Through Reference Segments (IPM)," **Proceedings of the 11th International Cost Engineering Congress**, 6th Annual Meeting, AFITEP, Paris, France, April, 1990.

Precision Estimating, Timberline Software Corporation, 9405 S. W. Gemini, Beaverton, Oregon 97005, U.S.A.

Québec Construction. Les 200 Leaders De La Construction En 1989. 441 Boul. Lebeau, Saint-Laurent, QC, Canada, H4N 1S2, Sept. 1990.

Rabchuk, J. E. "The Computer Revolution: Waiting for a Winner," **Highway & Heavy Construction**, Feb. 1991, 24-25.

Ramirez, M.R. and Arghya, D. "A Faster Learning Algorithm for Backpropagation Neural Networks in NDE Applications," in "Artificial Intelligence and Civil Engineering", B.H.V. Topping (ed.), **Proceedings of the 2nd International Conference on the Application of AI to Civil and Structural Engineering**, Civil-Comp Press, Oxford, England, Sept. 1991, pp. 275-283.

Rasdorf, W. and Herbert, M. "CIMS: A Construction Information Management System," **Proceedings of the Fifth ASCE Conference on Computing in Civil Engineering**, Alexandria, Virginia, March, 1988, pp. 33-45.

Rayburn, L., "Productivity Database and Job Cost Control Using Microcomputers," **Journal of Construction Engineering and Management**, ASCE, Vol., 115, No., 4, December, 1989, pp. 585-601.

Rehak, D.R., "Robotic Applications in Construction: A White Paper," in "**New Research Directions in Computerized Applications to Construction Engineering and Management Studies**," Ibbs, W. (ed.), the University of Illinois, Department of Civil Engineering, Construction Research Series No. 19, July, 1985, pp. 108-112.

Reynolds, G. "Information System for Managers," West Publishing Company, 1982.

Rezgui, A. and Tepedelenlioglu, N. "The Effect of the Slope of the Activation Function on the Back Propagation Algorithm," **Proceedings of the International Joint Conference on Neural Networks**, Vol. 1, Washington D.C., Jan. 1990, pp. I.707-I.710.

Rigler, A., Irvine, J. and Vogel, T. "Rescalling of Variables in Back Propagation Learning," **Neural Networks**, Pergamon Press, U.S.A., Vol. 4, No. 2, 1991, pp. 225-229.

Ringwald, R.C. "Bid Mark-up Calculation by Crew-Day Method," **Journal of the Construction Division**, ASCE, Vol. 108, No.CO4, Dec. 1982.

Riggs, L. "Project Control Techniques," **Proceeding of the International Symposium on Building Economics and Construction Management**, CIB, Sydney, Australia, March, 1990, pp. 11-25.

Riggs, L. "Job Site Project Control Using Microcomputers," **Proceedings of the ASCE 5th Conference**, 1988, 450-460.

Rosenblatt, F., "Principles of Neorodynamics," New York: Sparatan Books, 1959.

Rosenblatt, F., "Principles of Neorodynamics: Perceptrons and the Theory of Brain Mechanisms," Washington D.C., 1961, Sparatan Books.

Rosenshine, M. "Bidding Models: Resolution of a Controversy," **Journal of the Construction Division**, ASCE, Vol. 98, No. CO1, 1972, pp. 143-148.

Rumelhart, D.E. "Parallel Distributed Processing," Plenary Session, **IEEE International Conference on Neural Networks**, July 1988, San Diego, CA.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J., "Learning Internal representations by Error Propagation," In D.E., Rumelhart and J.L. McClelland (eds.) **"Parallel Distributed Processing: Explorations in the Microstructures of Cognition,"** Vol. 1: Foundation. MIT press, 1986, Cambridge, MA.

Runeson, G., "Incorporation of Market Conditions into Tendering Models," **Proceeding of the International Symposium on Building Economics And Construction Management**, CIB, Sydney, Australia, Vol. 6, 1990, pp. 393-404.

Russell, J. "Model for Owner Prequalification of Contractors," **Journal of management in Engineering**, ASCE, Vol.6, No. 1, 1990, pp. 59-75.

Russell, A. "REPCON: An Innovative Construction Management System," **Proceeding of the International Symposium on Building Economics And Construction Management**, CIB, Sydney, Australia, March, 1990, pp. 405-416.

Russell, A. and Caselton, W. "Extensions to Linear Scheduling Optimization," **Journal of Construction Engineering and Management**, ASCE, Vol. 114, No. 1, 1988, pp. 36-52.

Samad, T. "Backpropagation Improvement Based on Heuristic Arguments," **Proceedings of the International Joint Conference on Neural Networks,** Washington D.C., Vol. 1, Jan. 1990, pp. I.565-I.568.

Sey, Y., and Dikbas, A., "A Study on Factors Affecting Tender Price of Contractors," **Proceeding of the International Symposium on Building Economics And Construction Management**, CIB, Sydney, Australia, March, Vol. 6, 1990, pp. 451-464.

Scalero, R. and Tepedelenlioglu, N. "A Fast Training Algorithm for Neural Networks," **Proceedings of the International Joint Conference on Neural Networks,** Vol. 1, Washington D.C., Jan. 1990, pp. I.715-I.718.

Sejnowski, T.J., "Higher Order Boltzmann Machines, " **American Institute of Physics Conference Proceedings No. 151: Neural Networks for Computing,** Snowbird, Utah, 1986, pp. 398-403.

Sejnowski, T.J., and Rosenberg, C.R., "Parallel Networks that Learn to Pronounce English Text," **Complex Systems,** 1:145-68, 1987.

Seydel, J. and Olson, D. "Bids Considering Multiple Criteria," **Journal of Construction Engineering and Management**, ASCE, Vol. 116, No. 4, 609-623, Dec. 1990.

Shafer, S. "Estimate and Project Risk Analysis Approaches," **AACE Transactions,** 1991, pp. K.5.1-K.5.5.

Shaw-Taylor, J. and Anthony, M. "Sample Sizes for Multiple-Output Threshold Networks," **Computation in Neural Networks**, Vol.2, No. 1, IOP Publishing Ltd., U.K., 1991, pp.107-117.

Sietsma, J. and Dow, R. "Creating Artificial Neural Networks that Generalize," **Neural Networks**, Vol. 4, No. 1, Pergamon Press, 1991, pp. 67-79.

Silva, F.M. and Almedia, L.B. "Acceleration Techniques for the Back-Propagation Algorithm," Lecture Notes in Computer Science, 412, 1990, pp. 110-119.

Skitmore, M. Contract Bidding in Construction. Longman Group UK Limited, 1989.

Sontag, E., and Sussmann, H. "Back Propagation Separates Where Perceptrons Do," **Neural Networks**, Pergamon Press, U.S.A., Vol. 4, No. 2, 1991, pp. 243-249.

Stein, R.M. "Real Artificial Life," **BYTE**, Vol. 16, No. 1, Jan. 1991, pp. 289-298.

Stark, R., and Mayer, R., "Quantitative Construction Management-Uses of Linear Optimization," John Wiley & Sons, 1983.

Stephenson, P. "A System Prototype for Integrating Estimating Data and Planning Information in the Construction Industry," **Proceedings of the 7th ISARC**, Vol. 1, Bristol, England, June, 1990, pp. 334-342.

Stradal, O. and Cacha, J. "Time Space Scheduling Method," **Journal of the Construction Division**, ASCE, Vol. 108, No. CO3, Sept. 1982, pp. 445-457.

Stubbs, D., "Products to Stimulate and Simulate NeuroComputing Rap," **AI Expert,** June, 1990, pp. 61-63.

Suckarieh, G., "Construction Management Control with Microcomputers," **Journal of Construction Engineering and Management**, ASCE, Vol., 110, No., 1, March, 1984, pp., 72-78.

Szu, H. and Hartley, R. "Fast Simulated Annealing," **Physics Letters**, 1222 (3, 4), 1987, pp. 157-162.

Tarranza, N., and Carmichael, D. "An Investigation of Competitive Bidding Models," **Proceedings of The First East Asian Conference on Structural Engineering and Construction**, In "Structural Engineering and Construction," Kanok-Nukulchai et al. (eds), Vol. 1, Asian Institute of Technology, Bangkok, 1986, pp. 390-400.

Tavakoli, A. and Riachi R. "CPM Use in ENR Top 400 Contractors," **Journal of Management in Engineering**, ASCE, Vol. 6, No. 3, July, 1990, pp. 282-295.

Tavakoli, A., and Utomo, J. "Bid Markup Assistant: An Expert System," **Cost Engineering Journal**, Vol. 31, No. 6, 28-33, June 1989.

Touretzky, D., and Pomerleau, D. "What's Hidden in the Hidden Layers ?," **BYTE**, August 1989, pp. 227-233.

Tveter, D. "Better Speed Through Integers," **AI Expert**, Nov. 1990, pp. 40-46.

Tveter, D. "Getting a Fast Break with Backpropagation," **AI Expert**, July 1991, pp.36-43.

US Department of Energy (DOE) "Cost/Schedule System Criteria for Contract Performance Measurement- Data Analysis Guide," March, 1986.

Vanluchene, R. and Sun, R. "Neural Networks in Structural Engineering," **Microcomputers in Civil Engineering**, Vol. 5, No. 3, Elsevier Science Publishing Co., NY, 1990, pp. 207-215.

Waibel, A. and Hampshire, J. "Building Blocks for Speech," **BYTE**, Aug. 1989, pp.235-242.

Wassermann, P.D. "Neural Computing: Theory and Practice," Van Nostrand Reinhold, NY, 1989.

Wassermann, P.D., "Combined Backpropagation/Cauchy Machine," **Neural Networks: Abstracts of the First INNS Meeting**, Boston, 1988, Vol. 1, p. 556, Elmsford, NY: Pergamon Press.

Weir, M.K. "A Method for Self-Determination of Adaptive Learning Rates in Back Propagation," **Neural Networks**, Vol. 4, No. 3, 1991, Pergamon Press, pp. 371-379.

Werbos, P. "Backpropagation: Past and Future," **Proceedings of the IEEE Second Annual International Conference on Neural Networks**, June 1988, San Diego, CA, pp. I.343-I.353.

White, H. "Neural-Network Learning and Statistics," **AI Expert**, Dec. 1989, pp. 48-52.

Widrow, B., and Hoff, M., "Adaptive Switching Circuits," 1960 IRE Wescon Convention Record, 1960, New York: Institute of Radio Engineers.

Yeo, K. "Risks, Classification of Estimates, and Contingency Management," **Journal of Management in Engineering**, ASCE, Vol. 6, No.4, Oct. 1990, pp. 458-470.

Younker, D. "Project Cost Controls: Estimate Preparation Through Claim Defense," **Cost Engineering**, Vol. 32, No. 12, Dec. 1990, pp. 13-18.

Zozaya-Gorostiza, C. "A Knowledge-Based Planning for Construction Projects," Ph.D. Thesis, Department of Civil Engineering, Carnegie Mellon University, 1988.

APPENDICES

APPENDIX I: Questionnaire survey on bidding in construction.

a) English questionnaire

SURVEY ON BIDDING IN CONSTRUCTION

All responses will remain **FULLY CONFIDENTIAL** and will be used for educational and research purposes only. Please respond by putting a check mark (✓) next to your selection.

COMPANY: ______________________________ (Optional)

TITLE OR POSITION OF RESPONDANT: ______________________

PART 1: ABOUT YOUR FIRM

TYPE OF CONTRACTOR:

___(1) GENERAL or ___(2) SUB ________________ (Specify Trade)

SPECIALITY:
___(1) Buildings.
___(2) Engineering (Highways, Heavy Civil, etc.).
___(3) Industrial Facilities (Power Plants, Refineries,).

ANNUAL SALES ($ MILLIONS):

___(1) Under 1 ___(2) 1-5 ___(3) 5-10
___(4) 10-20 ___(5) 20-50 ___(6) 50-100
___(7) 100-200 ___(8) 200-500 ___(9) Over 500

VALUE OF CONSTRUCTION EQUIPMENT OWNED (% OF FIRM ASSETS):

___(1) None ___(2) Less than 25% ___(3) 25%-50%
___(4) 50%-75% ___(5) 75%-100%

PERCENTAGE OF WORK OBTAINED THROUGH COMPETITIVE BIDDING:

___(1) NONE ___(2) LESS THAN 25% ___(3) 25%-50%
___(4) 50%-75% ___(5) 75%-100%

AVERAGE JOB SIZE ($ MILLIONS):

___(1) Under 0.1 ___(2) 0.1-0.5 ___(3) 0.5-1
___(4) 1-5 ___(5) 5-10 ___(6) 10-50
___(7) 50-100 ___(8) 100-500 ___(9) Over 500

PERCENTAGE OF WORK SUBCONTRACTED ON AVERAGE JOB:

___(1) None ___(2) Less than 25% ___(3) 25%-50%
___(4) 50%-75% ___(5) 75%-100%

PERCENTAGE OF CONSTRUCTION EQUIPMENT LEASED ON AVERAGE JOB:

___(1) NONE ___(2) LESS THAN 25% ___(3) 25%-50%
___(4) 50%-75% ___(5) 75%-100%

PART 2: THE FIRM'S POLICY REGARDING BIDDING DECISION-MAKING PROCESS.

A. HOW DIRECT COSTS OF (LABOUR, EQUIPMENT, MATERIAL & SUBS) ARE ESTIMATED?

These are the direct costs that can be attributed to specific items of work.

___(1) Estimate the Labour and use a (%) for Equip. & Material.
___(2) Detailed Estimation.
___(3) ONLY Rough Estimate (e.g., Parametric Cost Estim. Methods).
___(4) (other)________________________.

B. HOW PROJECT OVERHEAD COSTS ARE ESTIMATED ?

These are project related costs that can not be attributed to specific items of work, such as (permits, temporary facilities,).

___ (1) Detailed Estimation.
___ (2) Aa a Percentage of (DIR. COST) Ranging from: ___% tO:___%.
___ (3) NOT Estimated at all. ___(4) (other)________________________.

C. HOW GENERAL OVERHEAD COSTS ARE ESTIMATED ?

These are costs that can not be related to a particular project (e.g., general office expenses).

___(1) Detailed Estimation.
___(2) As a Percentage of (DIR. COST) Ranging from: __% tO:__%.
___(3) As a Percentage of (PROJ. OVRHD) Ranging from: __% tO:__%.
___(4) As a Percentage of (DIR.COST + PROJ.OVRHD) from: __% tO:__%.
___(5) NOT Estimated at all. ___(6) (other)________________________.

D. HOW PERCENT MARKUP IS ESTIMATED ?

MARKUP covers mainly the profit.

___(1) Using a Statistical Bidding Strategy Model.
___(2) As a Percentage of (DIR.COST) Ranging from: __% tO:__%.
___(3) As a Percentage of (DIR.COST +PROJ.OVRHD) FROM: __% TO:__%.
___(4) As a Percentage of (DIR.COST +PROJ.OVRHD + GEN.OVRHD) from: __% tO:__%. ___(5) (other)________________________.

E. DOES THE ESTIMATED PERCENT MARKUP COVER:

Contingency ? ___Yes, ___No.
Inaccuracy in Cost Estimation ? ___Yes, ___No.
Other ________________________(specify).

F. FACTORS AFFECTING % MARKUP ESTIMATION:

(PLEASE, assign relative weights to indicate the importance of the given factors, as shown in the examples).

FACTOR	Weights	Example 1	Example 2
-Need for Work........		0.2	0.3
-Market Conditions....		0.2	0.3
-Job Uncertainty......		0.2	0.2
-Job Complexity.......		0.2	0.1
-Company Capabilities.		0.2	0.1
-________________(OTHER)		0	0
-________________(OTHER)		0	0
Total =	1.0	1.0	1.0

G. ANY ADDITIONAL COMMENTS REGARDING THE FIRM BIDDING AND MARKUP POLICY?

____

PART 3: CASES FROM PAST PROJECTS.

PLEASE, use this sheet for **ONE** project. It is appreciated if you use copies of this sheet to include as many projects as possible.

A. GENERAL INFORMATION

PROJECT: ______________________ (optional)

TYPE: ______________________ (Specify)

OWNER : ___PUBLIC, ___PRIV.
CONTRACT: ___L.S., ___U.P., ___OTHER

LOCATION: ______________________
EXECUTION START DATE: __________
CONTRACT DURATION: ____months

TOTAL BID PRICE: $__________

B. ASSESSMENT OF JOB UNCERTAINTY

DUE TO:	H				L
-Site Conditions....	5	4	3	2	1
-Owner Attitude.....	5	4	3	2	1
-Project Location...	5	4	3	2	1
-Safety hazard......	5	4	3	2	1
-Inaccuracy in Cost Estimation....	5	4	3	2	1
-Work Sensitivity to Weather.........	5	4	3	2	1
-(other) __________	5	4	3	2	1
OVERALL RANK OF UNCERTAINTY	5	4	3	2	1

C. ASSESSMENT OF JOB COMPLEXITY

	H				L
-Technology Needed...	5	4	3	2	1
-Resources Needed....	5	4	3	2	1
-Job Size...........	5	4	3	2	1
-Quality of Design & Drawings..........	5	4	3	2	1
-Staking of Trades...	5	4	3	2	1
-% Subcontracted.....	5	4	3	2	1
-Rigidity in following Specs..........	5	4	3	2	1
-(other) __________	5	4	3	2	1
OVERALL RANK OF COMPLEXITY	5	4	3	2	1

D. ASSESSMENT OF THE MARKET

	H				L
-Inflation & its Rate of Change.....	5	4	3	2	1
-Escalation & its Rate of Change.....	5	4	3	2	1
-Competition.........	5	4	3	2	1
(No. of competitors): ____(specify)					
-Economy Growth.....	5	4	3	2	1
-Resources Availabl.	5	4	3	2	1
-(other) __________	5	4	3	2	1
OVERALL RANK OF MARKET	5	4	3	2	1

E. ASSESSMENT OF FIRM ABILITY

	H				L
-Similar Experience..	5	4	3	2	1
-Mgmt. & Supervisn...	5	4	3	2	1
-Confidence in Work Force..........	5	4	3	2	1
-Financial Capabilty.	5	4	3	2	1
-(other) __________	5	4	3	2	1
OVERALL RANK OF ABILITY	5	4	3	2	1

F. ASSESSMENT OF NEED

	H				L
-Need for Work.....	5	4	3	2	1
-(other) __________	5	4	3	2	1

IN THIS JOB, YOUR MARKUP IS: _____%

FINAL OUTCOMES: ___WIN, or ___LOOSE by a Diff. of $____________.

-Project potential for change orders is: ___High, ___Med., ___Low.
-Project potential for claims is: ___High, ___Med., ___Low.
-If the project has already been constructed, actual duration is: _____(m).
-Project profitability is: ___High, ___Med., ___Low, ___Loss.

THANK YOU FOR YOUR COOPERATION

b) French questionnaire

ENQUÊTE SUR LES STRATÉGIES DE SOUMISSION

Vos réponses demeureront ENTIÈREMENT CONFIDENTIELLES et ne serviront qu' à des fins de recherche. Veuillez cocher (✓) la réponse qui vous convient le mieux.

NOM DE L'ENTREPRISE: ________________________________(facultatif)

TITRE OU FONCTION DU RÉPONDANT: ____________________________

1ière PARTIE: RENSEIGNEMENTS GÉNÉRAUX

TYPE D'ENTREPRENEUR:

___(1) GÉNÉRAL ou ___(2) SOUS-TRAITANT ____________________ (Indiquer)

SPÉCIALITÉ:

___(1) Bâtiments.
___(2) Génie (Routes, Civil,...).
___(3) Industriel (Centrales électriques, Raffineries,..).

CHIFFRE D'AFFAIRES ($ MILLIONS):

___(1) Moins de 1	___(2) 1-5	___(3) 5-10
___(4) 10-20	___(5) 20-50	___(6) 50-100
___(7) 100-200	___(8) 200-500	___(9) Plus de 500

VALEUR DE L'ÉQUIPEMENT DE CONSTRUCTION QUE VOUS POSSÉDEZ (% actif de l'entreprise):

___(1) Aucun	___(2) Moins de 25%	___(3) 25%-50%
___(4) 50%-75%	___(5) 75%-100%	

POURCENTAGE DES CONTRATS OBTENUS PAR SOUMISSION (COMPÉTITION):

___(1) Aucun	___(2) Moins de 25%	___(3) 25%-50%
___(4) 50%-75%	___(5) 75%-100%	

VALEUR MOYENNE DES CONTRATS ($ MILLIONS):

___(1) Moins de 0.1	___(2) 0.1-0.5	___(3) 0.5-1
___(4) 1-5	___(5) 5-10	___(6) 10-50
___(7) 50-100	___(8) 100-500	___(9) Plus de 500

POURCENTAGE D'UN CONTRAT TYPIQUEMENT MIS EN SOUS-TRAITANCE:

___(1) Aucun	___(2) Moins de 25%	___(3) 25%-50%
___(4) 50%-75%	___(5) 75%-100%	

POURCENTAGE DE L'ÉQUIPEMENT DE CONSTRUCTION QUE VOUS LOUEZ:

___(1) Aucun	___(2) Moins de 25%	___(3) 25%-50%
___(4) 50%-75%	___(5) 75%-100%	

2ième PARTIE: POLITIQUE DE L'ENTREPRISE FACE À LA PRÉPARATION DES SOUMISSIONS

A. COMMENT ESTIMEZ-VOUS LES COUTS DIRECTS ?

C-À-D, Les coûts associés à une unité de travail du projet.

___(1) En estimant la main d'oeuvre et en utilisant un pourcentage pour l'équipement et le matériel.
___(2) En faisant un estimé detaillé complet.
___(3) En faisant un estimé rapide (ex: estimé paramétrique).
___(4) (autre)________________.

B. COMMENT ESTIMEZ-VOUS LES FRAIS GÉNÉRAUX D'UN PROJET ?

C-À-D, Les coûts d'un projet qui ne peuvent être associés à une unité de travail, tels que les permis, bureaux de chantier,..

___(1) En faisant un estimé detaillé.
___(2) À partir d'un pourcentage des (COÛTS DIR.),variant de:__% à:__%.
___(3) Ces coûts ne sont pas estimés. ___(4) (autre)________________.

C. COMMENT ESTIMEZ-VOUS LES FRAIS GÉNÉRAUX DE L'ENTREPRISE ?

C-À-D, Les coûts qui ne peuvent être associés à un projet en particlier, tels que frais d'administration, fournitures, secrétariat,.

___(1) En faisant un estimé detaillé.
___(2) À partir d'un pourcentage des (COÛTS DIR.), de: __% à:__%.
___(3) À partir d'un pourcentage des (F.G. du PROJET), de: __% à:__%.
___(4) À partir d'un pourcentage des (COÛTS DIR. + F.G. du PROJET), de: __% à:__%.
___(5) Ces coûts ne sont pas estimés. ___(6) (autre)________________.

D. COMMENT ESTIMEZ-VOUS LA MARGE DE PROFIT ?

___(1) En utilisant un outil statistique.
___(2) À partir d'un pourcentage des (COÛTS DIR.), de: __% à:__%.
___(3) À partir d'un pourcentage des (COÛTS DIR.+ F.G. du PROJET), de: __% à:__%.
___(4) À partir d'un pourcentage des (COÛTS DIR.+ F.G. du PROJET + F.G. de L'ENTREPRISE), de: __% à:__%.
___(5) (autre)________________.

E. EST-CE-QUE LA MARGE DE PROFIT COMPREND:

Fond de contingence ? ___Oui, ___Non.
Possibilité d'erreurs dans l'estimé de coûts ? ___Oui, ___Non.

F. FACTEURS IMPORTANTS FACE À L'ETABLISSEMENT DE LA MARGE DE PROFIT:

(S.V.P., indiquer l'importance relative des facteurs telle que demontrée dans les exemples).

FACTEUR	IMP.REL.	Exemple 1	Exemple 2
-Besoin de travail........		0.2	0.3
-Conditions du marché......		0.2	0.3
-Incertitude associée au projet...........		0.2	0.2
-Complexité du projet......		0.2	0.1
-Expertise de l'entreprise		0.2	0.1
-________ (autre)		0	0
Total =	1.0	1.0	1.0

G. COMMENTAIRES ADDITIONNELS SUR LA POLITIQUE DE L'ENTREPRISE ?

____

3 ième PARTIE: EXEMPLES DE PROJETS SOUMISSIONNÉS.

S.V.P., utiliser cette feuille pour UN projet. Si possible, utiliser des copies de cette feuille pour inclure d'autres exemples.

A. RENSEIGNEMENTS GÉNÉRAUX

PROJET: ____________________ (facultatif)

TYPE: ____________________ (indiquer)

CLIENT : ___PUBLIC, ___PRIVÉ.
CONTRAT: ___À forfait,
___Prix unitaire, ___autre
ENDROIT: ____________________
DATE DE DÉBUT DES TRAVAUX: ____________
DURÉE DU PROJET: _____(m)

MONTANT DE LA SOUMISSION: $____________

B. ÉVALUATION DE L'INCERTITUDE

ATTRIBUABLE À:	H.				B
-Conditions du site..	5	4	3	2	1
-Relations avec le client...........	5	4	3	2	1
-Emplacement du projet.............	5	4	3	2	1
-Sécurité au chantier	5	4	3	2	1
-Possibilité d'erreurs dans l'estimé.......	5	4	3	2	1
-Influence des intempéries.........	5	4	3	2	1
PONDÉRATION DE L'INCERTITUDE	5	4	3	2	1

C. ÉVALUATION DE LA COMPLEXITÉ DU PROJET

	H.				B
-Technologie requise.	5	4	3	2	1
-Ressources requises.	5	4	3	2	1
-Taille du projet....	5	4	3	2	1
-Qualité du design et des dessins & devis.	5	4	3	2	1
-Congestion des	5	4	3	2	1
métiers.............	5	4	3	2	1
-% soustraitance.....					
-Spécificité des	5	4	3	2	1
devis...............	5	4	3	2	1
PONDÉRATION DE LA COMPLEXITÉ:	5	4	3	2	1

D. ÉVALUATION DU MARCHÉ

	H.				B
-Inflation..........	5	4	3	2	1
-Hausse des coûts...	5	4	3	2	1
-Compétition........	5	4	3	2	1
(# de compétiteurs): ____(indiquer)					
-Croissance de l'économie.........	5	4	3	2	1
-Ressources disponibles........	5	4	3	2	1
PONDÉRATION DU MARCHÉ:	5	4	3	2	1

E. ÉVALUATION DE L'EXPERTISE

	H.				B
-Expérience similaire.	5	4	3	2	1
-Gérance & supervision	5	4	3	2	1
-Confiance dans le personnel............	5	4	3	2	1
-Capacités financières..........	5	4	3	2	1
PONDÉRATION DE L'EXPERTISE	5	4	3	2	1

F. ÉVALUATION DU BESOIN DE TRAVAIL

	H.				B
-Besoin de travail.	5	4	3	2	1

LA MARGE DE PROFIT EST DE: ______%

RÉSULTAT FINAL: ___GAGNÉ, ou ___PERDU, par une différence:$__________.
sur ce projet:
-Le potentiel pour les changements d'ordre est:___Haut, ___Moyen, ___Bas.
-Le potentiel pour les litiges est:___Haut, ___Moyen, ___Bas.
-Si le projet a été construit, la durée réelle a été de:_____(m).
-La rentabilité du projet est: ___Haute, ___Moyenne,___Basse, ___Perte.

MERCI

APPENDIX II: Row data of past projects used for training and testing of markup networks.

CASE A: Data used for training the single-network model.

.... KEY for the data fields in an example project ...

Ex. #; **INPUTS (total = 30):**
Owner; Contract; Total bid; Contr. dur.; need for work.
Contractor size; Markup defin.; Markup components.
Site cond.; Owner attit.; Loc.; Hazz.; Inaccur.; Weather.
Tech.; Res.; Size; Qual.; Stack.; % sub.; Spcs.
Infl.; Escal.; Compet.; # of compet.; Econ. cond.; Res. avail.
Similar exper.; Mgmt cond.; Conf. in work force; Fin. capab.
DESIRED OUTPUTS (total = 7):
% Markup; Win(0); Diff.($)
C.O.s; claims; Dur. Exten.; Act. prof.

.....LIST OF DATA.....

1 INPUTS:
1 1 1.8 10 5
0 2 2
1 3 2 2 2 3
3 3 3 2 4 3 4
3 2 5 3 4
5 5 5 5
OUTPUTS:
5 0 .12
1 2 1 3

2 INPUTS:
2 1 4.92 11 4
1 0 3
4 2 3 1 2 3
2 2 3 4 3 1 2
1 1 1 3 3
5 4 3 5
OUTPUTS:
5 1 0
3 2 1 2

3 INPUTS:
1 1 3.6 16 4
1 0 3
3 4 5 5 4 3
3 3 2 2 2 5 5
1 2 4 1 3
2 3 4 4
OUTPUTS:
6 1 0
2 3 1 2

4 INPUTS:
2 3 23.65 20 3
1 2 0
3 5 3 2 2 2
4 4 4 4 3 5 3
3 3 3 2 3
2 3 3 3
OUTPUTS:
6 1 0
1 2 1 2

5 INPUTS:
2 1 2 1 4
1 0 0
2 2 4 1 1 3
2 3 3 2 2 1 3
3 4 5 3 3
5 5 5 4
OUTPUTS:
5 1 0
1 3 1 1

6 INPUTS:
2 1 10.5 14 3
1 0 4
3 5 1 1 2 1
2 2 3 2 2 4 5
4 2 4 2 2
5 3 5 5
OUTPUTS:
.1 1 2
1 3 1.286 3

7 INPUTS:
1 1 2.5 9 4
1 2 2
2 4 4 3 1 2
3 4 4 5 4 5 5

1 1 5 3 3
4 4 4 5
OUTPUTS:
3.5 1 .01
3 3 1 2

8 INPUTS:
2 1 2 8 4
1 0 1
2 1 1 1 2 2
2 2 2 1 2 2 2

1 1 3 2 3
4 4 4 4
OUTPUTS:
2.5 1 0
3 3 1 3

9 INPUTS:
2 1 33.5 18 4
1 0 1
2 2 1 1 1 4
1 4 2 3 4 3 1

4 4 4 2 5
5 5 3 5
OUTPUTS:
3 1 3
3 3 1 2

10 INPUTS:
2 1 12.3 11 4
1 0 1
1 3 1 1 2 4
1 3 1 2 3 4 3

3 2 5 4 2
5 5 3 5
OUTPUTS:
3 1 .5
2 3 1 2

11 INPUTS:
1 1 10 20 3
2 2 1
5 3 5 1 2 5
1 3 3 3 2 3 2

4 3 2 2 3
5 4 4 4
OUTPUTS:
10 1 5
3 3 1 1

12 INPUTS:
1 1 7.5 7 5
2 2 1
1 1 1 1 1 2
2 2 2 3 2 1 1

1 1 4 1 3
4 4 4 5
OUTPUTS:
1.5 1 0
3 3 1 3

13 INPUTS:
2 1 4.2 10 5
1 1 2
2 2 1 1 4 4
4 2 4 2 2 3 2

1 1 1 1 2
4 5 5 5
OUTPUTS:
8.2 1 0
2 3 1 2

14 INPUTS:
2 2 2 13 3
1 2 2
3 1 1 1 3 5
2 3 4 1 3 2 3

1 1 3 1 3
3 4 5 5
OUTPUTS:
2.3 1 0
2 3 1 2

15 INPUTS:
1 1 3.2 10 4
1 2 1
2 1 1 1 2 1
2 2 3 2 2 4 4

3 3 4 2 3
5 4 5 4
OUTPUTS:
3 1 .11
2 3 1 2

16 INPUTS:
2 1 2 12 4
1 2 1
3 1 1 1 1 3
2 4 4 3 2 2 2

2 2 4 3 3
4 4 4 3
OUTPUTS:
6.5 1 .5
2 3 1 2

17 INPUTS:
1 1 21 22 3
1 0 0
1 3 3 3 2 3
4 4 4 5 3 4 5

3 3 3 3 3
5 5 4 5
OUTPUTS:
2 1 11
1 3 1 1

18 INPUTS:
1 1 13.5 12 3
1 0 4
2 4 3 3 4 5
4 4 4 3 4 5 5

3 3 4 2 3
2 4 3 4
OUTPUTS:
3 1 2.25
2 3 1 3

19 INPUTS:
2 1 2 12 3
1 0 4
2 5 4 2 4 5
2 2 5 5 4 5 4

3 3 5 3 3
2 4 4 4
OUTPUTS:
3 1 .08
3 3 1.166667 2

20 INPUTS:
2 2 14.5 15 2
1 2 0
5 1 3 1 1 1
3 1 5 3 3 3 4

1 1 1 1 1
5 5 5 5
OUTPUTS:
7 1 0
2 3 1 1

21 INPUTS:
1 1 5 14 4
1 1 2
4 3 3 1 1 4
4 2 4 1 3 5 3

2 2 3 2 3
5 5 4 5
OUTPUTS:
5 1 0
2 3 1 1

22 INPUTS:
1 1 2.7 12 3
1 1 2
4 2 1 2 3 2
5 3 2 2 2 2 3

2 3 4 4 3
5 4 4 2
OUTPUTS:
4 1 .6
3 3 1 2

23 INPUTS:
1 1 2 7 4
0 2 0
4 5 5 5 1 2
3 3 2 3 3 5 5

3 2 3 3 3
3 4 5 5
OUTPUTS:
7 1 .05
3 2 1 3

24 INPUTS:
2 1 1.8 6 5
1 2 1
1 1 1 1 1 1
1 3 2 2 3 4 4

2 1 5 3 2
5 4 4 4
OUTPUTS:
0 0 .8
3 3 1 3

25 INPUTS:
2 1 2.2 4 5
1 2 1
3 1 1 1 1 4
1 2 1 1 1 2 2

1 2 5 3 3
4 4 4 4
OUTPUTS:
2 1 0
2 2 1 3

26 INPUTS:
2 1 .36 5 3
0 0 0
1 3 2 2 2 3
4 4 2 4 3 4 4

3 4 4 4 2
5 4 5 5
OUTPUTS:
9 1 0
2 3 1 2

27 INPUTS:
1 1 3.6 9 2
0 0 0
2 4 2 2 3 2
4 4 5 4 4 4 4

5 5 4 4 4
3 3 4 2
OUTPUTS:
13 0 2
1 2 1 1

28 INPUTS:
2 1 8.6 6 4
2 1 2
3 1 1 2 2 3
1 5 2 3 5 5 2

1 1 4 3 1
1 1 1 1
OUTPUTS:
2 1 0
2 3 1 3

29 INPUTS:
1 1 43.7 21 3
2 1 3
2 3 2 2 2 3
3 3 5 4 3 4 2

1 1 2 2 3
4 5 3 5
OUTPUTS:
4 1 0
2 3 1 2

30 INPUTS:
2 3 7.5 14 5
1 2 0
4 3 1 1 1 2
1 3 2 3 1 4 2

1 1 3 3 3
5 5 5 4
OUTPUTS:
7 1 0
3 3 1 2

31 INPUTS:
1 1 12.3 20 3
1 0 0
3 5 5 3 2 4
4 4 3 2 3 4 4

3 3 5 3 4
5 5 2 5
OUTPUTS:
5 1 .8
3 3 1 2

32 INPUTS:
1 1 5.4 16 4
1 0 0
1 2 3 2 3 2
3 2 2 3 3 4 4

2 2 5 2 4
4 4 4 4
OUTPUTS:
4 0 .15
3 2 1 2

33 INPUTS:
2 1 10.1 18 4
1 2 1
4 1 2 3 2 4
3 4 4 4 3 4 5

2 2 5 2 3
4 4 4 5
OUTPUTS:
6.5 1 2
2 3 1 2

34 INPUTS:
1 1 15.7 20 4
1 2 1
2 2 3 2 2 4
5 5 4 2 4 4 5

3 2 5 2 3
5 3 4 5
OUTPUTS:
4.5 1 1
1 1 1.25 2

35 INPUTS:
2 1 15.1 16 3
1 2 2
3 3 3 2 2 2
2 3 3 2 3 5 3

3 2 3 3 3
5 5 5 5
OUTPUTS:
5 1 1
2 2 1 1

36 INPUTS:
2 1 2 17 4
2 2 0
3 4 3 2 2 2
4 4 3 3 3 4 5

2 3 5 4 3
4 4 5 4
OUTPUTS:
3 0 0
3 1 1 2

37 INPUTS:
1 2 1.7 5 5
1 1 0
1 2 1 1 3 4
2 3 1 1 1 2 4

1 1 5 3 3
5 4 4 5
OUTPUTS:
3 1 0
2 3 1 3

38 INPUTS:
2 1 5.3 6 4
1 1 0
1 4 4 2 4 2
4 5 3 5 3 4 2

3 4 1 3 3
4 4 2 4
OUTPUTS:
10 1 0
1 1 1 1

39 INPUTS:
2 3 18.5 10 4
1 2 0
4 2 3 3 2 4
3 3 4 2 3 4 2

2 2 4 2 2
2 3 2 4
OUTPUTS:
6 1 0
1 2 1 2

40 INPUTS:
2 3 4.9 11 4
1 2 0
2 1 3 2 2 1
2 3 2 4 3 4 3

2 2 3 3 4
1 3 4 4
OUTPUTS:
6 1 0
2 2 1 2

41 INPUTS:
2 1 2.3 6 3
0 1 2
2 4 2 2 2 3
3 4 4 3 3 5 5

2 2 4 2 2
4 4 4 4
OUTPUTS:
5 1 .08
2 2 1 2

42 INPUTS:
1 1 22 24 5
2 2 1
5 1 1 5 1 2
5 4 1 5 1 1 2

1 1 2 5 1
1 5 5 5
OUTPUTS:
7 1 10
2 3 1 1

43 INPUTS:
2 1 21 22 5
2 2 1
3 3 1 1 1 1
1 1 1 2 1 4 5

1 1 4 5 1
5 5 5 5
OUTPUTS:
4 0 2
1 2 1 3

44 INPUTS:
2 3 18.3 18 3
2 1 0
3 5 2 1 2 3
4 4 3 1 3 2 3

2 2 1 2 4
5 5 4 5
OUTPUTS:
5 1 0
1 2 1 2

45 INPUTS:
2 1 5.2 8 2
2 1 0
1 2 2 1 3 1
4 4 2 3 2 4 4

2 2 4 3 4
3 4 3 5
OUTPUTS:
3.5 1 0
2 3 1 2

46 INPUTS:
1 1 2 12 3
0 0 1
4 3 3 2 2 3
1 1 3 4 3 2 4

3 2 4 2 3
4 4 3 4
OUTPUTS:
7.5 1 0
2 3 1.083 2

47 INPUTS:
1 1 2 14 4
0 0 1
5 3 3 4 4 1
2 3 3 2 3 3 3

2 2 4 3 4
2 4 2 5
OUTPUTS:
7.5 1 0
1 2 1.142857 4

48 INPUTS:
1 1 .49 4 4
0 2 1
3 3 5 3 2 5
3 4 3 5 3 1 5

3 3 4 3 3
3 3 2 3
OUTPUTS:
18 0 .2
2 1 1 1

49 INPUTS:
1 1 .61 4 4
0 2 1
2 4 3 3 2 4
3 3 4 3 2 1 5

4 4 3 4 3
5 5 5 5
OUTPUTS:
20 1 .2
2 1 1 2

50 INPUTS:
1 1 11.3 37 5
0 2 1
3 1 1 1 1 1
5 3 4 4 2 5 4

2 2 3 4 1
4 4 3 1
OUTPUTS:
5 1 0
2 3 1 3

51 INPUTS:
1 1 .39 7 3
0 0 1
3 2 1 2 2 4
3 4 4 4 3 5 2

3 3 3 2 3
4 4 4 4
OUTPUTS:
15 1 0
2 2 1 2

52 INPUTS:
1 1 3.5 9 4
0 0 0
4 1 1 1 1 3
2 2 3 2 3 4 3

1 1 2 3 3
2 4 4 5
OUTPUTS:
9 1 .5
2 2 1 1

53 INPUTS:
2 3 3.8 11 3
0 0 0
3 3 2 2 2 1
4 4 4 3 3 3 3

1 1 2 2 2
3 4 3 5
OUTPUTS:
8 1 0
1 2 1 1

54 INPUTS:
2 3 1.4 4 5
1 1 3
2 2 2 2 2 2
2 3 4 4 3 5 3

1 1 3 3 1
4 4 5 5
OUTPUTS:
4 1 0
3 3 1 2

55 INPUTS:
1 1 28.5 16 4
2 2 0
3 3 3 3 2 3
3 3 3 2 3 3 3

3 3 3 3 3
4 4 4 4
OUTPUTS:
4 1 10
2 3 1 2

56 INPUTS:
2 1 14.3 12 4
2 2 0
3 3 3 3 2 3
3 3 3 3 3 3 3

3 3 3 3 3
4 4 4 4
OUTPUTS:
4 1 0
2 3 1 2

57 INPUTS:
1 1 25 30 4
2 0 0
2 3 1 1 2 2
4 3 4 3 3 4 4

4 4 4 4 4
4 4 4 4
OUTPUTS:
5 1 0
1 2 1 1

58 INPUTS:
2 1 2.5 14 3
2 0 0
5 5 2 4 4 4
3 4 2 3 4 3 5

2 2 3 3 2
4 4 4 4
OUTPUTS:
1.5 1 0
1 1 1.142857 3

59 INPUTS:
1 1 6.5 18 1
1 2 2
1 1 1 1 1 5
3 3 3 4 4 4 4

3 3 4 3 3
3 4 5 5
OUTPUTS:
5 1 0
2 3 1 1

60 INPUTS:
2 1 7.5 18 1
1 2 2
3 3 2 2 2 2
4 4 4 3 4 4 3

1 1 3 2 3
3 4 4 5
OUTPUTS:
3 1 0
1 2 1 2

61 INPUTS:
2 1 7.5 12 4
1 0 0
1 1 1 1 3 3
2 4 2 2 2 4 4

2 2 2 3 3
5 5 5 5
OUTPUTS:
4.3 1 0
1 3 1 2

62 INPUTS:
2 1 3.76 12 3
2 2 0
1 4 1 1 1 1
1 1 3 3 1 3 3

1 4 4 2 3
4 4 4 4
OUTPUTS:
4 0 0
3 3 1 3

63 INPUTS:
2 1 2.3 6 4
2 2 0
1 4 3 3 1 1
1 1 2 2 2 4 2

4 1 5 2 4
4 4 4 4
OUTPUTS:
1.5 1 0
2 2 1 2

64 INPUTS:
2 1 5 15 4
1 1 0
1 4 4 4 2 2
3 3 2 2 3 3 2

3 3 4 3 3
3 2 4 4
OUTPUTS:
4 1 0
1 3 1.066667 4

65 INPUTS:
1 1 19 36 5
1 2 0
3 5 5 4 1 5
4 5 5 2 4 1 5

3 3 5 4 5
3 4 5 4
OUTPUTS:
3 1 10
1 1 1 2

CASE B: Data used for testing the single-network model.

.... KEY for the data fields in an example project ...

Ex. #; **INPUTS (total = 30):**

Owner; Contract: Total bid; Contr. dur.; need for work.
Contractor size; Markup defin.; Markup components.
Site cond.; Owner attit.; Loc.; Hazz.; Inaccur.; Weather.
Tech.; Res.; Size; Qual.; Stack.; % sub.; Spcs.
Infl.; Escal.; Compet.; # of compet.; Econ. cond.; Res. avail.
Similar exper.; Mgmt cond.; Conf. in work force; Fin. capab.
DESIRED OUTPUTS (total = 7) - not introduced to the neural network:
% Markup; Win(0); Diff.($)
C.O.s; claims; Dur. Exten.; Act. prof.

---

1 INPUTS:
2 3 5 48 4
1 2 0
5 5 2 5 1 1
5 5 5 1 5 5 5
4 4 5 2 2
5 5 5 5
OUTPUTS:
3 1 0
1 1 1 3

2 INPUTS:
2 3 5 0 2
1 2 0
1 1 3 2 1 3
5 5 5 1 4 4 4
4 3 5 1 3
5 5 5 5
OUTPUTS:
4 1 0
1 1 1 3

3 INPUTS:
2 3 2 10 5
1 2 1
1 1 1 1 2 1
5 4 5 5 3 3 4
3 3 3 3 3
3 4 3 4
OUTPUTS:
5 1 0
2 2 1 3

4 INPUTS:
1 1 19.6 16 4
1 0 0
2 1 3 1 1 2
3 3 5 2 2 3 2
1 2 3 2 4
5 4 5 5
OUTPUTS:
2 1 7
2 3 1.125 2

5 INPUTS:
2 1 7.5 8 5
1 2 0
1 1 1 1 1 3
1 1 3 3 5 5 3
1 1 5 3 5
5 5 5 5
OUTPUTS:
2 0 1.5
3 3 1 3

6 INPUTS:
1 1 7.5 12 5
1 2 0
3 3 3 2 2 3
3 3 3 4 3 3 5
1 1 5 3 5
5 5 3 5
OUTPUTS:
4 0 2
3 3 1 3

7 INPUTS:
2 1 1.3 13 3
1 2 2
4 4 5 5 4 4
4 4 4 5 5 1 5
2 3 5 3 3
4 4 4 4
OUTPUTS:
11 1 .61
2 2 1 2

CASE C: Data used for training the Sub-networks of the Hierarchical models.

Sub-network (1) for assessment of job Uncertainty.

	INPUTS						OUTPUT
Example #	(a)	(b)	(c)	(d)	(e)	(f)	OVERALL RANK
1	1	2	1	1	1	4	2
2	1	3	2	2	2	3	2
3	4	2	3	1	2	3	3
4	3	4	5	5	4	3	4
5	3	3	4	1	3	1	3
6	3	5	3	2	2	2	3
7	2	2	4	1	1	3	3
8	3	5	1	1	2	1	2
9	5	1	3	5	4	4	4
10	2	1	1	1	2	2	2
11	2	2	1	1	1	4	2
12	1	3	1	1	2	4	2
13	2	2	1	1	4	4	3
14	3	3	3	3	3	3	3
15	2	4	3	3	4	5	4
16	5	1	3	1	1	1	2
17	4	3	1	1	2	4	2
18	3	2	1	4	1	3	2
19	1	1	1	1	1	1	1
20	3	1	1	1	1	4	3
21	2	4	2	2	3	2	2
22	2	3	2	2	2	3	2
23	3	5	5	3	2	4	4
24	1	2	3	2	3	2	2
25	4	1	2	3	2	4	3
26	2	2	3	2	2	4	3
27	3	3	3	2	2	2	2
28	3	4	3	2	2	2	3
29	1	2	1	1	3	4	2
30	1	4	4	2	4	2	4
31	4	2	3	3	2	4	3
32	2	1	3	2	2	1	2
33	2	4	2	2	2	3	2
34	5	1	1	5	1	2	5
35	3	3	1	1	1	1	3
36	3	5	2	1	2	3	3
37	1	2	2	1	3	1	2
38	3	3	5	3	2	5	4
39	2	4	3	3	2	4	3
40	3	1	1	1	1	1	2

Sub-network (1) continued.

	INPUTS						OUTPUT
Example #	(a)	(b)	(c)	(d)	(e)	(f)	OVERALL RANK
41	3	2	1	2	2	4	3
42	4	1	1	1	1	3	2
43	3	3	2	2	2	1	2
44	1	5	1	1	2	2	3
45	4	5	2	3	2	2	4
46	3	3	3	3	2	3	3
47	2	3	1	1	2	2	2
48	5	5	2	4	4	4	4
49	1	3	1	1	3	1	2
50	1	3	4	1	1	1	2
51	1	4	3	3	1	1	3
52	2	3	4	2	2	4	3
53	1	1	1	1	2	1	1
54	1	1	1	1	1	3	1
55	3	3	3	2	2	3	3
56	5	3	1	3	3	1	3
57	4	4	5	5	4	4	4
58	1	1	1	1	1	2	1

(a) Uncertainty due to site conditions.
(b) Uncertainty due to owner attitude.
(c) Uncertainty due to project location.
(d) Uncertainty due to safety hazard.
(e) Uncertainty due to inaccuracy in estimate.
(f) Uncertainty due to work sensitivity to weather.

Sub-network (2) for assessment of job Complexity.

Example #	INPUTS (a)	(b)	(c)	(d)	(e)	(f)	(g)	OUTPUT OVERALL RANK
1	3	4	5	2	3	3	5	4
2	2	2	2	1	1	1	2	2
3	3	3	3	2	4	3	4	3
4	2	2	3	4	3	1	2	3
5	3	3	2	2	2	5	5	4
6	1	3	3	2	3	3	5	3
7	4	4	4	4	3	5	3	4
8	2	3	3	2	2	1	3	3
9	2	2	3	2	2	4	5	3
10	1	3	3	3	5	4	4	4
11	2	2	2	1	2	2	2	2
12	1	4	2	3	4	3	1	3
13	1	3	1	2	3	4	3	3
14	3	3	3	3	3	3	3	3
15	3	1	5	3	3	3	4	3
16	2	2	2	2	2	2	4	2
17	2	2	2	4	2	4	3	2
18	5	3	2	2	2	2	3	3
19	1	3	2	2	3	4	4	3
20	1	2	1	1	1	2	2	2
21	4	4	2	4	3	4	4	3
22	4	4	5	4	4	4	4	4
23	1	5	2	3	5	5	2	5
24	3	3	5	4	3	4	2	4
25	4	4	3	2	3	4	4	4
26	3	2	2	3	3	4	4	3
27	3	4	4	4	3	4	5	4
28	5	5	4	2	4	4	5	5
29	2	3	3	2	3	5	3	3
30	4	4	3	3	3	4	5	4
31	2	3	1	1	1	2	4	3
32	4	5	3	5	3	4	2	4
33	3	3	4	2	3	4	2	3
34	2	3	2	4	3	4	3	3
35	3	4	4	3	3	5	5	3
36	5	4	1	5	1	1	2	5
37	1	1	1	2	1	4	5	2
38	4	4	3	1	3	2	3	3
39	4	4	2	3	2	4	4	4
40	3	4	3	5	3	1	5	4
41	3	3	4	3	2	1	5	3
42	5	3	4	4	2	5	4	4

Sub-network (2) continued.

Example #	(a)	(b)	(c)	(d)	(e)	(f)	(g)	OVERALL RANK
	INPUTS							OUTPUT
43	3	4	4	4	5	2	3	3
44	2	2	3	2	3	4	3	3
45	4	4	4	3	3	3	3	3
46	2	2	2	4	3	2	5	4
47	2	2	3	5	3	1	5	4
48	2	3	4	4	3	5	3	4
49	3	3	3	2	3	3	3	3
50	4	3	4	3	3	4	4	4
51	3	4	2	3	4	3	5	4
52	4	2	5	3	3	4	4	4
53	1	1	3	3	1	3	3	2
54	1	1	2	2	2	4	2	2
55	5	4	5	5	3	3	4	4
56	1	1	3	3	5	5	3	3
57	3	3	3	4	3	3	5	3
58	5	2	3	2	2	2	2	2
59	4	4	4	5	5	1	5	4
60	2	3	2	4	3	3	3	3
61	4	4	3	3	1	4	4	3

(a) Technology needed.
(b) Resources needed.
(c) Job size.
(d) Quality of design and drawings.
(e) Stacking of trades.
(f) Percent subcontracted.
(g) Rigidity in following specifications.

Sub-network (3) for assessment of Market conditions.

Example #	INPUTS (a)	(b)	(c)	(d)	(e)	OUTPUT OVERALL RANK
1	1	1	4	1	1	2
2	3	2	5	3	4	3
3	1	2	4	1	3	2
4	3	3	3	2	3	3
5	3	4	5	3	3	3
6	4	2	4	2	2	3
7	1	1	3	2	3	2
8	4	4	4	2	5	2
9	3	2	5	4	2	2
10	1	1	4	1	3	2
11	2	2	4	3	3	3
12	2	3	4	4	3	3
13	2	1	5	3	2	2
14	1	2	5	3	3	3
15	3	4	4	4	2	2
16	5	5	4	4	4	1
17	1	1	4	3	1	2
18	1	1	2	2	3	4
19	3	3	5	3	4	3
20	2	2	5	2	4	2
21	2	2	5	2	3	2
22	3	2	5	2	3	3
23	3	2	3	3	3	3
24	2	3	5	4	3	3
25	1	1	5	3	3	2
26	3	4	1	3	3	3
27	2	2	4	2	2	2
28	2	2	3	3	4	3
29	1	1	2	5	1	2
30	1	1	4	5	1	2
31	2	2	4	3	4	3
32	3	3	4	3	3	3
33	4	4	3	4	2	2
34	2	2	3	4	1	3
35	1	1	2	3	3	4
36	1	1	2	2	2	2
37	3	3	5	3	5	3
38	2	3	5	3	4	3
39	3	3	3	3	3	3
40	2	2	3	3	2	3
41	1	4	4	2	3	3
42	4	1	5	2	4	3

Sub-network (3) continued.

Example #	INPUTS (a)	(b)	(c)	(d)	(e)	OUTPUT OVERALL RANK
43	2	3	4	3	3	4
44	1	1	5	3	5	3
45	1	1	3	2	2	2
46	2	3	5	3	3	3
47	1	1	4	2	2	2
48	1	2	5	2	4	2

(a) Inflation rate.
(b) Escalation rate.
(c) Competition.
(d) Economic growth.
(e) Resources available.

Sub-network (4) for assessment of Contractor capabilities.

	INPUTS				OUTPUT
Example #	(a)	(b)	(c)	(d)	OVERALL RANK
1	4	5	5	5	5
2	5	5	5	5	5
3	5	4	3	5	4
4	2	3	4	4	3
5	2	3	3	3	3
6	5	3	5	5	4
7	4	4	4	4	4
8	5	5	3	5	5
9	5	5	3	5	4
10	5	4	4	4	4
11	4	4	4	5	4
12	3	3	3	3	3
13	5	4	4	5	5
14	5	5	4	5	5
15	5	4	4	2	3
16	5	4	4	4	5
17	5	4	5	5	5
18	3	3	4	2	3
19	1	1	1	1	1
20	4	5	3	5	5
21	5	5	2	5	5
22	5	3	4	5	4
23	4	4	5	4	4
24	5	4	4	5	4
25	4	4	2	4	3
26	2	3	2	4	3
27	1	3	4	4	3
28	1	5	5	5	4
29	3	4	3	5	4
30	3	3	2	3	3
31	4	4	3	1	2
32	2	4	4	5	4
33	2	2	4	3	3
34	4	4	5	5	5
35	5	3	3	1	3
36	3	4	3	4	3
37	4	4	4	1	4
38	2	3	3	5	3

(a) Similar experience. **(c)** Confidence in work force.
(b) Management conditions. **(d)** Financial capabilities.

CASE D: Data used for training the Hierarchical models (Global network).

.... KEY for the data fields in an example project ...

Ex. #; **INPUTS (total = 30) - not all have been used:**
Owner; Contract; Total bid; Contr. dur.; need for work.
Contractor size; Markup defn.; Markup components.
Site cond.; Owner attit.; Loc.; Hazz.; Inaccur.; Weather.
Tech.; Res.; Size; Qual.; Stack.; % sub.; Spcs.
Infl.; Escal.; Compet.; # of compet.; Econ. cond.; Res. avail.
Similar exper.; Mgmt cond.; Conf. in work force; Fin. capab.
OVERALL RANKS FOR THE SUB-NETWORKS (total = 4):
Uncertainty; Complexity; Market Cond.; Firm Ability.
DESIRED OUTPUTS (total = 7):
% Markup; Win(0); Diff.($)
C.O.s; claims; Dur. Exten.; Act. prof.

.....LIST OF DATA.....

1 INPUTS:
1 1 1.8 10 5
0 2 2
1 3 2 2 2 3
3 3 3 2 4 3 4
3 2 5 3 4
5 5 5 5
Overall ranks:
2 3 3 5
OUTPUTS:
5 0 .12
1 2 1 3

2 INPUTS:
2 1 4.92 11 4
1 0 3
4 2 3 1 2 3
2 2 3 4 3 1 2
1 1 1 3 3
5 4 3 5
Overall ranks:
3 3 4 4
OUTPUTS:
5 1 0
3 2 1 2

3 INPUTS:
1 1 3.6 16 4
1 0 3
3 4 5 5 4 3
3 3 2 2 2 5 5
1 2 4 1 3
2 3 4 4
Overall ranks:
4 4 2 3
OUTPUTS:
6 1 0
2 3 1 2

4 INPUTS:
2 3 23.65 20 3
1 2 0
3 5 3 2 2 2
4 4 4 4 3 5 3
3 3 3 2 3
2 3 3 3
Overall ranks:
3 4 3 3
OUTPUTS:
6 1 0
1 2 1 2

5 INPUTS:
2 1 2 1 4
1 0 0
2 2 4 1 1 3
2 3 3 2 2 1 3
3 4 5 3 3
5 5 5 4
Overall ranks:
3 3 3 4
OUTPUTS:
5 1 0
1 3 1 1

6 INPUTS:
2 1 10.5 14 3
1 0 4
3 5 1 1 2 1
2 2 3 2 2 4 5
4 2 4 2 2
5 3 5 5
Overall ranks:
2 3 3 4
OUTPUTS:
.1 1 2
1 3 1.285714 3

7 INPUTS:
2 1 2 8 4
1 0 1
2 1 1 1 2 2
2 2 2 1 2 2 2

1 1 3 2 3
4 4 4 4
Overall ranks:
2 2 2 4
OUTPUTS:
2.5 1 0
3 3 1 3

8 INPUTS:
2 1 33.5 18 4
1 0 1
2 2 1 1 1 4
1 4 2 3 4 3 1

4 4 4 2 5
5 5 3 5
Overall ranks:
2 3 2 5
OUTPUTS:
3 1 3
3 3 1 2

9 INPUTS:
2 1 12.3 11 4
1 0 1
1 3 1 1 2 4
1 3 1 2 3 4 3

3 2 5 4 2
5 5 3 5
Overall ranks:
2 3 2 4
OUTPUTS:
3 1 .5
2 3 1 2

10 INPUTS:
2 2 14.5 15 2
1 2 0
5 1 3 1 1 1
3 1 5 3 3 3 4

1 1 1 1 1
5 5 5 5
Overall ranks:
2 3 1 5
OUTPUTS:
7 1 0
2 3 1 1

11 INPUTS:
2 1 2 10 3
1 1 1
3 3 3 3 3 3
3 3 3 3 3 3 3

3 3 3 3 3
3 3 3 3
Overall ranks:
1 1 1 5
OUTPUTS:
2 1 0
3 3 1 2

12 INPUTS:
2 1 2 0 3
1 1 1
3 3 3 3 3 3
3 3 3 3 3 3 3

3 3 3 3 3
3 3 3 3
Overall ranks:
1 1 2 5
OUTPUTS:
5 1 0
3 3 1 3

13 INPUTS:
2 1 1.8 6 5
1 2 1
1 1 1 1 1 1
1 3 2 2 3 4 4

2 1 5 3 2
5 4 4 4
Overall ranks:
1 3 2 5
OUTPUTS:
0 0 .8
3 3 1 3

14 INPUTS:
2 1 2.2 4 5
1 2 1
3 1 1 1 1 4
1 2 1 1 1 2 2

1 2 5 3 3
4 4 4 4
Overall ranks:
3 2 3 4
OUTPUTS:
2 1 0
2 2 1 3

15 INPUTS:
2 1 .36 5 3
0 0 0
1 3 2 2 2 3
4 4 2 4 3 4 4

3 4 4 4 2
5 4 5 5
Overall ranks:
2 3 2 5
OUTPUTS:
9 1 0
2 3 1 2

16 INPUTS:
1 1 3.6 9 2
0 0 0
2 4 2 2 3 2
4 4 5 4 4 4 4

5 5 4 4 4
3 3 4 2
Overall ranks:
2 4 5 3
OUTPUTS:
13 0 2
1 2 1 1

17 INPUTS:
2 1 8.6 6 4
2 1 2
3 1 1 2 2 3
1 5 2 3 5 5 2

1 1 4 3 1
1 1 1 1
Overall ranks:
1 5 2 1
OUTPUTS:
2 1 0
2 3 1 3

18 INPUTS:
1 1 43.7 21 3
2 1 3
2 3 2 2 2 3
3 3 5 4 3 4 2

1 1 2 2 3
4 5 3 5
Overall ranks:
2 4 4 5
OUTPUTS:
4 1 0
2 3 1 2

19 INPUTS:
1 1 12.3 20 3
1 0 0
3 5 5 3 2 4
4 4 3 2 3 4 4

3 3 5 3 4
5 5 2 5
Overall ranks:
4 4 3 5
OUTPUTS:
5 1 .8
3 3 1 2

20 INPUTS:
1 1 5.4 16 4
1 0 0
1 2 3 2 3 2
3 2 2 3 3 4 4

2 2 5 2 4
4 4 4 4
Overall ranks:
2 3 2 4
OUTPUTS:
4 0 .15
3 2 1 2

21 INPUTS:
2 1 10.1 18 4
1 2 1
4 1 2 3 2 4
3 4 4 4 3 4 5

2 2 5 2 3
4 4 4 5
Overall ranks:
3 4 2 4
OUTPUTS:
6.5 1 2
2 3 1 2

22 INPUTS:
1 1 15.7 20 4
1 2 1
2 2 3 2 2 4
5 5 4 2 4 4 5

3 2 5 2 3
5 3 4 5
Overall ranks:
3 5 3 4
OUTPUTS:
4.5 1 1
1 1 1.25 2

23 INPUTS:
2 1 15.1 16 3
1 2 2
3 3 3 2 2 2
2 3 3 2 3 5 3

3 2 3 3 3
5 5 5 5
Overall ranks:
2 3 3 5
OUTPUTS:
5 1 1
2 2 1 1

24 INPUTS:
2 1 2 17 4
2 2 0
3 4 3 2 2 2
4 4 3 3 3 4 5

2 3 5 4 3
4 4 5 4
Overall ranks:
3 4 3 4
OUTPUTS:
3 0 0
3 1 1 2

25 INPUTS:
1 2 1.7 5 5
1 1 0
1 2 1 1 3 4
2 3 1 1 1 2 4

1 1 5 3 3
5 4 4 5
Overall ranks:
2 3 2 4
OUTPUTS:
3 1 0
2 3 1 3

26 INPUTS:
2 1 5.3 6 4
1 1 0
1 4 4 2 4 2
4 5 3 5 3 4 2

3 4 1 3 3
4 4 2 4
Overall ranks:
4 4 3 3
OUTPUTS:
10 1 0
1 1 1 1

27 INPUTS:
2 3 18.5 10 4
1 2 0
4 2 3 3 2 4
3 3 4 2 3 4 2

2 2 4 2 2
2 3 2 4
Overall ranks:
3 3 2 3
OUTPUTS:
6 1 0
1 2 1 2

28 INPUTS:
2 3 4.9 11 4
1 2 0
2 1 3 2 2 1
2 3 2 4 3 4 3

2 2 3 3 4
1 3 4 4
Overall ranks:
2 3 3 3
OUTPUTS:
6 1 0
2 2 1 2

29 INPUTS:
2 1 2.3 6 3
0 1 2
2 4 2 2 2 3
3 4 4 3 3 5 5

2 2 4 2 2
4 4 4 4
Overall ranks:
2 3 2 4
OUTPUTS:
5 1 .08
2 2 1 2

30 INPUTS:
1 1 22 24 5
2 2 1
5 1 1 5 1 2
5 4 1 5 1 1 2

1 1 2 5 1
1 5 5 5
Overall ranks:
5 5 2 4
OUTPUTS:
7 1 10
2 3 1 1

31 INPUTS:
2 1 21 22 5
2 2 1
3 3 1 1 1 1
1 1 1 2 1 4 5

1 1 4 5 1
5 5 5 5
Overall ranks:
3 2 2 5
OUTPUTS:
4 0 2
1 2 1 3

32 INPUTS:
2 3 18.3 18 3
2 1 0
3 5 2 1 2 3
4 4 3 1 3 2 3

2 2 1 2 4
5 5 4 5
Overall ranks:
3 3 2 5
OUTPUTS:
5 1 0
1 2 1 2

33 INPUTS:
2 1 5.2 8 2
2 1 0
1 2 2 1 3 1
4 4 2 3 2 4 4

2 2 4 3 4
3 4 3 5
Overall ranks:
2 4 3 4
OUTPUTS:
3.5 1 0
2 3 1 2

34 INPUTS:
1 1 .49 4 4
0 2 1
3 3 5 3 2 5
3 4 3 5 3 1 5

3 3 4 3 3
3 3 2 3
Overall ranks:
4 4 3 3
OUTPUTS:
18 0 .2
2 1 1 1

35 INPUTS:
1 1 .61 4 4
0 2 1
2 4 3 3 2 4
3 3 4 3 2 1 5

4 4 3 4 3
5 5 5 5
Overall ranks:
3 3 2 5
OUTPUTS:
20 1 .2
2 1 1 2

36 INPUTS:
1 1 11.3 37 5
0 2 1
3 1 1 1 1 1
5 3 4 4 2 5 4

2 2 3 4 1
4 4 3 1
Overall ranks:
2 4 3 2
OUTPUTS:
5 1 0
2 3 1 3

37 INPUTS:
1 1 .39 7 3
0 0 1
3 2 1 2 2 4
3 4 4 4 3 5 2

3 3 3 2 3
4 4 4 4
Overall ranks:
3 3 3 4
OUTPUTS:
15 1 0
2 2 1 2

38 INPUTS:
1 1 3.5 9 4
0 0 0
4 1 1 1 1 3
2 2 3 2 3 4 3

1 1 2 3 3
2 4 4 5
Overall ranks:
2 3 4 4
OUTPUTS:
9 1 .5
2 2 1 1

39 INPUTS:
2 3 3.8 11 3
0 0 0
3 3 2 2 2 1
4 4 4 3 3 3 3

1 1 2 2 2
3 4 3 5
Overall ranks:
2 3 2 4
OUTPUTS:
8 1 0
1 2 1 1

40 INPUTS:
1 1 28.5 16 4
2 2 0
3 3 3 3 2 3
3 3 3 2 3 3 3

3 3 3 3 3
4 4 4 4
Overall ranks:
3 3 3 4
OUTPUTS:
4 1 10
2 3 1 2

41 INPUTS:
2 1 14.3 12 4
2 2 0
3 3 3 3 2 3
3 3 3 3 3 3 3

3 3 3 3 3
4 4 4 4
Overall ranks:
3 3 3 4
OUTPUTS:
4 1 0
2 3 1 2

42 INPUTS:
1 1 25 30 4
2 0 0
2 3 1 1 2 2
4 3 4 3 3 4 4

4 4 4 4 4
4 4 4 4
Overall ranks:
2 4 4 4
OUTPUTS:
5 1 0
1 2 1 1

43 INPUTS:
2 1 2.5 14 3
2 0 0
5 5 2 4 4 4
3 4 2 3 4 3 5

2 2 3 3 2
4 4 4 4
Overall ranks:
4 4 3 4
OUTPUTS:
1.5 1 0
1 1 1.143 3

44 INPUTS:
2 1 8 11 5
1 2 0
3 3 3 3 3 3
3 3 3 3 3 3 3

3 3 3 3 3
3 3 3 3
Overall ranks:
3 4 4 5
OUTPUTS:
2 1 0
2 3 1 3

45 INPUTS:
2 1 20 17 3
2 2 1
3 3 3 3 3 3
3 3 3 3 3 3 3

3 3 3 3 3
3 3 3 3
Overall ranks:
1 3 5 5
OUTPUTS:
5 1 9
2 3 1 2

46 INPUTS:
2 1 3.76 12 3
2 2 0
1 4 1 1 1 1
1 1 3 3 1 3 3

1 4 4 2 3
4 4 4 4
Overall ranks:
2 2 3 4
OUTPUTS:
4 0 0
3 3 1 3

47 INPUTS:
2 1 2.3 6 4
2 2 0
1 4 3 3 1 1
1 1 2 2 2 4 2

4 1 5 2 4
4 4 4 4
Overall ranks:
3 2 3 4
OUTPUTS:
1.5 1 0
2 2 1 2

48 INPUTS:
2 3 2 10 5
1 2 1
1 1 1 1 2 1
5 4 5 5 3 3 4

3 3 3 3 3
3 4 3 4
Overall ranks:
1 4 3 3
OUTPUTS:
5 1 0
2 2 1 3

49 INPUTS:
2 1 7.5 8 5
1 2 0
1 1 1 1 1 3
1 1 3 3 5 5 3

1 1 5 3 5
5 5 5 5
Overall ranks:
1 3 3 5
OUTPUTS:
2 0 1.5
3 3 1 3

50 INPUTS:
1 1 7.5 12 5
1 2 0
3 3 3 2 2 3
3 3 3 4 3 3 5

1 1 5 3 5
5 5 3 5
Overall ranks:
3 3 3 4
OUTPUTS:
4 0 2
3 3 1 3

51 INPUTS:
2 1 1.3 13 3
1 2 2
4 4 5 5 4 4
4 4 4 5 5 1 5

2 3 5 3 3
4 4 4 4
Overall ranks:
4 4 3 4
OUTPUTS:
11 1 .61
2 2 1 2

CASE E: Data used for testing the Hierarchical model (Global network).

.... KEY for the data fields in an example project ...

Ex. #; **INPUTS (total = 30) - not all have been used:**
Owner; Contract; Total bid; Contr. dur.; need for work.
Contractor size; Markup defin.; Markup components.
Site cond.; Owner attit.; Loc.; Hazz.; Inaccur.; Weather.
Tech.; Res.; Size; Qual.; Stack.; % sub.; Spcs.
Infl.; Escal.; Compet.; # of compet.; Econ. cond.; Res. avail.
Similar exper.; Mgmt cond.; Conf. in work force; Fin. capab.

<u>**Note:**</u> Overall ranks are produced by the trained sub-networks.

DESIRED OUTPUTS (total = 7) - not introduced to the neural network:
% Markup; Win(0); Diff.($)
C.O.s; claims; Dur. Exten.; Act. prof.

---

1 INPUTS:
1 1 2.5 9 4
1 2 2
2 4 4 3 1 2
3 4 4 5 4 5 5
1 1 5 3 3
4 4 4 5
OUTPUTS:
3.5 1 .01
3 3 1 2

2 INPUTS:
1 1 10 20 3
2 2 1
5 3 5 1 2 5
1 3 3 3 2 3 2
4 3 2 2 3
5 4 4 4
OUTPUTS:
10 1 5
3 3 1 1

3 INPUTS:
1 1 7.5 7 5
2 2 1
1 1 1 1 1 2
2 2 2 3 2 1 1
1 1 4 1 3
4 4 4 5
OUTPUTS:
1.5 1 0
3 3 1 3

4 INPUTS:
2 1 4.2 10 5
1 1 2
2 2 1 1 4 4
4 2 4 2 2 3 2
1 1 1 1 2
4 5 5 5
OUTPUTS:
8.2 1 0
2 3 1 2

5 INPUTS:
2 2 2 13 3
1 2 2
3 1 1 1 3 5
2 3 4 1 3 2 3
1 1 3 1 3
3 4 5 5
OUTPUTS:
2.3 1 0
2 3 1 2

6 INPUTS:
1 1 3.2 10 4
1 2 1
2 1 1 1 2 1
2 2 3 2 2 4 4
3 3 4 2 3
5 4 5 4
OUTPUTS:
3 1 .11
2 3 1 2

7 INPUTS:
2 1 2 12 4
1 2 1
3 1 1 1 1 3
2 4 4 3 2 2 2
2 2 4 3 3
4 4 4 3
OUTPUTS:
6.5 1 .5
2 3 1 2

8 INPUTS:
1 1 21 22 3
1 0 0
1 3 3 3 2 3
4 4 4 5 3 4 5
3 3 3 3 3
5 5 4 5
OUTPUTS:
2 1 11
1 3 1 1

9 INPUTS:
1 1 13.5 12 3
1 0 4
2 4 3 3 4 5
4 4 4 3 4 5 5
3 3 4 2 3
2 4 3 4
OUTPUTS:
3 1 2.25
2 3 1 3

10 INPUTS:
2 1 2 12 3
1 0 4
2 5 4 2 4 5
2 2 5 5 4 5 4

3 3 5 3 3
2 4 4 4
OUTPUTS:
3 1 .08
3 3 1.167 2

11 INPUTS:
1 1 5 14 4
1 1 2
4 3 3 1 1 4
4 2 4 1 3 5 3

2 2 3 2 3
5 5 4 5
OUTPUTS:
5 1 0
2 3 1 1

12 INPUTS:
1 1 2.7 12 3
1 1 2
4 2 1 2 3 2
5 3 2 2 2 2 3

2 3 4 4 3
5 4 4 2
OUTPUTS:
4 1 .6
3 3 1 2

13 INPUTS:
1 1 2 7 4
0 2 0
4 5 5 5 1 2
3 3 2 3 3 5 5

3 2 3 3 3
3 4 5 5
OUTPUTS:
7 1 .05
3 2 1 3

14 INPUTS:
2 3 7.5 14 5
1 2 0
4 3 1 1 1 2
1 3 2 3 1 4 2

1 1 3 3 3
5 5 5 4
OUTPUTS:
7 1 0
3 3 1 2

15 INPUTS:
1 1 2 12 3
0 0 1
4 3 3 2 2 3
1 1 3 4 3 2 4

3 2 4 2 3
4 4 3 4
OUTPUTS:
7.5 1 0
2 3 1.08 2

16 INPUTS:
1 1 2 14 4
0 0 1
5 3 3 4 4 1
2 3 3 2 3 3 3

2 2 4 3 4
2 4 2 5
OUTPUTS:
7.5 1 0
1 2 1.143 4

17 INPUTS:
2 3 1.4 4 5
1 1 3
2 2 2 2 2 2
2 3 4 4 3 5 3

1 1 3 3 1
4 4 5 5
OUTPUTS:
4 1 0
3 3 1 2

18 INPUTS:
1 1 6.5 18 1
1 2 2
1 1 1 1 1 5
3 3 3 4 4 4 4

3 3 4 3 3
3 4 5 5
OUTPUTS:
5 1 0
2 3 1 1

19 INPUTS:
2 1 7.5 18 1
1 2 2
3 3 2 2 2 2
4 4 4 3 4 4 3

1 1 3 2 3
3 4 4 5
OUTPUTS:
3 1 0
1 2 1 2

20 INPUTS:
2 1 7.5 12 4
1 0 0
1 1 1 1 3 3
2 4 2 2 2 4 4

2 2 2 3 3
5 5 5 5
OUTPUTS:
4.3 1 0
1 3 1 2

21 INPUTS:
2 1 5 15 4
1 1 0
1 4 4 4 2 2
3 3 2 2 3 3 2

3 3 4 3 3
3 2 4 4
OUTPUTS:
4 1 0
1 3 1.067 4

22 INPUTS:
1 1 19 36 5
1 2 0
3 5 5 4 1 5
4 5 5 2 4 1 5

3 3 5 4 5
3 4 5 4
OUTPUTS:
3 1 10
1 1 1 2

24 INPUTS:
2 3 5 0 2
1 2 0
1 1 3 2 1 3
5 5 5 1 4 4 4

4 3 5 1 3
5 5 5 5
OUTPUTS:
4 1 0
1 1 1 3

23 INPUTS:
2 3 5 48 4
1 2 0
5 5 2 5 1 1
5 5 5 1 5 5 5

4 4 5 2 2
5 5 5 5
OUTPUTS:
3 1 0
1 1 1 3

25 INPUTS:
1 1 19.6 16 4
1 0 0
2 1 3 1 1 2
3 3 5 2 2 3 2

1 2 3 2 4
5 4 5 5
OUTPUTS:
2 1 7
2 3 1.125 2

APPENDIX III: Backpropagation algorithm.

```
#include <stdio.h>
#include <math.h>
#include <ctype.h>
#ifndef VAX
/* declaration

#include <malloc.h>

/* define constants

#define NMXUNIT     70 /* units in a layer*/
#define NMXHLR      3  /* no. of hidden layers*/
#define NMXOATTR    10 /* output features*/
#define NMXINP      100 /* input samples*/
#define NMXIATTR    40 /* input features*/
#define SEXIT       3  /* exit successfully*/
#define RESTRT      2 /* restart*/
#define FEXIT       1 /* exit in failure*/
#define CONTNE      0 /* continue calculation*/

/* data base : declaration of variables */

float eta;      /* learning rate */
float alpha;    /* momentum rate */
float err_curr;
float maxe;
float maxep;
float *wtptr[NMXHLR+1];
float *outptr[NMXHLR+2];
float *errptr[NMXHLR+2];
float *delw[NMXHLR+1];
float target[NMXINP][NMXOATTR];
float input[NMXINP][NMXIATTR], ep[NMXINP];
float outpt[NMXINP][NMXOATTR];
int  nunit[NMXHLR+2], nhlayer,ninput, ninattr, noutattr;
int  result, cnt, cnt_num;
int  nsnew, nsold;
char  task_name[20], dtype[1];
FILE  *fp1, *fp2, *fp3, *fopen();
int  fplot10;


    /* random number generator */
long randseed = 568731L;
int random()
{
  randseed = 15625L * randseed +22221L;
  return((randseed >> 16) & 0x7FFF);
}


    /* alocate dynamic storage for the net */
init()
{
```

```
 int len1, len2, i, k;
 float *p1, *p2, *p3, *p4;
 len1 = len2 = 0;
 nunit[nhlayer+2] = 0;
 for (i=0; i<(nhlayer + 2); i++) {
        len1 += (nunit[i] + 1) * nunit[i+1];
        len2 += nunit[i] + 1;
 }
                        /* weights */
 p1=(float *) calloc(len1+1,sizeof(float));
                        /* output */
 p2=(float *) calloc(len2+1,sizeof(float));
                        /* errors */
 p3=(float *) calloc(len2+1,sizeof(float));
                        /* delw */
 p4=(float *) calloc(len1+1,sizeof(float));
                        /* set up initial pointers */
 wtptr[0] = p1;
 outptr[0] = p2;
 errptr[0] =p3;
 delw[0]   =p4;
                        /* set up the rest of pointers */
 for (i=1 ; i <(nhlayer +1)  ; i++) {
        wtptr[i] = wtptr[i-1] + nunit[i] * (nunit[i-1] +1);
        delw[i]  = delw[i-1] + nunit[i] * (nunit[i-1] +1);
 }
 for (i=1; i<(nhlayer+2); i++)  {
        outptr[i] = outptr[i-1] + nunit[i-1] + 1;
        errptr[i] = errptr[i-1] + nunit[i-1] + 1;
 }                      /* setup threshold outputs */
 for (i= 0; i<nhlayer +1; i++)  {
    *(outptr[i] + nunit[i]) = 1.0;
 }
}


              /* initialize weights with random between 0.5 and -0.5 */
initwt()
{
   int i,j;
   for (j = 0; j<nhlayer+1; j++)
          for (i=0; i<(nunit[j]+1) * nunit[j+1] ; i++)  {
               *(wtptr[j]+i) = random()/pow(2.0,15.0) - 0.5;
               *(delw[j] + i) = 0.0;
   }
}


              /* specify net architecture and
                    values for learning parameters */
set_up()
{
   int i;
   nunit[nhlayer+1] = noutattr;
   nunit[0]=ninattr;
   }
```

```
                /* read file for net architecture and
                   learning parameters. File name has .NET */
dread (taskname)
char *taskname;
{
  int i,j,c;
  char var_file_name[20];
  strcpy(var_file_name, taskname);
  strcat(var_file_name,".net");
  fp1 = fopen(var_file_name,"r");
  fscanf(fp1,"%s",&task_name);
  fscanf(fp1,"%d%d%d%f%f%f%f%d%d",&ninput,&noutattr,
       &ninattr,&eta,&alpha,&maxe,&maxep,&nhlayer,&cnt_num);
  for  ( i=0; i < nhlayer+2; i++)
       fscanf (fp1,"%d", &nunit[i]);
  fclose(fp1);
  }

                        /* read file containing weights
                           and thresholds. File name has
                           suffix _w.dat*/
  wtread(taskname)
  char *taskname;
  {
       int  i, j, c;
       char wt_file_name[20];
       strcpy(wt_file_name,taskname);
       strcat(wt_file_name,"_w.dat");
       fp2 = fopen(wt_file_name,"r");
       for (i =0; i <nhlayer +1; i++) {
          for (j =0; j < (nunit[i]+1) * nunit[i+1]; j++) {
             fscanf (fp2, "%f", (wtptr[i]+j));
          }
       }
       fclose(fp2);
  }


                        /* create file for net architecture
                           and learning parameters. File name has .net */
      dwrite(taskname)
      char *taskname;
      {
     int  i, j, c;
      char  var_file_name[20];
      strcpy(var_file_name,taskname);
      strcat(var_file_name, ".net");
      fp1 = fopen(var_file_name,"w+");
      fprintf (fp1,"%s\n",task_name);
      fprintf (fp1,"%u %u %u %f %f %2.9f %2.9f %u %u\n",ninput,noutattr,
                                               ninattr,eta,alpha,maxe,maxep,
                                               nhlayer,cnt_num);
      for (i=0; i <nhlayer+2;i++) {
         fprintf(fp1,"%d ",nunit[i]);
      }
      fprintf(fp1,"\n%d %f", cnt,err_curr);
```

```
    fprintf(fp1,"\n");

    for(i=0;i<ninput;i++)
    {
      for(j=0;j<noutattr;j++)
     fprintf(fp1,"%f        ",outpt[i][j]);
      fprintf(fp1,"\n");
    }
    fclose(fp1);
    }


                        /* create file for saving weights and
                           thresholds learned from training.
                           File name has suffix _w.dat */
wtwrite(taskname)
char *taskname;
{
     int  i, j, c, k;
     char wt_file_name[20];
     strcpy(wt_file_name,taskname);
     strcat(wt_file_name, "_w.dat");
     fp2 = fopen(wt_file_name,"w+");
     k=0;
     for (i=0; i <nhlayer+1; i++)
       for ( j=0; j <(nunit[i] +1) * nunit[i+1]; j++) {
         if(k==8) {
                k=0;
                fprintf(fp2,"\n");
           }
           fprintf(fp2,"%f ", *(wtptr[i] +j));
           k++;
      }
      fclose(fp2);
}


                    /* bottom_up calculation
                       of net for input pattern i */
void forward(i)
{
       int  m, n, p, offset;
       float net;
                              /* input level output calculation*/
       for (m=0; m<ninattr;m++)
          *(outptr[0]+m) = input[i][m];
                    /* hidden & output layer output calcultion*/
       for (m=1; m<nhlayer+2;m++) {
        for (n=0;n<nunit[m];n++)   {
          net=0.0;
          for (p=0; p<nunit[m-1]+1;p++) {
             offset = (nunit[m-1]+1) * n + p;
             net += *(wtptr[m-1]+offset)*(*(outptr[m-1]+p));
          }
          *(outptr[m]+n) = 1/(1+exp(-net));
        }
```

```
  }
  for (n=0; n<nunit[nhlayer+1];n++)
        outpt[i][n] = *(outptr[nhlayer+1]+n);
}


                  /* several condition are checked to
                     see whether learning should terminate */
int introspective (nfrom,nto)
int nfrom;
int nto;
{
      int i,flag;
                              /* reached max. itreration ?*/
      if (cnt>=cnt_num) return(FEXIT);
                              /* error for each pattern small enough? */
      nsnew =0;
      flag = 1;
      for (i =nfrom; (i < nto)&& (flag == 1); i++) {
         if (ep[i] <= maxep) nsnew++;
         else flag = 0;
      }
      if (flag == 1) return(SEXIT);
                              /* system total error small
                                        enough? */
      if ( err_curr <= maxe) return(SEXIT);
      return(CONTNE);
      }


                  /* bias (threshold) is treated as weight of link from a
                     virtual node whose output value is unity */
      int rumelhart(from_snum,to_snum)
      int from_snum;
      int to_snum;
      {
         int  i,j,k,m,n,p,offset, index;
         float out;
         char *err_file = "error.dat";
         nsold = 0;
         cnt = 0;
         result = CONTNE;
         if (fplot10 == 1)
            fp3=fopen(err_file,"w");
            do {
            err_curr = 0.0;
                                 /* for each pattern   */
            for (i=from_snum; i<to_snum;i++) {
                                 /* bottom_up calcultion */
                  forward(i);
                                 /* output_level error */
            for (m=0; m<nunit[nhlayer+1]; m++)  {
                  out= *(outptr[nhlayer+1] +m);
                  *(errptr[nhlayer+1]+m) = (target[i][m]-out)
                                             * (1-out) * out;
            }
```

```
                    /* hidden & input layer errors  */
 for (m= nhlayer+1; m>=1; m--)  {
      for ( n=0; n< nunit[m-1]+1; n++)  {
          *(errptr[m-1]+n)= 0.0;
          for (p=0; p<nunit[m]; p++)  {
                offset = (nunit[m-1]+1) * p + n;
                *(delw[m-1]+offset)=eta * (*(errptr[m]+p))
                    * (*(outptr[m-1]+n))
                    +alpha * (*(delw[m-1]+offset));
                *(errptr[m-1]+n) += *(errptr[m]+p)
                            * (*(wtptr[m-1]+ offset));
          }
          *(errptr[m-1]+n) = *(errptr[m-1]+n) *
                (1- *(outptr[m-1]+n)) *(*(outptr[m-1]
                +n));
      }
 }
                    /* weight changes */
 for (m=1 ; m<nhlayer+2; m++)  {
        for (n=0; n<nunit[m]; n++)  {
          for (p=0; p<nunit[m-1]+1; p++)  {
                offset= (nunit[m-1]+1) * n + p;
                *(wtptr[m-1]+offset) += *(delw[m-1]
                +offset);
          }
        }
 }
 ep[i] = 0.0;
 for (m=0; m<nunit[nhlayer+1]; m++)  {
        ep[i]  += fabs((target[i][m] -
                  *(outptr[nhlayer+1]+m)));
 }
 err_curr  += ep[i] * ep[i];
}
              /* normalized system error */
  err_curr = 0.5 * err_curr/ninput;
        /** save errors in file to draw the system
            error with plot10 **/
  if (fplot10==1)
  fprintf(fp3, "%1d, %2.9f\n",cnt,err_curr);
  cnt++;
                        /* update output with changed weights */
  for (i=from_snum; i<to_snum;i++) forward(i);
                        /* check condition for terminating
                              learning */
  result = introspective (from_snum,to_snum);
  wtwrite(task_name);
} while (result == CONTNE);
 return(result);
}


          /* read in the input data file specified by user
             during the interactive session*/
user_session()
{
```

```
int  i, j, showdata;
char fnam[20];
FILE  *fp;
            /* for task with name task_name, input data file
                 of is "task_name.lrn" */
strcpy(fnam,task_name);
strcat(fnam,".lrn");
fp = fopen(fnam,"r");
                    /* Do you want to look at data just read ? (Y/N) */
showdata = ((dtype[0] == 'y') || (dtype[0] == 'Y'));
for ( i=0; i<ninput;i++)  {
      for (j=0; j<ninattr;j++)  {
         fscanf(fp,"%f",&input[i][j]);
         if (showdata) printf("%f  ",input[i][j]);
      }
      for (j=0; j<noutattr;j++) {
         fscanf(fp,"%f",&target[i][j]);
         if (showdata) printf("%f\n",target[i][j]);
      }
 }
 fclose(fp);
}


                    /* main body of learning */
learning()
{
      int result;
      char wt_name[15];
      FILE *fy;
      strcpy(wt_name,task_name);
      strcat(wt_name, "_w.dat");
      user_session();
      set_up();
      init();
      do  {
         if ((fy=fopen(wt_name,"r")) == NULL)
         {
 initwt();
         }
         else
         {
         fclose(fy);
         wtread(task_name);
         }
         result = rumelhart(0,ninput);
      } while (result == RESTRT);
       dwrite(task_name);
 }


                    /* main body of output generation */
 output_generation( )
 {
       int  i,c,m,nsample;
```

```c
char  dfile[20], var_file_name[20],ans[10];
FILE *ft;
        /* If task is already in the memory, data
         files for task do not need to be read in
         But, if it is a new task, data files
         should be read in to reconstruct the net */
   init();
   wtread(task_name);
 strcpy(dfile,task_name);
  strcat(dfile,".rec");
  fp1=fopen(dfile,"r");
  fscanf(fp1,"%d", &nsample);
       /* create file for the final output
          from recall patterns. File name
          has  .out */
       strcpy(var_file_name,task_name);
       strcat(var_file_name, ".out");
       ft=fopen(var_file_name,"w");
  for(i=0;i<nsample;i++)
    for (m=0;m<ninattr;m++)
         fscanf(fp1,"%f",&input[i][m]);
                 /*output generation calculation starts */
  for(i=0;i<nsample;i++)
  {
    forward(i);
    for (m=0;m<noutattr;m++)
        fprintf(ft,"%f\n", *(outptr[nhlayer+1]+m));
   }
   fclose(fp1);
   fclose(ft);
 }


 /****************** MAIN ********************/
 main(int argc, char *argv[])
 {
   char select[20];
   strcpy(task_name, argv[1]);  /* argument for task name */
   strcpy(dtype, argv[2]);      /* argument for viewing input data (Y/N) */
   strcpy(select, argv[3]);     /* argument for (L)earn or (O)utput */
   fplot10=atoi(argv[4]);       /* argument for (1)create error file or not(0) */
   dread(task_name);
        switch(select[0])  {
          case 'o':
          case 'O':
            output_generation ();
            break;
          case 'l':
          case 'L':
            learning();
            break;
          default:
            break;
         }
}
#endif
```

APPENDIX IV: Training and testing errors of neural networks.

Case (a): Single-network model - network (30 30 7) - Hand crafting.

a.1 Training Results.

Neural Network Results for colct

Ex. #	.... ERRORS (%) in OUTPUT attributes....... (1)	(2)	(3)	(4)	(5)	(6)	(7)	.. STATISTICS .. MEAN,	ST.DEV.
1	-0.15	0.00	0.91	0.00	0.00	0.04	0.00	0.11	0.329
2	-0.05	0.00	0.20	0.00	0.00	-0.12	0.00	0.01	0.088
3	-0.18	0.00	0.20	0.00	0.00	0.03	0.00	0.01	0.103
4	-0.53	0.00	0.00	0.00	0.00	-0.06	0.00	-0.09	0.185
5	-0.47	0.00	0.01	0.00	0.00	-0.04	0.00	-0.07	0.164
6	0.43	0.00	0.19	0.00	0.00	-0.02	0.00	0.09	0.155
7	-0.61	0.00	0.08	0.00	0.00	-0.05	0.00	-0.08	0.217
8	-1.69	0.00	0.11	0.00	0.00	0.08	0.00	-0.21	0.605
9	0.24	0.00	-0.80	0.00	0.00	-0.06	0.00	-0.09	0.304
10	-0.90	0.00	-0.74	0.00	0.00	0.06	0.00	-0.23	0.380
11	-0.22	0.00	0.48	0.00	0.00	0.06	0.00	0.05	0.197
12	1.98	0.00	0.37	0.00	0.00	0.06	0.00	0.34	0.679
13	-0.50	0.00	0.04	0.00	0.00	0.09	0.00	-0.05	0.185
14	-0.62	0.00	0.03	0.00	0.00	0.24	0.00	-0.05	0.244
15	-1.26	0.00	-35.74	0.00	0.00	0.18	0.00	-5.26	12.451
16	-0.28	0.00	-1.99	0.00	0.00	0.03	0.00	-0.32	0.687
17	-0.34	0.00	-9.10	0.00	0.00	0.15	0.00	-1.33	3.176
18	-0.73	0.00	-0.20	0.00	0.00	0.04	0.00	-0.13	0.256
19	-0.08	0.00	-59.54	0.00	0.00	0.14	0.00	-8.50	20.838
20	-0.31	0.00	0.00	0.00	0.00	0.11	0.00	-0.03	0.120
21	0.20	0.00	0.13	0.00	0.00	0.13	0.00	0.07	0.078
22	-0.04	0.00	0.15	0.00	0.00	0.09	0.00	0.03	0.062
23	-0.10	0.00	-2.60	0.00	0.00	0.01	0.00	-0.39	0.905
24	0.72	0.00	0.24	0.00	0.00	0.04	0.00	0.14	0.248
25	-1.12	0.00	0.15	0.00	0.00	-0.03	0.00	-0.14	0.403
26	-0.59	0.00	0.06	0.00	0.00	-0.09	0.00	-0.09	0.210
27	-0.24	0.00	-0.34	0.00	0.00	-0.03	0.00	-0.09	0.133
28	-1.55	0.00	0.13	0.00	0.00	0.02	0.00	-0.20	0.552
29	-0.20	0.00	0.24	0.00	0.00	-0.01	0.00	0.00	0.120
30	-0.18	0.00	0.01	0.00	0.00	-0.03	0.00	-0.03	0.064
31	-0.08	0.00	-0.43	0.00	0.00	0.09	0.00	-0.06	0.156
32	0.14	0.00	-6.73	0.00	0.00	-0.02	0.00	-0.94	2.364
33	-0.05	0.00	-0.15	0.00	0.00	-0.05	0.00	-0.04	0.050
34	-0.05	0.00	-0.05	0.00	0.00	-0.03	0.00	-0.02	0.023
35	-0.57	0.00	-0.83	0.00	0.00	0.06	0.00	-0.19	0.331
36	-0.91	0.00	0.37	0.00	0.00	-0.05	0.00	-0.08	0.362
37	-0.62	0.00	0.14	0.00	0.00	0.00	0.00	-0.07	0.231
38	-0.34	0.00	0.01	0.00	0.00	0.05	0.00	-0.04	0.125
39	0.13	0.00	0.00	0.00	0.00	0.09	0.00	0.03	0.049
40	0.05	0.00	0.00	0.00	0.00	0.07	0.00	0.02	0.028
41	0.13	0.00	-0.39	0.00	0.00	0.16	0.00	-0.01	0.166
42	-0.00	0.00	-0.04	0.00	0.00	0.11	0.00	0.01	0.043
43	-0.16	0.00	0.45	0.00	0.00	0.10	0.00	0.06	0.175
44	0.48	0.00	0.00	0.00	0.00	0.17	0.00	0.09	0.170
45	-0.22	0.00	0.00	0.00	0.00	0.14	0.00	-0.01	0.097
46	0.10	0.00	0.36	0.00	0.00	-0.09	0.00	0.05	0.136
47	-0.01	0.00	0.14	0.00	0.00	0.05	0.00	0.03	0.051
48	0.30	0.00	1.33	0.00	0.00	-0.06	0.00	0.23	0.464
49	0.14	0.00	2.70	0.00	0.00	0.03	0.00	0.41	0.936
50	-0.14	0.00	0.08	0.00	0.00	0.05	0.00	-0.00	0.062

Ex. #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	MEAN,	ST.DEV.
	.... ERRORS (%) in OUTPUT attributes......							.. STATISTICS ..	
51	0.27	0.00	0.32	0.00	0.00	0.02	0.00	0.09	0.133
52	0.11	0.00	-0.58	0.00	0.00	0.03	0.00	-0.06	0.215
53	-0.51	0.00	0.00	0.00	0.00	0.00	0.00	-0.07	0.179
54	-0.33	0.00	0.00	0.00	0.00	0.07	0.00	-0.04	0.123
55	-0.42	0.00	-0.24	0.00	0.00	-0.01	0.00	-0.10	0.157
56	0.20	0.00	0.25	0.00	0.00	-0.03	0.00	0.06	0.105
57	-0.42	0.00	0.32	0.00	0.00	0.06	0.00	-0.01	0.199
58	-0.77	0.00	0.01	0.00	0.00	0.10	0.00	-0.09	0.277
59	0.35	0.00	0.23	0.00	0.00	0.03	0.00	0.09	0.132
60	-1.35	0.00	0.13	0.00	0.00	0.14	0.00	-0.15	0.494
61	0.45	0.00	0.17	0.00	0.00	-0.02	0.00	0.08	0.160
62	-0.40	0.00	0.03	0.00	0.00	0.00	0.00	-0.05	0.143
63	-1.63	0.00	0.00	0.00	0.00	-0.00	0.00	-0.23	0.571
64	-0.01	0.00	0.01	0.00	0.00	-0.01	0.00	-0.00	0.005
65	-0.59	0.00	-0.06	0.00	0.00	0.03	0.00	-0.09	0.205

Attribute (1): Markup Estiamte (%). Attribute (2): Win/Loose.
Attribute (3): Difference ($). Attribute (4): Potential for C.O.'s.
Attribute (5): Potential for calims. Attribute (6): Duration extention.
Attribute (7): Project profitability.

ATTRIBUTES' ERROR ANALYSIS:

At.	MEAN-ERROR (%)	ST. DEV.	MAXIMUM-(EX.)	MINIMUM-(EX.)	RANGE
1	-0.25	0.575	1.9800 (12)	-1.6940 (8)	3.6740
2	0.00	0.000	0.0000 (1)	0.0000 (1)	0.0000
3	-1.69	8.584	2.7000 (49)	-59.5375 (19)	62.2375
4	0.00	0.000	0.0000 (1)	0.0000 (1)	0.0000
5	0.00	0.000	0.0000 (1)	0.0000 (1)	0.0000
6	0.04	0.073	0.2372 (14)	-0.1158 (2)	0.3530
7	0.00	0.000	0.0000 (1)	0.0000 (1)	0.0000

GLOBAL ERROR ANALYSIS:

Average error (%): -0.27 St. Deviation : 3.304
Maximum error is : 2.7000 Occured at Ex.: 49 - Attr.: 3
Minimum error is : -59.5375 Occured at Ex.: 19 - Attr.: 3
Variation Range : 62.2375

a.2 Testing results.

Neural Network Results for test-c

Ex. #	.... ERRORS (%) in OUTPUT attributes....... (1)	(2)	(3)	(4)	(5)	(6)	(7)	.. STATISTICS .. MEAN,	ST.DEV.
1	16.30	0.00	0.00	0.00	0.00	2.08	-66.67	-6.90	25.018
2	-66.89	0.00	0.00	0.00	200.00	-25.23	-66.67	5.89	83.932
3	85.74	0.00	0.00	-50.00	0.00	11.23	-33.33	1.95	39.752
4	115.50	0.00	26.09	0.00	0.00	5.01	-50.00	13.80	46.618
5	-59.05	0.00	-72.65	0.00	-33.33	-15.37	0.00	-25.77	27.937
6	52.77	0.00	-92.31	-33.33	-33.33	-4.15	-33.33	-20.53	40.895
7	-38.62	0.00	-99.43	50.00	0.00	-0.17	50.00	-5.46	48.062

Attribute (1): Markup Estiamte (%). Attribute (2): Win/Loose.
Attribute (3): Difference '$). Attribute (4): Potential for C.O.'s.
Attribute (5): Potential or calims. Attribute (6): Duration extention.
Attribute (7): Project profitability.

ATTRIBUTES' ERROR ANALYSIS:

At.	MEAN-ERROR (%)	ST. DEV.	MAXIMUM-(EX.)	MINIMUM-(EX.)	RANGE
1	15.11	67.186	115.4975 (4)	-66.8900 (2)	182.3875
2	0.00	0.000	0.0000 (1)	0.0000 (1)	0.0000
3	-34.04	48.188	26.0933 (4)	-99.4328 (7)	125.5261
4	-4.76	29.161	50.0000 (7)	-50.0000 (3)	100.0000
5	19.05	75.292	200.0000 (2)	-33.3333 (5)	233.3333
6	-3.80	11.615	11.2326 (3)	-25.2278 (2)	36.4604
7	-28.57	38.539	50.0000 (7)	-66.6667 (1)	116.6667

GLOBAL ERROR ANALYSIS:

Average error (%): -5.29 St. Deviation : 49.833
Maximum error is : 200.0000 Occured at Ex.: 2 - Attr.: 5
Minimum error is : -99.4328 Occured at Ex.: 7 - Attr.: 3
Variation Range : 299.4328

Weighted error = .7955352

Case (b): Hierarchical model - global network (9 35 7) - Hand crafting.

b.1 Training Results.

Neural Network Results for globe

Ex. #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	MEAN,	ST.DEV.
	.... ERRORS (%) in OUTPUT attributes.......							.. STATISTICS ..	
1	0.75	0.00	-58.37	0.00	0.00	0.04	0.00	-8.23	20.472
2	0.08	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.029
3	-0.14	0.00	0.00	0.00	0.00	0.01	0.00	-0.02	0.049
4	0.12	0.00	0.00	0.00	0.00	0.04	0.00	0.02	0.041
5	-0.54	0.00	0.02	0.00	0.00	0.14	0.00	-0.05	0.203
6	0.55	0.00	-100.00	0.00	0.00	0.03	0.00	-14.20	35.027
7	-2.19	0.00	0.36	0.00	0.00	0.02	0.00	-0.26	0.797
8	-3.01	0.00	0.13	0.00	0.00	0.20	0.00	-0.38	1.077
9	4.45	0.00	0.81	0.00	0.00	-0.08	0.00	0.74	1.542
10	0.05	0.00	0.14	0.00	0.00	0.08	0.00	0.04	0.052
11	-1.91	0.00	0.00	0.00	0.00	0.15	0.00	-0.25	0.678
12	0.56	0.00	0.00	0.00	0.00	0.06	0.00	0.09	0.192
13	0.94	0.00	0.84	0.00	0.00	-0.02	0.00	0.25	0.405
14	-11.90	0.00	0.05	0.00	0.00	0.12	0.00	-1.67	4.173
15	0.06	0.00	0.00	0.00	0.00	0.04	0.00	0.01	0.022
16	0.05	0.00	-100.00	0.00	0.00	0.08	0.00	-14.27	35.000
17	-0.78	0.00	0.00	0.00	0.00	0.06	0.00	-0.10	0.279
18	0.13	0.00	0.07	0.00	0.00	0.04	0.00	0.04	0.048
19	0.04	0.00	-100.00	0.00	0.00	0.01	0.00	-14.28	34.996
20	0.24	0.00	-99.96	0.00	0.00	0.04	0.00	-14.24	34.995
21	0.11	0.00	0.28	0.00	0.00	0.07	0.00	0.07	0.098
22	-1.41	0.00	-99.99	0.00	0.00	0.02	0.00	-14.48	34.912
23	-0.04	0.00	0.67	0.00	0.00	0.01	0.00	0.09	0.236
24	-0.30	0.00	0.80	0.00	0.00	0.06	0.00	0.08	0.316
25	0.36	0.00	0.04	0.00	0.00	0.01	0.00	0.06	0.123
26	0.11	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.037
27	0.05	0.00	0.04	0.00	0.00	0.01	0.00	0.01	0.020
28	-0.11	0.00	0.07	0.00	0.00	-0.00	0.00	-0.01	0.050
29	-0.65	0.00	-99.95	0.00	0.00	-0.01	0.00	-14.37	34.938
30	0.25	0.00	-0.40	0.00	0.00	-0.03	0.00	-0.03	0.177
31	-0.64	0.00	0.29	0.00	0.00	-0.05	0.00	-0.06	0.260
32	-0.08	0.00	0.00	0.00	0.00	-0.03	0.00	-0.02	0.028
33	0.22	0.00	0.00	0.00	0.00	0.04	0.00	0.04	0.077
34	-0.08	0.00	-100.00	0.00	0.00	-0.07	0.00	-14.31	34.982
35	-0.04	0.00	-96.57	0.00	0.00	-0.03	0.00	-13.80	33.789
36	-0.31	0.00	0.00	0.00	0.00	-0.03	0.00	-0.05	0.109
37	-0.04	0.00	0.00	0.00	0.00	0.01	0.00	-0.00	0.014
38	-0.03	0.00	-100.00	0.00	0.00	-0.04	0.00	-14.30	34.988
39	-0.05	0.00	0.00	0.00	0.00	-0.02	0.00	-0.01	0.018
40	-1.60	0.00	-51.47	0.00	0.00	-0.02	0.00	-7.58	17.926
41	-1.60	0.00	48.53	0.00	0.00	-0.02	0.00	6.70	17.084
42	0.29	0.00	0.00	0.00	0.00	-0.07	0.00	0.03	0.108
43	-0.95	0.00	0.01	0.00	0.00	-0.01	0.00	-0.14	0.332
44	4.48	0.00	0.01	0.00	0.00	-0.02	0.00	0.64	1.568

Ex. #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	MEAN,	ST.DEV.
	.... ERRORS (%) in OUTPUT attributes.......							.. STATISTICS ..	
45	-0.39	0.00	-0.54	0.00	0.00	0.03	0.00	-0.13	0.217
46	-1.04	0.00	0.53	0.00	0.00	-0.05	0.00	-0.08	0.435
47	25.49	0.00	0.58	0.00	0.00	-0.01	0.00	3.72	8.889
48	-0.04	0.00	0.00	0.00	0.00	-0.00	0.00	-0.01	0.013
49	-6.06	0.00	-0.56	0.00	0.00	0.08	0.00	-0.94	2.104
50	0.99	0.00	-0.26	0.00	0.00	-0.11	0.00	0.09	0.378
51	0.11	0.00	0.12	0.00	0.00	-0.01	0.00	0.03	0.055

Attribute (1): Markup Estiamte (%). Attribute (2): Win/Loose.
Attribute (3): Difference ($). Attribute (4): Potential for C.O.'s.
Attribute (5): Potential for calims. Attribute (6): Duration extention.
Attribute (7): Project profitability.

ATTRIBUTES' ERROR ANALYSIS:

At.	MEAN-ERROR (%)	ST. DEV.	MAXIMUM-(EX.)	MINIMUM-(EX.)	RANGE
1	0.09	4.200	25.4900 (47)	-11.8950 (14)	37.3850
2	0.00	0.000	0.0000 (1)	0.0000 (1)	0.0000
3	-18.70	39.578	48.5255 (41)	-100.0000 (6)	148.5255
4	0.00	0.000	0.0000 (1)	0.0000 (1)	0.0000
5	0.00	0.000	0.0000 (1)	0.0000 (1)	0.0000
6	0.02	0.058	0.2038 (8)	-0.1116 (50)	0.3154
7	0.00	0.000	0.0000 (1)	0.0000 (1)	0.0000

GLOBAL ERROR ANALYSIS:

Average error (%): -2.66 St. Deviation : 16.407
Maximum error is : 48.5255 Occured at Ex.: 41 - Attr.: 3
Minimum error is :-100.0000 Occured at Ex.: 6 - Attr.: 3
Variation Range : 148.5255

b.2 Testing results.

Neural Network Results for test-g

Ex. #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	MEAN,	ST.DEV.
	.... ERRORS (%) in OUTPUT attributes.......							.. STATISTICS ..	
1	-45.55	66.67	13.23	-75.00	-60.00	-30.01	-80.00	-30.09	49.106
2	-93.70	25.00	-92.43	-40.00	-20.00	562.91	-80.00	37.40	218.249
3	-98.59	-100.00	-32.28	200.00	-50.00	-48.65	0.00	-18.50	95.043
4	-81.17	66.67	-0.28	-25.00	-25.00	-43.38	-33.33	-20.21	42.115
5	-50.55	100.00	0.02	60.00	-71.43	160.77	100.00	42.69	79.596
6	-67.21	400.00	-71.07	200.00	-25.00	143.55	0.00	82.90	161.553
7	-80.02	100.00	-92.44	0.00	20.00	-16.01	-40.00	-15.50	60.384
8	-91.16	150.00	-100.00	0.00	50.00	-11.51	-50.00	-7.52	80.664
9	-97.36	100.00	-100.00	-60.00	143.90	-33.54	60.00	1.86	91.407
10	-88.13	400.00	-100.00	-66.67	-40.00	76.87	-33.33	21.25	163.738
11	-89.17	66.67	-94.10	-40.00	-40.00	-39.69	20.00	-30.90	53.107
12	2.26	100.00	-99.77	-60.00	50.00	210.67	200.00	57.59	111.624
13	-2.63	66.67	-99.79	-25.00	-50.00	-24.87	-40.00	-25.09	46.824
14	246.79	150.00	-99.99	-60.00	50.00	-46.93	60.00	42.84	114.928
15	-77.02	25.00	-100.00	50.00	-50.00	45.00	100.00	-1.00	69.256
16	-62.90	0.00	-100.00	53.85	-40.00	974.40	50.00	125.05	350.704
17	-97.69	66.67	-100.00	-66.67	-25.00	-62.76	-50.00	-47.92	52.673
18	-57.48	-100.00	-100.00	-40.00	-66.67	25.77	60.00	-39.77	56.747
19	40.54	48.15	-100.00	50.00	-75.00	-20.29	33.33	-3.32	57.980
20	-88.90	66.67	-100.00	0.00	0.00	24.33	-75.00	-24.70	58.912
21	-89.21	66.67	-99.98	-33.33	-60.00	56.56	500.00	48.67	194.101
22	-52.09	-100.00	19.80	-60.00	-25.00	12.14	-33.33	-34.07	38.663
23	28.31	0.00	-99.98	-80.00	2400.00	-3.82	-66.67	311.12	853.906
24	-26.27	400.00	-100.00	-75.00	-50.00	-14.07	100.00	33.52	160.867
25	-95.07	25.00	-100.00	-25.00	20.00	2.11	-20.00	-27.57	47.469

Attribute (1): Markup Estiamte (%).
Attribute (2): Win/Loose.
Attribute (3): Difference ($).
Attribute (4): Potential for C.O.'s.
Attribute (5): Potential for calims.
Attribute (6): Duration extention.
Attribute (7): Project profitability.

ATTRIBUTES' ERROR ANALYSIS:

At.	MEAN-ERROR (%)	ST. DEV.	MAXIMUM-(EX.)	MINIMUM-(EX.)	RANGE
1	-48.56	72.000	246.7909 (14)	-98.5907 (3)	345.3816
2	87.59	132.438	400.0000 (6)	-100.0000 (3)	500.0000
3	-77.96	40.275	19.8004 (22)	-100.0000 (8)	119.8004
4	-8.71	74.385	200.0000 (3)	-80.0000 (23)	280.0000
5	78.43	476.457	2400.0000 (23)	-75.0000 (19)	2475.0000
6	75.98	222.800	974.4040 (16)	-62.7594 (17)	1037.1633
7	27.27	118.495	500.0000 (21)	-80.0000 (1)	580.0000

GLOBAL ERROR ANALYSIS:

Average error (%): 19.15 St. Deviation :222.601
Maximum error is :2400.0000 Occured at Ex.: 23 - Attr.: 5
Minimum error is :-100.0000 Occured at Ex.: 3 - Attr.: 2
Variation Range :2500.0000

Case (c): Single-network model - network (30 30 7) - Genetic Algorithms.

c.1 Training Results.

Neural Network Results for colct

Ex. #	ERRORS (%) in OUTPUT attributes (1)	(2)	(3)	(4)	(5)	(6)	(7)	STATISTICS MEAN,	ST.DEV.
1	-60.17	0.00	-99.99	0.00	0.00	-11.14	0.00	-24.47	36.938
2	-9.94	0.00	0.06	0.00	-50.00	-13.41	0.00	-10.47	16.950
3	-51.76	0.00	0.03	0.00	-33.33	-10.93	0.00	-13.71	19.248
4	-28.75	0.00	0.76	0.00	-50.00	-8.41	0.00	-12.34	18.250
5	-66.18	0.00	0.00	100.00	-33.33	-7.82	100.00	13.24	58.949
6	1.67	0.00	-100.00	0.00	-33.33	-25.99	33.33	-17.76	39.083
7	-13.15	0.00	0.00	0.00	-33.33	-10.04	50.00	-0.93	23.539
8	-35.03	0.00	0.00	0.00	-33.33	-7.59	33.33	-6.09	21.540
9	5.19	0.00	-100.00	0.00	-33.33	-12.46	50.00	-12.94	42.461
10	-46.76	0.00	-100.00	50.00	-33.33	-8.36	100.00	-5.49	60.541
11	-54.82	0.00	9.24	0.00	-33.33	-20.45	100.00	0.09	45.701
12	49.16	0.00	0.04	0.00	-33.33	-16.33	33.33	4.70	26.054
13	-70.09	0.00	0.00	0.00	-33.33	-10.31	100.00	-1.96	47.962
14	7.11	0.00	0.00	0.00	-33.33	-10.04	100.00	9.11	39.064
15	2.01	0.00	-66.97	0.00	-33.33	-18.03	50.00	-9.48	33.374
16	-22.58	0.00	-95.06	0.00	-33.33	-8.61	50.00	-15.65	40.559
17	7.76	0.00	-9.57	0.00	-33.33	-4.29	0.00	-5.63	12.301
18	-32.22	0.00	-99.99	0.00	-33.33	-9.24	33.33	-20.21	38.704
19	-40.22	0.00	-100.00	0.00	-33.33	-20.42	100.00	-13.43	55.914
20	-48.31	0.00	0.29	0.00	-33.33	-17.49	100.00	0.17	44.321
21	-42.53	0.00	0.00	0.00	-33.33	-11.34	100.00	1.83	43.137
22	-1.17	0.00	-96.44	0.00	-33.33	-14.40	100.00	-6.48	53.888
23	-15.16	0.00	-99.94	0.00	-50.00	-9.73	33.33	-20.21	39.756
24	5.21	0.00	-100.00	0.00	-33.33	-9.56	33.33	-14.91	39.251
25	19.50	0.00	0.00	0.00	0.00	2.56	33.33	7.91	12.303
26	-73.55	0.00	0.00	0.00	-33.33	-2.02	50.00	-8.41	34.905
27	-17.28	0.00	-100.00	0.00	-50.00	3.08	100.00	-9.17	56.340
28	-1.49	0.00	0.00	0.00	-66.67	-0.03	33.33	-4.98	27.734
29	-45.06	0.00	0.01	0.00	-33.33	-4.74	50.00	-4.73	28.048
30	-19.23	0.00	0.01	0.00	-33.33	-8.45	0.00	-8.72	12.075
31	-22.08	0.00	63.23	0.00	-33.33	-17.92	0.00	-1.44	29.041
32	-32.84	0.00	-35.27	0.00	-50.00	-17.75	-50.00	-26.55	19.649
33	-36.67	0.00	4.88	0.00	-66.67	-16.46	0.00	-16.42	24.487
34	-4.99	0.00	-46.83	0.00	0.00	-35.22	-50.00	-19.58	21.631
35	-44.95	0.00	-18.66	-50.00	-50.00	-18.14	0.00	-25.97	20.628
36	28.93	0.00	0.41	0.00	0.00	-17.58	0.00	1.68	12.684
37	-59.66	0.00	0.00	-50.00	0.00	-5.87	33.33	-11.74	29.856
38	-24.10	0.00	0.04	0.00	0.00	-2.00	0.00	-3.72	8.348
39	-25.99	0.00	0.01	0.00	-50.00	2.44	50.00	-3.36	28.290
40	7.07	0.00	0.00	-50.00	0.00	1.67	50.00	1.25	26.838
41	-34.39	0.00	-100.00	-50.00	0.00	2.08	100.00	-11.76	56.970
42	-66.99	0.00	-0.00	0.00	0.00	3.43	100.00	5.21	45.213
43	-20.41	0.00	-96.66	0.00	0.00	4.09	33.33	-11.38	37.760
44	-69.53	0.00	0.00	0.00	0.00	-8.38	100.00	3.16	46.035
45	-60.53	0.00	0.00	0.00	-33.33	-6.26	100.00	-0.02	46.003
46	-73.18	0.00	0.00	50.00	-33.33	-14.96	50.00	-3.07	40.752
47	-47.13	0.00	0.00	0.00	-50.00	-9.81	0.00	-15.28	21.326
48	-23.57	0.00	-100.00	50.00	0.00	-9.29	100.00	2.45	57.446
49	-43.17	0.00	-99.99	50.00	0.00	-12.19	50.00	-7.91	48.656
50	-26.01	0.00	1.33	50.00	-33.33	-17.84	33.33	1.07	28.563

Ex. #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	MEAN,	ST.DEV.
	.... ERRORS (%) in OUTPUT attributes.......							.. STATISTICS ..	
51	-54.25	0.00	0.01	50.00	0.00	-6.84	50.00	5.56	33.324
52	-46.68	0.00	-98.44	50.00	-50.00	-12.31	0.00	-22.49	43.884
53	-46.66	0.00	0.00	100.00	-50.00	-7.37	0.00	-0.57	45.871
54	-16.84	0.00	0.00	0.00	-33.33	-10.70	0.00	-8.70	11.831
55	-20.52	0.00	-0.57	50.00	-33.33	-9.21	0.00	-1.95	24.175
56	-14.10	0.00	0.75	0.00	-33.33	-14.24	0.00	-8.70	11.850
57	-49.49	0.00	2.13	0.00	-50.00	-17.26	0.00	-16.37	21.959
58	120.00	0.00	0.00	0.00	0.00	-20.21	0.00	14.26	43.729
59	-29.64	0.00	1.39	0.00	-33.33	-19.19	0.00	-11.54	14.281
60	-12.55	0.00	1.69	0.00	-50.00	-17.23	0.00	-11.16	17.263
61	-43.94	0.00	0.10	0.00	-33.33	-12.78	0.00	-12.85	17.110
62	-26.76	0.00	0.06	0.00	-33.33	-14.37	0.00	-10.63	13.324
63	17.05	0.00	0.00	0.00	0.00	-5.82	0.00	1.60	6.617
64	-11.68	0.00	0.05	0.00	-33.33	-5.44	0.00	-7.20	11.429
65	36.50	0.00	-0.25	0.00	0.00	-5.21	0.00	4.44	13.212

Attribute (1): Markup Estiamte (%). Attribute (2): Win/Loose.
Attribute (3): Difference ($). Attribute (4): Potential for C.O.'s.
Attribute (5): Potential for calims. Attribute (6): Duration extention.
Attribute (7): Project profitability.

ATTRIBUTES' ERROR ANALYSIS:

At.	MEAN-ERROR (%)	ST. DEV.	MAXIMUM-(EX.)	MINIMUM-(EX.)	RANGE
1	-24.27	32.781	120.0000 (58)	-73.5467 (26)	193.5467
2	0.00	0.000	0.0000 (1)	0.0000 (1)	0.0000
3	-25.82	44.117	63.2325 (31)	-100.0000 (6)	163.2325
4	6.15	27.044	100.0000 (5)	-50.0000 (35)	150.0000
5	-28.72	18.835	0.0000 (1)	-66.6667 (28)	66.6667
6	-10.46	7.428	4.0850 (43)	-35.2176 (34)	39.3026
7	35.64	41.538	100.0000 (5)	-50.0000 (32)	150.0000

GLOBAL ERROR ANALYSIS:

Average error (%): -6.78 St. Deviation : 35.983
Maximum error is : 120.0000 Occured at Ex.: 58 - Attr.: 1
Minimum error is :-100.0000 Occured at Ex.: 6 - Attr.: 3
Variation Range : 220.0000

c.2 Testing results.

Neural Network Results for test-c

Ex. #	(1)	(2)	(3)	(4)	(5)	(6)	(7)	MEAN,	ST.DEV.
	.... ERRORS (%) in OUTPUT attributes.......							.. STATISTICS ..	
1	-11.71	0.00	7.05	0.00	0.00	-11.63	-66.67	-11.85	23.248
2	-21.03	0.00	0.00	0.00	100.00	-10.91	-66.67	0.20	46.299
3	-11.38	0.00	0.00	-50.00	-50.00	-3.22	-66.67	-25.90	26.432
4	22.73	0.00	-91.51	-50.00	-66.67	-26.66	-50.00	-37.44	36.313
5	41.32	0.00	-99.96	0.00	-33.33	-10.85	0.00	-14.69	40.388
6	-26.53	0.00	-99.99	0.00	-33.33	-13.95	33.33	-20.07	38.359
7	-36.17	0.00	-99.98	-50.00	-50.00	-10.00	50.00	-28.02	43.765

Attribute (1): Markup Estiamte (%). Attribute (2): Win/Loose.
Attribute (3): Difference ($). Attribute (4): Potential for C.O.'s.
Attribute (5): Potential for calims. Attribute (6): Duration extention.
Attribute (7): Project profitability.

ATTRIBUTES' ERROR ANALYSIS:

At.	MEAN-ERROR (%)	ST. DEV.	MAXIMUM-(EX.)	MINIMUM-(EX.)	RANGE
1	-6.11	25.866	41.3175 (5)	-36.1750 (7)	77.4925
2	0.00	0.000	0.0000 (1)	0.0000 (1)	0.0000
3	-54.91	49.716	7.0509 (1)	-99.9900 (6)	107.0409
4	-21.43	24.744	0.0000 (1)	-50.0000 (3)	50.0000
5	-19.05	52.273	100.0000 (2)	-66.6667 (4)	166.6667
6	-12.46	6.559	-3.2174 (3)	-26.6636 (4)	23.4462
7	-23.81	47.020	50.0000 (7)	-66.6667 (1)	116.6667

GLOBAL ERROR ANALYSIS:

Average error (%): -19.68 St. Deviation : 38.959
Maximum error is : 100.0000 Occured at Ex.: 2 - Attr.: 5
Minimum error is : -99.9900 Occured at Ex.: 6 - Attr.: 3
Variation Range : 199.9900

Weighted error = .3608808

Case (d): Single-network model - (30 30 7) - Average of Hand crafting and Genetic Algorithms.

d.1 Training Results.

Neural Network Results for colct

Ex.	.... ERRORS (%) in OUTPUT attributes.......							.. STATISTICS ..	
#	(1)	(2)	(3)	(4)	(5)	(6)	(7)	MEAN,	ST.DEV.
1	-30.16	0.00	-49.54	0.00	0.00	-5.55	0.00	-12.18	18.348
2	-4.99	0.00	0.13	0.00	0.00	-6.76	0.00	-1.66	2.709
3	-25.97	0.00	0.12	0.00	0.00	-5.45	0.00	-4.47	8.977
4	-14.64	0.00	0.38	0.00	0.00	-4.24	0.00	-2.64	5.121
5	-33.32	0.00	0.01	0.00	-33.33	-3.93	0.00	-10.08	14.762
6	1.05	0.00	-49.90	0.00	-33.33	-13.01	0.00	-13.60	18.812
7	-6.88	0.00	0.04	0.00	0.00	-5.04	0.00	-1.70	2.741
8	-18.36	0.00	0.06	0.00	-33.33	-3.75	33.33	-3.15	18.913
9	2.72	0.00	-50.40	0.00	-33.33	-6.26	0.00	-12.47	19.308
10	-23.83	0.00	-50.37	0.00	-33.33	-4.15	50.00	-8.81	29.737
11	-27.52	0.00	4.86	0.00	-33.33	-10.20	0.00	-9.45	13.991
12	25.57	0.00	0.20	0.00	-33.33	-8.13	33.33	2.52	20.347
13	-35.29	0.00	0.02	0.00	-33.33	-5.11	50.00	-3.39	26.193
14	3.25	0.00	0.01	0.00	-33.33	-4.90	50.00	2.15	22.720
15	0.38	0.00	-51.35	0.00	0.00	-8.92	50.00	-1.41	27.263
16	-11.43	0.00	-48.52	0.00	-33.33	-4.29	50.00	-6.80	28.763
17	3.71	0.00	-9.34	0.00	0.00	-2.07	0.00	-1.10	3.718
18	-16.48	0.00	-50.10	0.00	-33.33	-4.60	33.33	-10.17	24.787
19	-20.15	0.00	-79.77	0.00	-33.33	-10.14	50.00	-13.34	36.326
20	-24.31	0.00	0.14	0.00	-33.33	-8.69	0.00	-9.45	12.827
21	-21.17	0.00	0.07	0.00	-33.33	-5.61	0.00	-8.58	12.396
22	-0.61	0.00	-48.15	0.00	0.00	-7.15	50.00	-0.84	26.362
23	-7.63	0.00	-51.27	0.00	0.00	-4.86	33.33	-4.35	22.957
24	2.96	0.00	-49.88	0.00	0.00	-4.76	0.00	-7.38	17.475
25	9.19	0.00	0.08	0.00	0.00	1.27	33.33	6.27	11.478
26	-37.07	0.00	0.03	0.00	-33.33	-1.05	50.00	-3.06	26.514
27	-8.76	0.00	-50.17	0.00	0.00	1.52	0.00	-8.20	17.425
28	-1.52	0.00	0.07	0.00	-33.33	-0.01	0.00	-4.97	11.591
29	-22.63	0.00	0.13	0.00	-33.33	-2.38	0.00	-8.32	12.788
30	-9.71	0.00	0.01	0.00	0.00	-4.24	0.00	-1.99	3.473
31	-11.08	0.00	31.40	0.00	-33.33	-8.92	0.00	-3.13	17.852
32	-16.35	0.00	-21.00	0.00	0.00	-8.89	0.00	-6.61	8.297
33	-18.36	0.00	2.37	0.00	-33.33	-8.26	0.00	-8.23	12.237
34	-2.52	0.00	-23.44	0.00	0.00	-17.62	0.00	-6.23	9.220
35	-22.76	0.00	-9.75	0.00	0.00	-9.04	0.00	-5.94	8.001
36	14.01	0.00	0.39	0.00	0.00	-8.81	0.00	0.80	6.206
37	-30.14	0.00	0.07	0.00	0.00	-2.93	0.00	-4.72	10.430
38	-12.22	0.00	0.02	0.00	0.00	-0.98	0.00	-1.88	4.235
39	-12.93	0.00	0.00	0.00	0.00	1.26	0.00	-1.67	4.621
40	3.56	0.00	0.00	0.00	0.00	0.87	50.00	7.78	17.280
41	-17.13	0.00	-50.19	0.00	0.00	1.12	50.00	-2.31	27.456
42	-33.50	0.00	-0.02	0.00	0.00	1.77	100.00	9.75	38.655
43	-10.28	0.00	-48.10	0.00	0.00	2.09	33.33	-3.28	22.232
44	-34.52	0.00	0.00	0.00	0.00	-4.11	50.00	1.62	22.960
45	-30.37	0.00	0.00	0.00	-33.33	-3.06	50.00	-2.39	25.362
46	-36.54	0.00	0.18	0.00	0.00	-7.52	50.00	0.87	23.563
47	-23.57	0.00	0.07	0.00	0.00	-4.88	0.00	-4.05	8.144
48	-11.64	0.00	-49.33	0.00	0.00	-4.67	100.00	4.91	42.127
49	-21.52	0.00	-48.64	0.00	0.00	-6.08	0.00	-10.89	17.053
50	-13.07	0.00	0.70	0.00	0.00	-8.89	0.00	-3.04	5.154

Ex. #	.... ERRORS (%) in OUTPUT attributes....... (1)	(2)	(3)	(4)	(5)	(6)	(7)	.. STATISTICS .. MEAN,	ST.DEV.
51	-26.99	0.00	0.17	50.00	0.00	-3.41	0.00	2.82	21.328
52	-23.29	0.00	-49.51	50.00	0.00	-6.14	0.00	-4.13	27.805
53	-23.58	0.00	0.00	0.00	0.00	-3.68	0.00	-3.90	8.138
54	-8.59	0.00	0.00	0.00	0.00	-5.32	0.00	-1.99	3.260
55	-10.47	0.00	-0.41	0.00	0.00	-4.61	0.00	-2.21	3.719
56	-6.95	0.00	0.50	0.00	-33.33	-7.13	0.00	-6.70	11.314
57	-24.96	0.00	1.23	0.00	0.00	-8.60	0.00	-4.62	8.855
58	100.51	0.00	0.00	0.00	0.00	-10.05	0.00	12.92	35.926
59	-14.65	0.00	0.81	0.00	-33.33	-9.58	0.00	-8.11	11.698
60	-6.95	0.00	0.91	0.00	-50.00	-8.55	0.00	-9.23	17.014
61	-21.75	0.00	0.13	0.00	-33.33	-6.40	0.00	-8.76	12.464
62	-13.58	0.00	0.04	0.00	-33.33	-7.18	0.00	-7.72	11.525
63	7.71	0.00	0.00	0.00	0.00	-2.91	0.00	0.68	3.038
64	-5.85	0.00	0.03	0.00	0.00	-2.73	0.00	-1.22	2.111
65	17.96	0.00	-0.15	0.00	0.00	-2.59	0.00	2.17	6.504

Attribute (1): Markup Estiamte (%). Attribute (2): Win/Loose.
Attribute (3): Difference ($). Attribute (4): Potential for C.O.'s.
Attribute (5): Potential for calims. Attribute (6): Duration extention.
Attribute (7): Project profitability.

ATTRIBUTES' ERROR ANALYSIS:

At.	MEAN-ERROR (%)	ST. DEV.	MAXIMUM-(EX.)	MINIMUM-(EX.)	RANGE
1	-11.63	19.590	100.5117 (58)	-37.0703 (26)	137.5819
2	0.00	0.000	0.0000 (1)	0.0000 (1)	0.0000
3	-13.75	23.475	31.4025 (31)	-79.7688 (19)	111.1713
4	1.54	8.635	50.0000 (51)	0.0000 (1)	50.0000
5	-13.08	16.665	0.0000 (1)	-50.0000 (60)	50.0000
6	-5.21	3.716	2.0920 (43)	-17.6239 (34)	19.7159
7	16.15	25.482	100.0000 (42)	0.0000 (1)	100.0000

GLOBAL ERROR ANALYSIS:

Average error (%): -3.71 St. Deviation : 19.419
Maximum error is : 100.5117 Occured at Ex.: 58 - Attr.: 1
Minimum error is : -79.7688 Occured at Ex.: 19 - Attr.: 3
Variation Range : 180.2804

d.2 Testing results.

Neural Network Results for test-c

Ex. #	.... ERRORS (%) in OUTPUT attributes....... (1)	(2)	(3)	(4)	(5)	(6)	(7)	.. STATISTICS .. MEAN.	ST.DEV.
1	2.30	0.00	3.53	0.00	0.00	-4.78	-66.67	-9.37	23.512
2	-43.96	0.00	0.00	0.00	200.00	-18.07	-66.67	10.19	81.044
3	37.18	0.00	0.00	-50.00	0.00	4.01	-33.33	-6.02	26.065
4	69.11	0.00	-32.71	-50.00	-33.33	-10.83	-50.00	-15.39	38.574
5	-8.87	0.00	-86.31	0.00	-33.33	-13.11	0.00	-20.23	29.144
6	13.12	0.00	-96.15	0.00	-33.33	-9.05	0.00	-17.92	34.564
7	-37.40	0.00	-99.71	0.00	-50.00	-5.08	50.00	-20.31	43.879

Attribute (1): Markup Estiamte (%). Attribute (2): Win/Loose.
Attribute (3): Difference ($). Attribute (4): Potential for C.O.'s.
Attribute (5): Potential for calims. Attribute (6): Duration extention.
Attribute (7): Project profitability.

ATTRIBUTES' ERROR ANALYSIS:

At.	MEAN-ERROR (%)	ST. DEV.	MAXIMUM-(EX.)	MINIMUM-(EX.)	RANGE
1	4.50	37.041	69.1113 (4)	-43.9594 (2)	113.0706
2	0.00	0.000	0.0000 (1)	0.0000 (1)	0.0000
3	-44.48	44.512	3.5261 (1)	-99.7057 (7)	103.2318
4	-14.29	22.588	0.0000 (1)	-50.0000 (3)	50.0000
5	7.14	80.601	200.0000 (2)	-50.0000 (7)	250.0000
6	-8.13	6.541	4.0076 (3)	-18.0677 (2)	22.0753
7	-23.81	39.698	50.0000 (7)	-66.6667 (1)	116.6667

GLOBAL ERROR ANALYSIS:

Average error (%): -11.29 St. Deviation : 44.677
Maximum error is : 200.0000 Occured at Ex.: 2 - Attr.: 5
Minimum error is : -99.7057 Occured at Ex.: 7 - Attr.: 3
Variation Range : 299.7057

APPENDIX V: Weights and biases of markup networks.

Case (a): (30 30 7) network trained based on Hand crafting.

1.331015	-.556213	-.490623	2.359778	1.168467	2.472702	-1.983665
-.667249	1.225607	-1.735365	.095764	-.182296	.800744	-1.303399
1.52601	-1.254021	.958074	1.153337	-1.115468	-.252196	-1.419079
-.936344	-.60618	-2.729327	-.745179	.051598	-1.734166	-.817614
-.939544	1.097208	-1.499026	-3.172981	-.467908	6.177381	1.083886
-.212172	-.964469	1.569931	2.481362	-1.126309	.968566	-1.793196
.053664	-.055945	2.691343	-1.861378	-.579569	-1.383561	.171284
.782425	-4.294883	-.191	-.705326	-.489864	.635923	.344537
-.188688	-2.430994	.983876	-1.221732	-1.040896	-1.128935	-1.495311
.092386	-4.631149	-.441862	1.29447	1.774183	4.180949	-.317808
-2.906161	.393944	.602952	-1.371652	-1.030528	.493029	-1.103892
1.065669	.896123	-2.058643	-.380728	1.00429	-.710088	.655285
.603132	.367422	-1.815765	.042575	-1.111783	-1.197018	.340323
.277604	-.305524	2.819018	-1.684573	-.901606	.798056	-1.365609
-2.280711	-3.370234	.305736	.349454	-.924913	-.176588	-1.767736
1.18522	.085458	-1.619761	1.125706	.120646	.143589	.724241
-.093177	-.541085	-2.06825	-.357584	-1.361398	-.771623	-.01045
.886189	-.514563	1.252265	-.077204	.750465	1.082817	-.347257
-1.339508	-.951524	.091668	-.225425	.890273	-.252255	-.548738
.966916	.416457	-.76997	-.765211	-1.496717	.919982	.34732
-.424683	1.211578	-.497789	-1.011219	-.493799	.069463	.889429
.689124	-.282117	.164251	-.540291	-.085661	.402852	-.801665
.212458	.466782	1.471652	-1.993104	.151072	-.156392	-1.869183
-.198665	.502527	1.172552	.894935	.083524	-1.026831	.165584
.944048	-2.532531	.333186	-1.735381	.922348	.912359	-2.06259
-1.581931	-.480077	-.197184	.279553	1.966944	1.960024	-2.576428
1.420707	2.127065	-.066462	.262043	-3.863426	-.156675	1.096208
1.310212	1.496463	1.020796	1.89303	2.202811	-1.238947	.923725
-1.497626	.52946	-2.033455	-.93798	-1.415123	-3.344662	-.41538
1.140135	.846493	-2.540633	.729661	-.431047	-.981875	-.27749
1.230093	-.972329	-.002728	-.54715	2.569762	-3.57302	-.891083
-1.705462	-.012797	.831384	1.201597	-2.059456	-.857279	1.281597
3.152099	-1.808718	1.081689	-.060922	1.744864	.240277	2.076233
-1.524749	-3.255914	1.320662	.203919	-.1495	-3.114195	.491422
-1.778998	-.491633	.114678	2.178015	.290824	-1.146003	-2.169821
-1.051187	-4.249578	-1.321719	-.668492	.240842	-.250663	.107947
-.638599	-1.505816	-1.416428	-1.28474	-.223846	-.557702	.053341
-1.771635	-.187834	.113483	.606597	-.124627	-.261247	-.254469
.091862	.384989	.393219	-.985972	1.224733	.393208	.529203
-.520053	-1.784665	.516937	.236777	3.133316	-.105009	2.464895
3.011564	.633835	-3.002367	3.400175	2.055051	.383934	2.384887
1.806037	2.115594	2.000087	-.458413	1.017488	-1.080826	-2.62531
-3.1847	-3.911088	1.623947	-.222825	-2.616629	-.050302	.112689
1.614969	.747276	-.465069	3.894208	-2.760556	2.658653	-3.920029
1.448394	.603616	2.077032	.425997	-1.138224	-2.981954	1.213273
1.204456	.699066	.514088	.143626	.002344	.77838	.532612
.264199	1.372996	.161565	-.945784	-2.113333	-.622933	.487422
-3.408199	.385763	-.072145	.16051	.820342	-.005691	2.096216
-.651301	-.342817	-1.235053	-.009116	.421697	-.833427	.586485
-2.500151	-3.009156	1.784598	-.563897	1.750758	1.049625	-1.868226
-2.558865	-.562039	-.644775	-.6773	1.13552	-.755316	1.557881
-.354381	-.847165	-1.351226	-.863734	.53833	-.370259	1.475446
-.800551	1.282947	-.81357	-.011213	2.058278	2.466497	1.08694
-.255773	.463379	-.553764	-1.522055	-.79189	.390596	1.79762
-.117623	2.319538	.856743	-.113683	-.712303	-.476394	-1.012417
-1.853437	-.271072	-1.2404	-2.521326	1.245987	.137004	.615105
-.546059	1.675816	-.476792	-.855057	-.521958	1.289657	.243625
-1.67013	.862534	-1.005225	.075782	-.735416	.056374	-.160564
-.447009	-1.365012	-1.703634	-1.352797	-1.066725	-.377248	1.044169
.090759	-.126354	-.153337	-1.589341	.82744	-.66223	1.366182
.159686	-.190725	.635263	-1.200729	.369686	.439407	.142279
-.034654	.064536	.32407	1.312801	.863167	.755784	-.781756

2.429393	-1.302466	-4.371489	1.389228	-2.916261	.390429	-1.[illegible]31177
-1.549251	-.077071	-3.401626	.019491	-1.701636	-1.568947	.678524
1.742691	-.428093	1.719407	-5.091938	.639532	2.62336	.411863
-.894538	.542301	-.321371	.066175	-1.868186	1.365588	.097412
-.534738	-1.02147	.458887	.579793	-1.379687	.789688	-1.421637
-1.720181	-.381033	.343407	2.189765	.386718	-.3373	-.284164
-1.35761	.094584	-.034355	.706404	.642168	-.037309	1.203194
2.136161	-.806978	1.455994	-1.037264	-1.033105	-1.930596	.595909
-1.768622	-.262392	1.195273	-.666668	-2.216135	.430179	-1.688577
-1.266015	-1.449755	-1.21606	-2.822843	-.381008	1.18329	-1.50685
1.698772	.594507	1.504304	1.318505	-.320818	1.242771	2.063376
-1.329564	.19501	.972747	1.714482	.587236	.208919	1.87784
-.747899	-1.003831	-1.249244	-2.846122	.990562	-2.060399	1.07694
-2.383648	-1.031253	-.678868	.993862	-.871013	2.401528	1.214061
-1.351919	-1.20745	-2.60245	-.578671	-.692635	1.041946	-1.073023
.633637	-.10795	.371537	.884027	-.901963	-1.212519	-.382299
-.64007	-.05848	-.123111	-.666072	-.367242	.171972	-.768092
-.596681	-1.015864	-1.496595	-.033172	.094938	-1.270865	1.679412
-1.138593	-.932965	2.527111	2.39972	.953683	.531585	-.527257
-1.018382	.659175	-.813395	2.146905	2.19348	.624716	2.402696
-1.863407	-1.998907	-3.273862	-.565621	-3.055571	.353403	.45783
.372812	-2.139973	1.433252	-3.292531	.712197	-.766162	3.483045
-.19157	-.140956	.555892	1.170869	1.048311	-.81747	-.542559
-2.394175	-1.329364	.616855	-.612109	-.076928	-1.565067	.15587
-.26956	-1.992684	.310857	.671046	.31635	.795632	-1.340382
.309989	-1.015241	.387809	-.696144	.494448	-1.083	-.228758
2.22914	.267991	-1.25094	-.282342	-.099908	1.126893	2.365698
2.565461	-.010277	-1.655701	-2.859353	-2.715341	.571584	-.809275
-.260533	-.588943	1.280888	.603061	.862006	1.494793	-2.636132
-1.237801	-.684645	.215763	-.893726	1.005251	-1.028758	-1.359837
-2.991382	-2.193141	-.268816	1.480857	-1.493614	2.099959	.025322
-.616288	.153282	.217997	-1.20224	-1.289735	-.229194	-.5367
-1.056986	1.180681	1.468159	.181173	-.020824	.362471	-.597734
.375169	-.763605	-1.208533	1.56792	.709002	-.602484	-.071495
.456175	.276224	.441144	-.376158	.22094	1.024932	-2.457707
.453996	.36346	-.342406	2.295222	.106013	2.964532	.848958
1.81538	-1.042796	1.585328	1.924469	-1.13103	2.140713	.324172
1.195984	1.131867	.19883	-1.455444	.735806	-3.119353	1.335527
-1.138572	-1.448347	-2.092757	-.554836	-.999603	.161759	-1.407807
1.478762	-3.512478	-.26836	.218933	2.965196	-1.240528	-2.185302
.304357	.571284	3.144245	.61361	-3.555557	-.665895	-2.526247
-1.536803	.001996	.932749	.824458	2.225572	-2.057438	-.38646
1.478496	-1.657942	1.456386	-.049757	-2.492607	-1.613717	-1.668839
.264697	-1.956529	-.204664	.630588	-1.410312	-.226572	2.068042
.369741	-.232904	3.133663	-2.793328	-1.090882	1.011203	-1.435567
2.258423	-.348597	-2.539152	-2.171341	-1.781513	-1.836553	-1.546085
-1.433607	-2.226917	-.199986	.840209	.486224	-1.670745	-.916092
.76955	1.705754	-1.406381	.302093	.576389	.657562	-.642276
-.921402	.902937	2.824649	-.030057	-2.337137	1.203279	-.746556
-3.665528	.57562	-2.427361	1.75641	.256854	-2.279787	-.198875
-1.125675	.222788	-1.737888	-.862502	-.871175	1.214033	-.109549
1.876465	-1.290673	-.153194	3.423204	-.112784	.320334	-.285091
-.192133	-.568952	-1.375931	.546887	1.043898	-1.087809	-.430646
-1.437359	3.083728	1.469189	-.152699	-1.135407	-.177307	-.062508
1.522041	1.968864	1.889139	.154837	.153401	1.216735	.157052
-1.390412	-.259081	-.152148	-.081994	.823453	.234138	-2.896524
-.39965	-.251109	-.835105	.110467	-1.278322	.279089	-.248103
1.364345	-1.101195	-2.436188	-.87715	.17844	-.40983	-2.221487
1.277333	-1.943442	1.690299	-.315763	.413824	.202927	-.196708
-.044798	-.745318	-1.928789	.050383	1.278069	-.945157	1.131613
-1.460961	.257751	2.331377	-.640843	.78752	-.893869	-.183669
-.969474	-.61962	.065025	.059839	.795749	-1.916729	-.031155
1.680867	.463904	-.047523	-2.354422	-2.514755	-.643488	-.530937
-1.54675	-.069661	-1.534994	.071512	.129022	.396618	-1.708837
-.978067	.411261	.611054	.269229	-.470527	.727375	.961031
-.92032	.723932	.261498	-.061426	1.208864	-.670125	.208406
-2.176757	-1.230171	.424473	-.917736	-1.145814	-1.870139	.302945
-1.971701	.442795	-.10575	1.801484	-1.240917	-.27594	.034117
-1.048608	-1.266591	.702939	1.134101	-.763197	-1.669744	.463601

.420628	1.925059	.197851	.228133	.158268	-1.534969	-.982052
.253103	.07253	-3.538308	1.189454	-.259276	.520636	-2.763248
-.778821	.489489	.270824	-1.419213	1.109658	.379488	-.654407
-1.072769	-.212899	.834621	-1.152507	.603942	-2.430555	-.574143
.124147	.298439	.288243	-.108204	.124079	.394471	1.282149
-.376232	.82207	-.134982	-.81044	-.758325	.30142	.556316
-.197475	-1.747799	.906344	1.760594	-.043427	.975738	1.158512
4.441262	.314503	1.889651	.697699	8.708808	2.336044	4.580277
.66075	.611246	-5.629931	.33559	-7.3599	1.30419	5.233608
.590107	1.20446	-2.57871	5.238606	6.272315	-4.815401	-2.931901
2.758072	-2.826154	.225323	-1.472054	-.554413	.432129	4.287093
-2.841232	1.853769	-.910213	-4.193786	3.304947	5.407819	.075923
-1.254114	-2.262826	-.702906	-2.575352	.545381	.245468	-.398981
.551961	-.802254	-2.130886	-2.363508	-5.98514	-.756222	1.481678
-2.904089	.750035	-1.945033	-4.358785	-3.779021	-.415842	-1.398608
1.079481	-1.927802	-.273542	.307133	-.67736	.115684	.371351
.56642	-.641992	1.062149	1.095425	.553888	1.579934	-1.546425
-.475492	1.849439	-.026229	-.090694	.884546	-.813579	-.517926
.654927	-1.018782	.704615	-.707086	-1.189985	.676553	-2.030867
.573557	-1.457996	.285331	-.970244	-.720896	.944526	.634706
2.062165	.004604	.537796	-.64152	-.065698	1.649303	.742523
-1.249561	.432291	-.749425	-.066588	.033103	2.952914	-1.291012
.873685	.159663	.63838	-.89957	-.31696	.498829	-.811687
-1.752168	.351296	-.801255	-.084889	2.255187	1.705897	-.884675
.340766	-.151077	.251226	.212541	-1.012541	-.149427	-.188936
.013491	-.650309	.241241	.366499	.04664	.088113	.391126
-.253986	.125143	-.220867	-.296042	-.330994	-.267444	-.134415
-.34339	-.105348	.488221	.260615	.615598	.044136	-.562004
-.37772	.748362	.290286	1.037776	-.636084	-1.200886	.714286
-.892625	-.657715	.432028	-.364143	-.954693	.614331	-.114016
.500437	1.768621	.885792	2.107244	.299372	-1.632945	-1.886085
.468822	.873464	.055974	.288714	-.027662	-.394714	-1.288285
-1.372475	-.182735	.693341	-.824388	1.34306	-.558538	

Case (b): (30 30 7) network trained based on Genetic Algorithms.

-1.969185	-4.176526	-5.341074	-.547657	1.593931	8.066082	-2.555077
-3.636892	5.115227	-.484823	4.321731	4.467312	1.061678	-3.084365
5.695858	-6.088528	5.051897	1.784426	-1.251293	-3.547814	-1.466403
-3.324861	.476054	-4.043254	-3.055845	-.716164	-1.111426	-1.653072
6.268697	-4.445659	-3.424141	-5.082867	-2.346463	8.937734	-3.957169
-1.924604	-3.465438	1.908196	2.017196	-1.553789	1.992327	-4.909778
-.566299	-1.397615	3.702519	-4.338675	-5.09969	-4.069905	-1.17331
-.445504	-7.073601	-2.423685	-1.715032	-4.504192	-1.416415	-1.005938
-2.050669	-5.037906	.10471	-4.87311	1.799217	-3.96244	-2.108956
1.808013	-12.74346	-3.853451	3.343736	3.835183	8.139953	4.394859
-3.471188	1.059343	2.770686	1.656968	-.532204	3.695972	-.633486
4.37552	7.534591	-2.910826	-.919807	1.816153	.197738	.466118
3.454162	2.978616	2.351573	1.798226	1.614014	-1.802794	5.401123
-.847886	3.188609	10.12063	-3.016321	-2.237771	5.991438	-7.657878
-4.026672	-2.418916	5.936629	-.054374	-5.8475	-3.370342	-5.338723
-3.150161	-4.253559	-4.000535	1.25317	-2.141654	.305752	3.016606
-1.512413	3.894367	-8.698733	-1.150655	-7.629331	.55078	-1.294036
2.180089	1.267451	-.350241	-3.011635	.482782	-6.592844	-6.868747
-8.582134	-5.58119	-7.336302	-3.487334	5.88795	-9.431079	5.462282
1.144426	5.187143	-4.406417	-.969546	2.229138	10.30434	-.973968
.083756	2.182986	2.721476	3.958149	4.019555	10.04603	-6.015446
.330559	-4.264158	-8.412618	-2.378631	-4.727689	-2.138766	-6.370122
-.038831	.484167	1.535848	.813194	8.983171	1.872807	-5.070398
4.329708	3.228173	2.296929	3.96728	1.760601	1.698816	5.674325
-4.237861	-3.701465	-.099065	-2.003131	7.071147	-.445017	-4.442353
-5.560038	.766506	.51947	.054157	5.230216	4.542935	-2.827525
-.721079	3.365447	-1.604783	2.166457	-11.9289	-3.922826	4.23804
-.055423	-1.097004	1.804178	6.235057	9.49988	-6.193199	1.640126
-6.403955	2.244953	-3.808555	-2.871084	-7.546527	-9.205441	-1.40585
2.510711	1.137066	-11.36796	-1.27034	-3.503491	-.03694	2.134688
4.872579	1.515981	-5.837242	-2.699738	2.210748	-14.406	-7.064112
-6.418717	-4.436183	7.288825	1.020087	-1.474664	-5.094157	2.411021
8.017383	-.785358	4.924424	-.432884	6.152433	.236905	5.81841
-4.96998	-10.50269	-.896365	1.267581	.257032	-9.41903	-.589535
-4.569901	-3.423388	.382947	2.450612	.487224	-4.038153	1.214539
-.228496	-9.868256	-3.008257	2.721018	4.033623	-8.49006	5.679558
3.480325	.580571	1.443091	-.423131	-4.707721	-4.059383	1.407024
-6.018366	-4.726287	-2.347062	2.874528	9.786039	-2.078291	-11.31146
-1.736871	8.321681	7.276998	.627523	11.66967	6.613665	-1.559517
-6.604584	2.136447	4.037733	3.282849	-2.647497	1.209599	4.672249
3.623384	2.31412	-.084208	3.854589	3.748847	1.65631	4.490122
4.139971	5.012671	4.982064	.401298	1.691982	-.221718	-.652851
-2.298618	-3.688817	2.484816	.556276	-1.569661	2.708632	2.268479
2.307984	2.593986	1.848695	7.16394	-1.424538	5.223494	-2.664704
2.996619	2.617699	-2.307516	3.337375	-4.797431	-5.459437	3.575204
-.359757	1.132122	-4.022599	-6.504344	4.462515	-1.063855	3.526998
3.442838	5.389279	-.913882	3.200577	-4.855936	-3.627662	.04198
-11.36887	4.621578	.325777	-1.254404	2.079718	.796546	3.961403
-4.914181	-2.095297	.764982	7.750411	1.952032	-4.758925	3.278466
-2.59184	-4.411992	3.105489	.698455	-.176094	2.68971	-.674538
-2.995125	.827106	-.479186	-2.32179	1.446163	-1.754647	.559047
-2.353673	-2.846959	-2.35109	-1.863169	-1.461053	-1.369973	.476211
-1.799712	-.715928	-1.812497	-1.010234	1.058653	1.467218	-.912431
-2.25473	-.536621	-1.553764	-2.522055	-2.79189	-1.609404	-.20238
-2.117623	.319538	-1.143257	-1.113683	-2.712303	-1.476394	-2.012417
-3.853437	-2.271072	-2.2404	-4.521326	-.754013	-.862996	-.384895
-2.546059	.675816	-1.476792	-1.855057	-1.521958	.289657	-.756375
-3.67013	-.137466	-3.005225	-.924218	-2.735416	-.943626	-2.160564
-2.447009	-3.365012	-2.703634	-3.352797	-2.066725	-1.377248	-.955831
-1.909241	-2.126354	-2.153337	-2.589341	-1.17256	-2.66223	-.633818
-1.840314	-1.190725	-1.364737	-3.200729	-1.630314	-.560593	-.857721
-2.034654	-1.935464	-1.67593	.312801	-.136833	-.244216	-1.781756
1.429393	-3.302466	-5.371489	.389228	-4.916261	-.609571	-2.931177
-2.549251	-1.077071	-4.401626	-.980509	-3.701636	-2.568947	-1.321476
-.257309	-1.428093	.719407	-6.091938	-.360468	1.62336	-.588137

-2.894538	-.457699	-1.321371	-1.933825	-3.868186	.365588	-.902588
-2.534738	-2.02147	-1.541113	-.420207	-3.379687	-1.210312	-2.421637
-3.720181	-2.381033	-.656593	.189765	-1.613282	-1.3373	-2.284164
-3.35761	-.905416	-2.034355	-.293596	-1.357832	-2.037309	-.796806
.136161	-1.806978	-.544006	-2.037264	-3.033105	-2.930596	-1.404091
-3.768622	-1.262392	.195273	-1.666668	-4.216135	-1.569821	-3.688577
-3.266015	-3.449755	-2.21606	-3.822843	-1.381008	.18329	-3.50685
.698772	-1.405493	.504304	.318505	-1.320818	-.757229	1.063376
-3.329564	-.80499	-.027236	.714482	-.412764	-.791081	.87784
-1.747899	-3.003831	-2.249244	-3.846122	-1.009438	-4.060399	.07694
-4.383648	-3.031253	-1.678868	-1.006138	-1.871013	.401528	-.785939
-2.351919	-2.20745	-3.60245	-1.678671	-1.692635	.041946	-3.073023
-.366363	-2.10795	-1.628463	-1.115973	-2.901963	-3.212519	-1.382299
-1.64007	-1.05848	-2.123111	-2.666072	-2.367242	-.828028	-2.768092
-2.596681	-2.015864	-3.496595	-2.033172	.978162	1.606381	2.094373
-1.055014	-2.205147	10.01198	5.396251	3.508372	3.2624	.082907
-.819501	1.709639	.413385	3.567717	4.205142	-.519586	5.432617
-.749098	-2.885177	-7.865818	1.572053	-4.596139	3.416451	2.572718
3.273478	-4.22921	7.866528	-4.170886	1.991863	-3.58831	6.120948
2.289007	-2.197576	2.439415	2.019177	2.777265	-5.911504	-1.562857
-6.075902	-6.413908	-.81397	.028097	1.051514	3.257135	1.780802
-5.606196	-.447006	.160362	3.361848	.375704	-3.049593	.193007
-4.809656	2.181483	.789054	.271148	1.318102	7.776166	-6.639746
-2.143137	2.748231	4.588748	-2.640188	5.32563	4.711346	-5.733386
-13.76148	7.527771	4.319958	6.930849	6.642532	1.767699	5.4821
3.482289	.791192	1.013252	1.693817	-5.757227	.58742	-3.347973
-1.953415	-1.504194	-.35152	-.549048	2.442409	-.38758	.5818
-1.371369	4.460822	.981291	-2.217453	-5.762932	4.305124	3.374729
4.302258	-.479131	-1.572159	-1.179362	3.146736	.170273	-4.246901
.231131	2.842656	-1.882909	-1.607657	.324342	.738761	1.639106
-2.290405	4.365641	-3.085327	1.049658	3.051589	3.14729	-.631576
2.710001	-3.646482	-1.141197	-4.080925	-.330114	.844971	-7.143452
2.590544	-3.255225	-1.368854	1.666976	.203821	-.477195	-2.11513
4.061327	-1.560205	-5.720005	-1.115189	4.350337	.655922	.581692
-.769334	-.071134	-4.51972	.587737	-2.248299	-7.13668	2.273346
-2.889975	-.516131	-2.54192	7.24762	-1.632206	.067387	-3.749014
8.044986	-3.349811	3.815588	-4.526849	7.815166	2.711226	3.358284
1.849931	4.735991	.222097	-.159622	.040802	1.908829	1.021655
1.05397	2.026014	2.889474	.422897	.91543	-2.082013	.329647
-1.282334	-5.440104	1.62797	.758755	1.056942	-2.465285	4.650715
.884045	-.760687	-.268183	2.125515	2.857195	3.279014	-1.034954
.544036	2.919564	-.402326	-.49186	-.493238	.843692	.920567
-.2920[illegible]	2.704364	.757617	-1.738025	-.846981	-.806554	-1.705179
-.350134	-.657455	.133132	-.783627	-1.877967	-1.986904	-2.746785
-.668843	-.293718	-.550434	-.9852	-1.838613	-3.360619	.572479
-.746278	.630877	1.063963	.548259	-.293077	2.212334	-1.898007
-3.838462	-7.420074	-4.155664	9.468699	-1.543461	-1.963364	-1.924736
-2.6266	-4.741459	-6.422688	-4.49013	2.684515	1.913349	-2.351541
8.09464	-2.571353	-1.990613	13.4044	6.644071	-1.853279	-.018929
.434666	.078178	-3.246123	4.337408	2.925452	-3.691183	1.229802
-.549397	2.936914	-2.944978	3.012797	1.913142	1.261716	-2.093122
1.282926	3.620796	2.549658	-.934676	1.793216	-.174155	-.433125
-4.016903	-.155147	-1.308838	-3.950042	3.011713	-1.131137	1.434805
1.033626	3.180437	-4.800977	-.417328	-1.509239	3.572187	-1.181372
4.340213	-4.880956	-1.496721	.335065	-.82156	-1.40983	-4.221487
-.722667	-2.943442	-.309701	-2.315763	-.586176	-1.797073	-2.196708
-1.044798	-2.745318	-2.928789	-1.949617	-.721931	-1.945157	-.868387
-3.460961	-1.742249	1.331377	-1.640843	-1.21248	-2.893869	-1.183669
-1.969474	-1.61962	-.934975	-1.940161	-.204251	-2.916729	-1.031155
.680867	-1.536096	-2.047523	-3.354422	-3.514755	-2.643488	-1.530937
-2.54675	-2.069661	-2.534994	-.928488	-.870978	-.603382	-2.708837
-1.978067	-1.588739	-1.388946	-1.730771	-2.470527	-1.272625	-1.038969
-2.92032	-1.276068	-1.738502	-2.061426	.208864	-2.670125	-.791594
-3.176757	-3.230171	-1.575527	-3.145894	-2.165046	-2.856927	-1.188477
-3.891167	-.416091	-1.024506	-.148906	-3.037895	-.990819	-1.695296
-2.761699	-2.01395	1.495233	2.386006	.037448	-2.319083	.290329
.978555	2.587537	.526794	.616015	1.026531	-.899455	-.814957
.601232	.373993	-3.679805	1.068376	-.985072	.746056	1.432317
.391658	-.261707	.536166	.440157	.391666	.667496	-.659033

.176999	-.127585	.513633	-1.335447	.647625	-4.415168	-.152882
.454183	-.244272	1.220799	.397786	-.67417	-.363505	.48043
-.087823	-.572231	-.238926	-.630235	-.299868	.298803	.422831
-.10343	-2.072731	-3.125787	4.578082	-11.94356	1.687069	-13.14127
11.26755	2.894962	2.628836	-5.195225	6.235946	7.978697	9.233454
1.358452	1.394084	-11.0499	.714091	-7.73163	2.731031	9.231379
3.274044	12.69322	.220174	7.644374	3.915484	-4.827797	-13.63
4.1554	-5.070262	.806673	-3.330[illegible]58	-11.43353	1.316211	9.608222
8.510884	2.083467	-4.145131	-1.285564	7.667645	10.68762	-6.908036
2.872734	1.86499	-1.649403	-5.387851	.894746	-.254035	-.345196
1.017471	-1.347409	-3.685959	-3.25181	-7.534746	-2.750983	-1.66272
-1.996642	.418992	-6.058228	2.257035	-7.451615	-.971228	-1.510297
.406164	-.779714	-1.208304	-.479345	-1.874043	-.03926	-.057013
2.189393	-1.223194	2.041439	.077092	.274705	2.067031	-1.495868
-.747316	1.005019	.420668	-.247186	.099594	-.130922	-2.174098
-.946256	-2.384426	1.238829	-.446451	-2.810476	-.189067	-1.397135
.371378	-1.225399	-.325197	1.010257	-2.587975	-.175665	-.337969
.304419	1.28377	-.579384	.645954	.823285	.64477	1.394742
-.702766	-.192035	-.998201	-.228324	.714503	2.769502	-1.19043
1.7528	-.238087	-.413818	.447325	-1.167552	.09931	-.29934
-1.45886	-.241841	-.912739	-.508601	2.488396	1.727012	-.378014
-.030513	.237469	-.104385	.450023	.050802	.123663	-.201716
.482385	-.347687	-.025435	-.052519	.104265	.332801	.226959
-.282408	-.259884	-.024363	-.272576	.080406	.057372	.187818
.36344	.007485	.03449	-.180497	.068091	.014191	-.468067
-.155651	.101539	-.188203	.726786	.181623	.764283	3.503052
-1.664334	.19008	-.645137	.745208	-.532732	.203356	-2.802401
1.000087	1.454983	1.441396	1.721487	.3192	-1.683059	-3.190531
.262166	-1.187002	.557756	.14734	2.756791	.535303	-1.438733
-.403609	-2.455005	.877864	-.164478	2.401907	.274678	

APPENDIX VI: Relative importance of markup attributes.

Case (a): (30 30 7) network trained based on Hand crafting.

Relative importance of input attributes

Input Attributes	OUTPUT ATTRIBUTES (1)	(2)	(3)	(4)	(5)	(6)	(7)
Contract type------	1.9	3.0	2.3	2.7	3.2	2.3	2.7
Owner type---------	4.5	4.6	4.5	4.8	4.9	4.3	4.2
Competition--------	4.6	4.8	5.8	4.4	4.1	5.0	4.3
Need for work------	3.5	3.7	2.8	3.1	2.9	3.2	2.8
Total cost --------	4.3	4.7	5.5	4.1	4.0	4.9	5.5
Contract duration -	4.6	4.3	4.0	4.7	4.5	4.3	3.9
Company size-------	3.9	3.4	3.7	3.6	3.8	3.6	3.9
Markup definition--	3.7	4.0	3.7	4.0	4.2	3.6	3.7
Markup components--	4.0	3.7	4.5	4.3	4.2	4.0	3.5
Unc. due to Site co	4.1	4.6	3.3	4.4	4.6	4.6	4.8
Unc. due to Owner A	4.7	3.8	4.0	4.1	4.0	4.4	4.2
Unc. due to Locatio	4.4	2.8	4.1	4.0	3.9	3.8	4.0
Unc. due to Safety	4.4	4.4	5.0	5.0	5.3	4.5	5.1
Unc. due to Inaccur	3.1	3.1	3.2	3.1	2.8	2.6	2.9
Unc. due to weather	3.8	3.3	2.2	3.6	2.6	3.2	3.3
Complex. due to Tec	1.2	1.9	1.5	1.4	1.7	1.4	1.7
Complex. due to Res	2.5	2.7	2.9	2.8	3.4	3.3	3.4
Complex. due to Job	2.2	2.7	2.6	2.5	2.2	3.0	2.7
Quality of Drawings	4.2	3.2	3.4	3.5	3.0	3.6	3.3
Stacking of Trades.	3.6	3.3	3.5	3.0	3.2	3.6	3.6
% Subcontracted.---	3.0	3.2	3.5	3.0	2.8	2.8	2.8
Rigidity in Specs.-	4.0	4.0	4.1	3.7	3.0	3.9	3.7
Change in Inflation	3.4	3.9	3.0	3.8	2.7	3.7	4.0
Change in Escalatio	2.0	2.3	1.8	1.9	2.5	2.0	2.0
Economic growth.---	2.8	2.4	2.2	2.3	2.5	2.4	2.0
Resources available	2.3	2.2	2.4	2.5	3.0	2.2	2.6
Firm Expertise.----	2.2	1.9	1.8	2.2	2.2	2.3	2.0
Management Experien	2.7	1.9	1.9	2.2	3.0	2.7	2.3
Confidence in Work	2.3	2.5	3.5	2.4	2.3	2.2	2.2
Financial Capabilit	2.3	3.7	2.9	2.8	3.4	2.6	3.0

Attribute (1): Markup Estiamte (%). Attribute (2): Win/Loose.
Attribute (3): Difference ($). Attribute (4): Potential for C.O.'s.
Attribute (5): Potential for calims. Attribute (6): Duration extention.
Attribute (7): Project profitability.

Case (b): (30 30 7) network trained based on Genetic Algorithms.

Relative importance of input at.ributes

Input Attributes	OUTPUT ATTRIBUTES (1)	(2)	(3)	(4)	(5)	(6)	(7)
Contract type------	3.0	3.4	3.1	3.0	3.7	3.1	3.3
Owner type---------	4.4	3.7	4.2	4.4	3.4	4.1	4.3
Competition--------	3.7	3.9	4.6	3.6	3.5	4.0	3.4
Need for work------	3.1	2.7	3.0	2.9	2.6	2.6	2.8
Total cost --------	4.5	5.2	6.0	4.3	4.4	5.0	4.5
Contract duration -	3.9	4.6	3.9	3.7	3.3	3.6	3.5
Company size-------	2.4	2.9	2.9	3.3	3.5	3.0	3.5
Markup definition--	2.6	3.3	3.0	3.1	3.5	2.5	3.0
Markup components--	3.5	3.9	4.6	4.4	4.8	4.5	4.4
Unc. due to Site Co	4.5	4.8	3.9	4.3	4.5	4.5	4.7
Unc. due to Owner A	3.8	3.3	3.1	2.3	3.4	2.7	2.6
Unc. due to Locatio	4.1	3.9	3.9	4.2	3.3	3.8	3.7
Unc. due to Safety	4.2	4.1	4.0	3.8	3.6	3.5	4.1
Unc. due to Inaccur	2.8	2.9	2.7	3.0	3.6	2.6	3.3
Unc. due to weather	2.7	3.1	2.8	2.9	2.9	3.5	2.9
Complex. due to Tec	3.0	2.7	3.0	2.7	2.9	2.6	2.3
Complex. due to Res	3.4	2.8	2.9	3.0	3.5	3.8	3.5
Complex. due to Job	2.7	2.6	2.3	2.2	2.0	2.6	2.3
Quality of Drawings	4.0	3.2	3.7	3.7	3.4	3.7	4.0
Stacking of Trades.	2.9	2.6	3.2	2.7	3.0	3.0	2.5
% Subcontracted.---	3.9	2.9	4.4	3.8	3.4	4.1	2.8
Rigidity in Specs.-	3.6	3.9	3.8	4.1	3.9	3.3	4.2
Change in Inflation	3.2	3.7	2.9	3.3	3.0	3.5	3.2
Change in Escalatio	2.3	2.1	1.8	2.9	2.2	2.2	2.3
Economic growth.---	3.6	3.0	2.7	2.6	2.2	2.7	2.8
Resources available	3.3	3.0	2.8	3.4	3.3	3.3	3.8
Firm Expertise.----	2.1	2.4	3.1	2.9	3.1	3.3	2.7
Management Experien	2.5	2.1	2.1	2.8	3.1	2.9	2.8
Confidence in Work	2.8	2.7	2.4	3.3	3.2	2.9	2.6
Financial Capabilit	3.5	4.6	3.3	3.3	3.6	3.0	4.2

Attribute (1): Markup Estiamte (%). Attribute (2): Win/Loose.
Attribute (3): Difference ($). Attribute (4): Potential for C.O.'s.
Attribute (5): Potential for calims. Attribute (6): Duration extention.
Attribute (7): Project profitability.