Dynamic Traffic Signal Control Using A Self-learning, Fuzzy-neural Intelligent System

by

Jian Wu

M.A.Sc., China Geoscience University, 1987

A Thesis Submitted in Partial Fullfillment of the Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Mechanical Engineering

We accept this thesis as conforming to the required standard

[redacted]

Dr. Zuomin Dong (Dept. of Mechanical Engineering), Supervisor

[redacted]

Dr. Wei-Wu Zhou (Dept. of Mechanical Engineering, Ministry of Transportation and Highways, B.C.), Co-supervisor

[redacted]

Dr. Meyer Nahon (Dept. of Mechanical Engineering), Departmental Member

[redacted]

Dr. Nikitas J. Dimopoulos (Dept. of Electrical Engineering), External Examiner

University of Victoria

Abstract

Optimal system performance in a traffic system is achieved by planning, control, and scheduling of the system's many traffic movements. This task of optimizing the time and facility conflicting activities is challenging. In the last a few decades, extensive research has been carried out on the planning and control of manufacturing processes to improve productivity and reduce manufacturing costs. In this work, we apply the quantitative intelligent system concept, developed in the optimal planning of manufacturing activities, to the dynamic signal control problem of a corridor traffic system to minimize traffic delays.

Today, the ever-increasing demand for individual mobility and reduced traffic delays, coupled with economic and environmental restrictions on increases in physical road capacity, requires efficient control of signalized intersection traffic. Present traffic signal control is based upon static traffic control strategy using several fixed timing plans. These timing plans contain optimal control parameters for representative traffic patterns manually identified by a traffic engineer. Several timing plans are applied during different time periods according to the traffic demand statistics, regardless the actual traffic flow condition at the instant. To improve the quality of signalized traffic control, and to reduce traffic delay and congestions caused by the inaccurate estimation of traffic demands, the development of a dynamic control strategy based upon the on-line acquired traffic flow condition becomes necessary.

In this work, an approach for dynamic traffic signal control, based upon a fuzzy-neural intelligent system and traffic delay minimization, is introduced. This approach consists of three major parts: (a) automated traffic flow pattern identification using fuzzy pattern clustering, (b) optimization of traffic control parameters (timing plan design) for identified traffic flow patterns, and (c) dynamic traffic signal control by real-time traffic flow monitoring, traffic flow pattern matching using the fuzzy-neural system, and execution of stored optimal control parameters.

The traffic flow pattern identification is carried out using the fuzzy pattern clustering and matching techniques. A mathematical model is first introduced to quantify the fuzzy traffic conditions. The traffic condition at a moment is expressed as a hyper point in a m-dimensional traffic parameter space. Similar traffic conditions show as clouds of hyperpoints. The quantified traffic condition description allows the "characteristic groups" of traffic flow conditions being

recognized as traffic patterns, using the fuzzy clustering methods. These traffic patterns are closely studied. The optimal timing plans of these traffic patterns, which contain the optimal signal control parameters, are generated using commercial software through extensive optimization.

Dynamic traffic control is accomplished using fuzzy traffic pattern clustering/matching methods and traffic plan optimization. The task is carried out in two steps: off-line learning and on-line control. The off-line learning part identifies all representative traffic patterns based upon previously collected traffic data, designs a timing plan for each identified traffic pattern using traffic delay minimization, and trains a fuzzy-neural system using the traffic pattern – optimal timing plan pairs generated. The on-line control part senses traffic flow in real-time, matches the sensed traffic flow condition with the best fitted traffic pattern, assigns the optimal timing plan of the matched traffic pattern to related traffic controllers dynamically. A method for short-term traffic flow condition prediction is also developed to offset the short delay in traffic flow condition sensing, and quasi-optimal traffic signal parameter updating at the controller.

The approach makes dynamic traffic signal control of a corridor traffic system with quasi-optimal performance possible. The system is self-adaptive and capable of carrying out self-learning to varying traffic demands. Computer simulation and prototype testing using the real traffic data have demonstrated significant traffic delay reduction.

The research directly contributes to static and dynamic traffic control research and practice. It also extends the research and applications of the *quantitative intelligent system approach*, and benefits the research on intelligent scheduling and planning for time and facility conflict activities. The research on developing a hybrid fuzzy-neural system combines the reasoning ability of a fuzzy system and the learning ability of a neural network, which is critical for a self-learning and self-adaptive, intelligent system.

Examiners:

[redacted]

Dr. Zuomin Dong (Dept. of Mechanical Engineering), Supervisor

[redacted]

Dr. Wei-wu Zhou (Dept. of Mechanical Engineering,
Ministry of Transportation and Highways, B.C.), Co-supervisor

[redacted]

Dr. Meyer Nahon, (Dept. of Mechanical Engineering), Departmental Member

[redacted]

Dr. Nikitas J. Dimopoulos, (Dept. of Electrical Engineering), External Examiner

Contents

Titlepage i

Abstract ii

Contents v

List of Figures ix

List of Tables x

Acknowledgements xi

Foreword xii

Dedication xiii

Chapter 1
Introduction **1**
- 1.1 General 1
- 1.2 Traffic Planning and Control Practice 2
  - 1.2.1 Terminology 3
  - 1.2.2 State of the Art of Traffic signal control 5
  - 1.2.3 Limitations of Present Traffic signal control Methods . . 10
- 1.3 The Proposed Approach for Dynamic Traffic signal control . . . 12
- 1.4 Resemblance of Manufacturing and Traffic Planning and Control 16
- 1.5 Thesis Outline 20

Chapter 2
Intelligent Dynamic Traffic Control – An Overview **22**
- 2.1 Quantitative Representation of Traffic Flow Conditions 22
- 2.2 An Overview of the Intelligent Dynamic Traffic Control System 24
  - 2.2.1 On-line Part Function Modules 24
  - 2.2.2 Off-line Part Function Module 30
  - 2.2.3 Communication between the On-line and Off-line Parts .. 32

Chapter 3
Traffic Pattern Recognition **34**
3.1 Fuzzy Set Theory and Fuzzy Clusters 35
3.2 Fuzzy c-Means (FCM) Algorithm 38
3.3 Clustering Quality Assessment 40
3.4 Primitive Traffic Pattern Identification 43
3.5 Automatic and Systematic Primitive Traffic Pattern Identification 49

Chapter 4
Design of Optimal Timing Plan **56**
4.1 Description of TRANSYT-7F 56
4.2 Traffic Flow Simulation 57
4.3 Optimization Objective Function 58
4.4 Evaluation of the Objective Function 59
4.4.1 Calculation of Traffic Delay 59
4.4.2 Calculation of Traffic Stops 64
4.4.3 Maximum Back of Queue 67
4.5 A New Traffic signal control Optimization Formulation 68

Chapter 5
Short-term Traffic Demand Prediction **70**
5.1 A Brief Overview 70
5.2 Calculation of Current Traffic Demand 73
5.3 Weighting Factors for Traffic Demand Prediction 74
5.4 Testing Results 75
5.4.1 Fixed Weighting Factor Method 75
5.4.2 Adaptive Weighting Factor Method 76

Chapter 6
Fuzzy-Neural Dynamic Traffic Decision Making **80**
6.1 An Introduction to Fuzzy-Neural Networks 80
6.1.1 Artificial Neural Networks (ANN) 80
6.1.2 Fuzzy-Neural Networks 82
6.1.3 A Taxonomy of Neural Nets Used for Pattern Classification 83
6.1.4 Continuous Input and Supervised Neural Network 86
6.2 ANN Training and The Back-propagation Algorithm 88
6.3 Traffic Flow Pattern Matching 94
6.4 Test Results 96
6.4.1 Test on Pattern Selection Consistency 96
6.4.2 Test on Dynamic Traffic Flow Variation Response 97
6.4.3 Robustness and Stability of the Approach 98

Chapter 7
Conclusions 101

Bibliography 104

Appendix A
Fuzzy Clustering Output 109

Appendix B
Trained Fuzzy-neural Network Output 113

List of Figures

1.1 Resemblance of Production and Traffic Control Problems 21

2.1 Representation of Traffic Volume Changes 25

2.2 Representation of Traffic Flow Cloud 26

2.3 Functional Modules of the Intelligent Dynamic Traffic Control System (Software) 27

2.4 Functional Modules of the Intelligent Dynamic Traffic Control Structure (Hardware) 28

3.1 Explanation of Fuzzy Clustering Quality Assessment 42

3.2 Duncan Traffic System Sketch 44

3.3 Intersection Traffic Volume at Allenby Street 45

3.4 Intersection Traffic Volume at Boys Street 46

3.5 Duncan System Northbound Traffic Volume through Five Intersections 1. Allenby; 2. Boys; 3. Trunk; 4. Coronation; 5. James 47

3.6 Duncan System Southbound Traffic Volume through Five Intersections 1. Allenby; 2. Boys; 3. Trunk; 4. Coronation; 5. James 48

3.7 Fuzzy Pattern Identification Flow Chart 52

3.8 Time Interval Partitions in Allenby 53

3.9 Time Interval Partitions in Boys 54

4.1 Typical Delay Measurement 60

4.2 Illustration of the Basic TRANSYT Definition of Uniform Delay 61

4.3 Derivation of Uniform Delay 62

4.4 Derivation of Stops and Maximum Back of Queue 65

4.5 Reduction of Stops as a Function of Length of Delay 66

4.6 Degree of Saturation 67

5.1 Prediction Algorithm Description 72

5.2 Fixed Weighting Traffic Flow Prediction Simulation I (Duncan at Boys Road., Northbound through) 77

5.3 Adaptive Weighting Traffic Flow Prediction Simulation I (Duncan at Boys Road., Northbound Through) 79

6.1 A Typical Three Layers ANN 81

6.2 Fuzzy-neural Network with Fuzzy Neurons Substituting the Weighting Elements 83

6.3 Fuzzy-neural Network Which Embed the Neural Network into Fuzzy Reasoning System 84

6.4 Computational Element or Node Which Forms a Weighted Sum of N Inputs and Passes the Result through a Nonlinearity 85

6.5 A Taxonomy of Six Types of ANNs Used as Classifier 86

6.6 Types of Decision Regions 87

6.7 Structure of A Three Layer Neural Network Structure 89

6.8 A Fuzzy-neural Network Used in Traffic Dynamic Control 96

6.9 Total Traffic Delay for Dynamic and Static Traffic Signal Control 99

List of Tables

3.1 Traffic Delay and Stops of Manual and Fuzzy Timing Plan . . . 55

5.1 Fixed Weight Factor Test 75

5.2 Prediction Errors of Historical Average and Fixed Weighting Factor Prediction (Based upon 14683 Vehicles at This Movement in A Day) 76

5.3 Comparison between Historical Average and Adaptive Weight Prediction 78

6.1 The Comparison of Convergence Results 94

6.2 Fuzziness Numbers as Output of Pattern Matching 100

Acknowledgements

During my study at the University of Victoria I had the opportunity to meet many people, whose work, lecture, knowledge, and experience have been of great benefit to me. I especially wish to thank my thesis supervisor, Dr. Zuomin Dong, for his constant support, encouragement and valuable guidance through this work, my co-supervisor, Dr. Weiwu Zhou, for his kind support, encouragement and advice on traffic control technologies. I also wish thank Dr. Meyer Nahon and Dr. Nikitas J. Dimopoulos for diligently reading my thesis and making valuable comments for the improvement of the scripts.

This work cannot be conducted without the financial and technocal support from the Ministry of Transportation and Highways of British Columbia. Many individuals at the Ministry have generously passed their knowledge and experience to me. The Ministry has also provided me the necesssay traffic data and traffic control facilities. These are invaluable to the success of this project. I especially wish to thank Mr. Edward Miska and Ms. Ann Lee for many stimulating technical discussions.

Last but certainly not least I wish to thank my wife Yanjun whose love and support kept me going during these years, and my mother, without whom none of this would have been possible.

Foreword

Dedication

To My Family

Chapter 1

Introduction

1.1 General

Today's industrial environment and increasing international competition demand improvements on the efficiency of our traffic system to keep up with the pace of economic development. The increase of road traffic demands often causes traffic congestion in urban and suburban areas worldwide, both in industrialized countries and in the metropolitan areas of developing countries. With financial and environmental constraints on building new road infrastructures, a cost-effective approach for improving the efficiency of a traffic system is to apply advanced computer technology in traffic signal control and to develop new and more effective traffic signal control and management techniques.

To accommodate the need for more efficient traffic systems, the concept of the Intelligent Vehicle and Highway System (IVHS) was introduced in late 1980's [5, 10]. The Intelligent Vehicle and Highway System consists of many subsystems using advanced computer, communications, and sensor technologies to improve the performance of transportation systems to relieve traffic congestion, improve safety, and reduce pollution and energy waste. The IVHS has four major components:

- **Advanced Traveler Information Systems (ATIS)** — providing travelers with information to ensure that their journey is as safe, efficient, comfortable, and enjoyable as possible;
- **Advanced Fleet Management Systems (AFMS)** — employing a va-

riety of technologies to improve the safety, efficiency, and control of fleet operations;

- **Advanced Traffic Management Systems (ATMS)** — combining new and existing traffic management and control system to optimize traffic flow on freeways, urban expressways, and arterial;

- **Advanced Vehicle Control Systems (AVCS)** — As the most sophisticated component of the IVHS systems, the function of AVCS can be further divided according to the periods of its development. In the near future, AVCS systems will include driver aids such as adaptive cruise control, collision and skid (traction) warning, and vehicle/driver performance monitoring. In the mid-term, vision enhancement and lane change and merge warning will be provided. Certain driving functions, such as skid control and braking for collision avoidance, may also be automated, further enhancing driver capabilities. In the long term, AVCS is expected to include fully automated highways in which the system will take over the driving task completely, on dedicated highway facilities in high demand inter city routes or metropolitan corridors;

The research conducted in this thesis focuses on the advanced traffic signal control methodology with dynamically optimized traffic signal control parameters, and contributes to the area of Advanced Traffic Management Systems of IVHS.

1.2 Traffic Planning and Control Practice

Earlier research on efficient traffic signal planning and control goes back almost half a century, but it is only in the last two decades that sophisticated and complex control algorithms have been developed and applied.

From the late 1940's to early 1960's, traffic planning focused on the expansion of roadway capacity to meet the increasing number of vehicles. Owing to the enormous growth of traffic demands, it was recognized that by simply expanding road capacity the problem would not be solved. The need for efficient traffic signal control was recognized. Environmental problems of air and noise pollution related to vehicle operation also became evident and had to be addressed. These led to the earlier traffic engineering research focusing on traffic signal designs for safe and efficient movement of people and goods using existing roads.

The present traffic signal control system consists of signs, signals, and markings. The use of traffic signals at signalized intersections (or traffic signal control), controls traffic flow go through the intersections by allocating appropriate time among conflicting traffic flow demands. The synchronized traffic signals at a series of intersections allow traffic to pass through an artery without stopping, which is called signal progression. Appropriate time offsets of these synchronized signals must be calculated in this progression system.

This research focuses on the intelligent traffic signal control in a signalized multiple intersection artery system.

1.2.1 Terminology

For clarity, various terminology on traffic signal control are given as follows:

- **cycle** — one complete sequence of signal indications.
- **cycle length** — total time for the signal to complete one cycle, given the symbol C (sec).
- **phase** — a time interval of a cycle which allows a group of traffic movements receiving the right of way simultaneously.

- **interval** — period of time during which all signal indication remain constant.

- **change interval** — the "yellow" and/or "all-red" intervals which occur at the end of a phase to provide for clearance of the intersection before conflict movements are released, given the symbol Y_i for phase i (sec).

- **green time** — time within a given phase during which the "green" indication is shown, given G_i for phase i (sec).

- **lost time** — time during which the intersection is not effectively used by any movement; these times occur during a portion of the change interval when the intersection is cleared and at the beginning of each phase as the first few cars in a standing queue experience start-up delays (sec or sec/phase).

- **effective green time** — time during which a given phase is effectively available for stable moving platoons of vehicles in the permitted movements. This is generally taken to be the green time plus the change interval minus the lost time for the designated phase, given the symbol g_i for phase i (sec).

- **offset** — the *signal offset*, simply as the offset, is defined as the difference between green initiation times during which the through-movement is permitted in a main corridor traffic system.

- **volume** — the traffic volume is defined as the number of vehicles that pass a point on a highway, or a given lane, or a direction of a highway, during a specified time interval. The unit for volume is simply "vehicles", although it is often expressed as vehicles per unit of time.

- **speed** — Speed is defined as a rate of motion, in distance per unit of time.

- **density** — Density is defined as the number of vehicles occupying a given length of highway or lane, and is generally expressed as vehicles per mile (vpm) or vehicles per mile per lane (vpmpl).
- **occupancy** — Occupancy is the time percentage of the vehicles occupying the vehicle detective loop.
- **bandwidth** — The bandwidth is the time that allows the vehicles go through given intersections without stopping.
- **Travel-time delay or total delay** — Delay can be defined in a number of ways, including stopped delay, approach delay, travel-time delay, time-in-queue delay. The travel-time delay, generally used in traffic signal control optimization, is defined as the difference between the time the vehicle clears the intersection and the time it would have done so if it had passed through the intersection at the desired speed, without stopping.

1.2.2 State of the Art of Traffic signal control

The purposes of traffic signal control include maximizing the efficiency of existing traffic systems without new road construction, maintaining safe traffic flow through the traffic system, minimizing total traffic delay and individual vehicle delay, as well as reducing air and noise pollution. The traffic signal control can be carried out in two different ways:

- **Static Traffic Signal Control** - Static traffic signal control is widely used to control traffic flow at signalized intersections. In static traffic signal control, the timing plans are pre-designed to match the traffic conditions. In most cases, the timing plans are calculated for morning and afternoon peak hour conditions and for the periods between these peaks. Plans may also be calculated for weekends and other special events.

In existing static traffic signal control methods the traffic flow data are collected from the field and then downloaded to a computer. In practice a traffic engineer usually divides the time of a day into several time periods according to traffic flow patterns based upon experience. The timing plans are usually calculated in week or season based upon the statistical averages of recorded traffic flow data. These timing plans are specified using special optimal traffic control algorithms (on the basis of maximizing the bandwidth or optimizing a combination of traffic measures of effectiveness (M.O.E.) such as delays and stops) based upon given traffic flow patterns. Various commercial software is available for the traffic flow optimization. For example:

- Bandwidth program — (e.g. SIGSET, MAXBAND, and PASSER II) These programs operate on the principle of maximizing the band width, to successfully encounter green signals when traveling along a street.
- Optimization Program — (e.g. TRANSYT, SOAP, and SIGOP II) These programs operate on the principle that the value of one or more network-wide traffic MOE (measure of effectiveness) should be optimized.

The pre-determined timing plans are later uploaded from the traffic signal control computer to the controllers to direct traffic flow. In static traffic signal control the cycle length, phase, green time, and offset are all preset and kept constant in a specific control period. The signal change repeats in a constant fashion. Depending upon the available equipment, several preset timing plans may be used for a day with different timing plans automatically changed for different periods of time [18].

- **Dynamic Traffic Signal Control** – Dynamic traffic signal control directs traffic flow based upon the actual traffic conditions, rather than the historical traffic flow data as used in static traffic signal control. The dynamic traffic signal timing forms a closed-loop control system. The traffic signal control system consists of vehicle detection, traffic flow condition prediction, control strategy design, and control strategy decision making sub-systems. As a new research area, the dynamic traffic signal control has gained more and more attention [24, 27]. Previous research on this area can be divided into two groups: the cycle-based and the non cycle-based signal timing plan method. In the cycle-based method the timing plan is re-set cycle by cycle and in the non cycle-based method the timing plan is re-set in a time interval, say 5 minute or 15 minute.

  The cycle-based signal timing plan approach dynamically changes the timing parameters by adopting different cycles. The method is only suitable for an isolated intersection. For a multiple intersection traffic system, when new timing plans of different cycles are changed the traffic system under control requires a transition time (several minutes or more, depending upon the complexity of the system), to reach the next optimized traffic flow state. This slow system response is unable to accomplish the objective of dynamic control [23, 24, 25]. An example of the cycle-based timing plan method — the Microprocessor Optimized Vehicle Actuation (MOVA) was developed by the Transport and Road Research Laboratory (TRRL) (Berkshire, UK) in 1988. It has been widely applied to isolated intersections in the United Kingdom.

  SCOOT (Split, Cycle and Offset Optimization Technique) is developed by TRRL (Transport and Road Research Laboratory) and the United Kingdom Department of Transportation of United Kingdom and industry in collaboration with the Ferranti, GEC and Plessey traffic systems companies SCOOT is a traffic responsive method for coordinating signals. Although the cycle based SCOOT calculates timing plans by considering coordination it is an isolated intersection timing plan approach. [12]

SCAT (Sydney Coordinated Area Traffic System) developed in Australia is a traffic-responsive system designed to provide both local (tactical) control and area (strategic) control. In light traffic conditions, SCAT attempts to minimize stops, in medium traffic to minimize total delay.

The non cycle-based timing plan approach generates adequate timing plans on-line, within 5 to 30 minute control periods. For example, the Road Commission for Oakland County, Michigan (RCOC) has deployed the Advanced Traffic Management System (ATMS) that utilizes an adaptive and coordinated traffic signal control method for multiple intersections. The signal control is driven exclusively by a video image processing based vehicle detection system [17, 19, 27]. The Optimization Policies for Adaptive Control (OPAC) developed by the University of Lowell is an approach based on the dynamic-programming technique and the demand-responsive decentralized traffic signal control strategy. [25]. Two other non cycle-based timing plan approaches for isolated intersection dynamic traffic signal control are the Microprocessor Optimized Vehicle Actuation (MOVA) system developed by TRRL [41] and the Signal Control at Isolated Intersection (SCII) system, developed at the Arizona State University [8]. The latter has a knowledge-based traffic signal control decision making system, and represents the application of an expert system to dynamic traffic signal control. The Real Time Optimization Program RTOP has operated in Metropolitan Toronto, Canada is a group of computer programs. These programs compute optimal signal timing parameters (cycle lengths, offsets and splits) on a real time basis, in response to the changes of traffic patterns.

The optimal timing plan design in dynamic traffic signal control can be classified into two major categories: the mathematical model based approach and the knowledge system based approach. The mathematical model based approach relies on a complex mathematical model for the traffic system. Dynamic traffic signal control is carried out by analyzing traffic flow conditions from the traffic flow data collected in real time, and by calculating the optimal timing plan parameters using the mathematical model. The knowledge system based approach, on the other hand, records traffic signal control knowledge in the knowledge base of an expert system. The expert system analyzes the collected traffic flow data in real time and searches for appropriate parameters of the timing plan. Quasi-optimal control parameters can be obtained for the dynamic control of a traffic system. The advantages and limitations of these two dynamic control strategies will be discussed in section 1.1.3.

The dynamic traffic signal control methods are based upon the use of either optimal or quasi-optimal timing plan parameters. The process of dynamic traffic signal control can be broken into the following four steps:

1. Acquisition of traffic flow data — To identify the traffic flow status, traffic flow data are collected from the field using either inductive loops or other detectors. These sensors are installed at appropriate road locations. The collected traffic flow data normally include volumes, speeds, densities, queue lengths, occupancies, etc., depending upon the need of the traffic controller and the scope of traffic analysis. For instance, the NEMA (a type of traffic industry standard) type controller requires volume and occupancy data for traffic signal control.

2. Prediction of traffic flow state — The prediction of the traffic flow condition plays an important role in dynamic traffic signal control. Traditionally, prediction of the traffic flow condition is carried out according to historical data. More recent research focuses on the short-term prediction based on the real time traffic conditions. This prediction can offset the time delay during traffic flow data acquisition/analysis and optimal timing plan parameter generation, and predict small upcoming changes of traffic flow state.

3. Traffic flow state analysis and timing plan generation — Based upon the acquired traffic flow data and the traffic state prediction, the traffic flow condition is analyzed. Appropriate timing plan parameters are identified using either the mathematical model based, or the knowledge system based optimal timing plan generation methods.

4. Implementation — The identified optimal or quasi-optimal timing planning parameters are converted into an appropriate form for implementing on the traffic controllers. The traffic signals on the road will be set using these parameters to accomplish the task of optimal and dynamic traffic signal control.

1.2.3 Limitations of Present Traffic signal control Methods

Present traffic signal control methods, either the widely used static traffic signal control technique, or the recently developed dynamic traffic signal control schemes, have many limitations.

The static signal control method predicts traffic flow state at any moment based upon the average traffic flow condition in a week and month or a specific season. Different traffic flow patterns are identified intuitively by a traffic engineer based upon his/her experience. This approach, as an open loop control strategy and a rough estimation scheme, is unable to accurately reflect the actual traffic flow conditions. The timing plans are generated in the "back room" far ahead of time using relatively simple and inaccurate mathematical models. The traffic flow state, predicated using this method, may be quite different from the actual traffic flow conditions on the road for a specific time (either over-estimated or under-estimated); the assigned timing plan parameters thus can seldom be optimal.

Dynamic traffic signal control was introduced to solve the problems faced by the conventional static traffic signal control practice. In principle, the dynamic signal control method can overcome the "open-loop" and "rough estimation" drawbacks of the static control approach. However, the previously discussed two typical dynamic traffic signal control schemes, i.e. mathematical model and conventional knowledge based model are very difficult to implement using present computer and traffic controller hardware.

To truly represent a traffic network using the mathematical model based approach, the mathematical model becomes quite complex. It is difficult to describe a large, time-varying and random traffic system using a simple and reliable analytic model. The optimization of dozens or hundreds of timing plan parameters, based upon this complex model, normally requires hours of CPU time on a work station. This delay precludes the implementation of this type of dynamic traffic signal control.

In the conventional, rule-based expert system approach [8], dynamic traffic signal control is conducted by matching traffic flow conditions of the rules and by assigning the timing plan parameters contained in the matched rules. These production rules normally have the following form:

IF north bound through volume is more than VOL_1
and less than VOL_2;
and south bound through volume is more than VOL_3
and less than VOL_4;
and north bound left turn volumes is more than VOL_5
and less than VOL_6;
and south bound left turn volumes more than VOL_7
and less than VOL_8;
and east bound through ...

THEN use the timing plan parameter SET_1

These production rules are based upon the binary logic with "true" or "false." However, traffic flow presents a lot of fuzziness and uncertainty which cannot be simply represented using binary logic. For instance, it is difficult to draw a clear boundary between the "large" and "small" traffic volume, simply based upon the number of passing vehicles in each unit time. It is also difficult to identify different traffic flow patterns, due to the large number of traffic flow parameters involved and the gradual change from each pattern. In addition, if one intends to represent all possible traffic flow patterns, the number of traffic flow patterns to be considered will become very large, due to the extraordinary number of possible combinations of traffic flow parameters of a large traffic network. This imposes exceptional difficulties on the search of the rule base. It becomes infeasible to either carry out this task using an on-site personal computer or transmit data through a regular communication line.

1.3 The Proposed Approach for Dynamic Traffic signal control

In dynamic traffic signal control, the major technical obstacles that have inhibited the application of optimal control strategy can be classified into four areas:

- appropriate identification of different traffic flow patterns to truly represent the actual traffic flow conditions using a systematic approach; fast traffic flow data processing and traffic flow condition assessment for on-line traffic signal control decision making;
- optimization of traffic signal control parameters of a traffic system based upon traffic flow condition assessment, considering all relevant objectives of optimal traffic signal control;
- and, efficient short-term prediction of traffic flow conditions.

This research focuses on the development of new methods to solve these four technical problems and to improve previous traffic signal control methods. A study on automated traffic pattern clustering using fuzzy pattern classification, on-line traffic flow pattern recognition using artificial neural networks (ANN), multiple-objective traffic signal control parameter optimization, and short-term adaptive prediction has been carried out.

The fuzzy pattern classification scheme has been used to replace the binary-logic-based expert system used in previous dynamic traffic signal control research to truly reflect traffic flow conditions with inherent fuzziness and uncertainty. The concept of fuzzy sets was introduced by Lotfi Zadeh in 1965, as a new way to represent vagueness [17]. As a generalization of the conventional set theory, fuzzy sets embed conventional set models into a larger setting. This setting endows fuzzy models to capture various aspects of incompleteness or imperfection, in whatever information and data of a real process. Fuzzy clustering is a method for identifying the appropriate classification and natural subgroup of events with fuzziness and uncertainty. Traffic flow condition is a typical event with fuzziness and uncertainty. For instance, one knows that traffic is heavy during rush hours, but the exact number of vehicles passing through the traffic system in unit time may vary significantly. The fuzzy clustering based traffic pattern clustering method is introduced in this work to truly represent the real traffic flow condition, which cannot be handled properly by the conventional expert system approach.

Artificial neural networks are used for the on-line traffic flow pattern recognition due to their robustness in pattern matching and high processing speed in data processing. ANN is a mathematical model of a multiple node and link network that has been developed to emulate the human neuron system. An ANN normally consists of nodes that associate with values and links that associate with weights. The nodes form network layers, including an input layer (the lowest layer), an output layer (the top layer), and some hidden layers. These layers of the network are inter-connected by links. Information passes from nodes of the lower layer to nodes of higher layer through weighted links. The optimal weights of these links for an application can be determined by feeding sample data to the nodes of the input and output layers. The weight adjusting process is carried out using a training algorithm, until the values of the output layer match the given outputs. The ANN, once trained and set with appropriate weights, can provide an ideal output for any given input data at the input layer. In addition to the high computation rates provided by massive parallelism, ANN is characterized by its robustness, fault tolerance, and smoothing capabilities, and it can provide a "good" output from multiple, incomplete and inaccurate data inputs [16]. Traffic flow demands and their mathematical models are inherently non-linear, multi-variable and stochastic. Sudden traffic demand changes due to isolated events, or errors of traffic demand data, introduced by a failed detector, often introduce undesired disturbances. The capabilities of ANN make it an ideal tool for identifying traffic flow patterns under these undesirable conditions from on-line acquired traffic data.

Traffic timing plan optimization is a multiple-objective optimization problem. Competitive objectives include minimum total delay, minimum individual delay, minimum fuel consumption, maximum bandwidth and maximum safety factor. In this work, a new optimization formulation is introduced to transfer the multiple objective optimization problem into a single objective function that represents a balance of these objectives. This formulation allows competitive optimal traffic signal control objectives to be considered jointly to produce a balanced control plan.

Another functional element of the intelligent and dynamic traffic signal control system is a short-term traffic flow condition prediction module. The prediction is based upon both the historical average traffic flow status and the current trend of traffic flow to improve prediction accuracy. This short-term traffic flow condition prediction is used to obtain the traffic flow condition of the next instance to allow the traffic signal control demand to account for the dynamic variation of traffic conditions.

The intelligent traffic signal control system introduced in this work consists of two parts: an off-line subsystem and an on-line subsystem. The off-line part has three major function modules: traffic pattern clustering and identification, optimal timing plan generation for each identified traffic pattern prototype, and traffic pattern training. This part generates a look-up table of optimal timing plans for the quick timing plan selection of the following on-line module. The on-line part of the system has five major components: traffic flow data acquisition, short-term traffic flow condition prediction, classification of traffic flow condition into one of the traffic pattern cluster using the fuzzy-neural networks, timing plan setting using the optimal timing plan parameters of the identified cluster, according to the look-up table.

A modular design approach is adopted in the development of the intelligent dynamic traffic signal control system to increase flexibility and reduce costs. The approach allows any component of the system to be replaced by new software and hardware without affecting the others. The widely-used LM-series traffic controllers and 486 DX2-66 microcomputer are used to avoid unnecessary re-design of the control hardware. Other controllers and computer systems can also be used by adopting different communication interface. The software was developed using the C++ window environment. The timing plans for identified primitive traffic patterns are generated using the widely used commercial software *PASSER II/III* and *TRANSYT-7F.*

1.4 Resemblance of Manufacturing and Traffic Planning and Control

In the last a few decades, extensive research has been carried out on the planning and control of manufacturing processes to improve productivity and reduce manufacturing costs. The planning and control of batch production systems present significant resemblance to dynamic traffic control. In this work, we apply the quantitative intelligent system approach [7], developed in the optimal planning of manufacturing activities, to the dynamic signal control problem of a corridor traffic system to minimize traffic delays.

With the rapid advance of information technology and importunate demand of higher productivity, new intelligent and integrated software tools have been introduced in the planning and control of manufacturing systems. The functions of these systems range from demand forecasting, optimization of production control parameters, such as batch size, routing plan, timing and detailed machining parameters, and dynamic adjustments for varying market demands and status of manufacturing equipment. These systems include:

- Computer based demand forecasting system
    - Time series prediction using regression
    - Seasonal and cyclic forecasting
- Integrated production control system

- Aggregate planning and master scheduling
- Material requirement planning I and II (MRP I and MRP II)
- Job sequencing and scheduling

▪ Computer-aided process planning system (CAPP)

- Planning of time or equipment conflict machining operations
- Determination of optimal machining parameters
- Optimal scheduling of machining operations

▪ The control programs for a Flexible Manufacturing System (FMS)

- Control of manufacturing cells
- Local network communications
- Coordination of time conflict manufacturing operations

These software tools analyze the status of the manufacturing system and the demands of the parts to be produced to generate an optimal plan, based upon the acquired manufacturing knowledge and the cost/delay minimization or productivity/profit maximization criteria. A typical example is the job scheduling program for controlling part flow in a flexible manufacturing system, as shown in Figure 1.1a. The function of the control program is to minimize the total workspan (machining and waiting time) for parts to be machined, thus achieving maximum productivity. To a large extent, the traffic control system has a similar function as the production scheduling system. The objective of traffic control is to find the optimal control parameters such as the green time, cycle length and offset at each intersection of a traffic network to minimize the total traffic delay. If we consider each intersection as a machine tool and each vehicle as a part, the traffic flow problem, as illustrated in Figure 1.1b, can be treated using a similar approach for optimal production process control. For the same reason, the technology developed in advanced traffic control may have potential applications in integrated production planning and control.

Due to the great resemblance between the manufacturing systems and traffic systems, the development of advanced dynamic traffic control methodology will also benefit the research on intelligent scheduling and planning for time and facility conflict manufacturing activities. The work will also extend the research and applications of the *quantitative intelligent system approach*, and contribute to the study of intelligent planning and scheduling of time and facility conflict activities in general.

1.5 Thesis Outline

The remaining chapters of this thesis are devoted to the following topics:

Chapter 2 describes the characteristics of a dynamic traffic system, including its mathematical representation and physical structure.

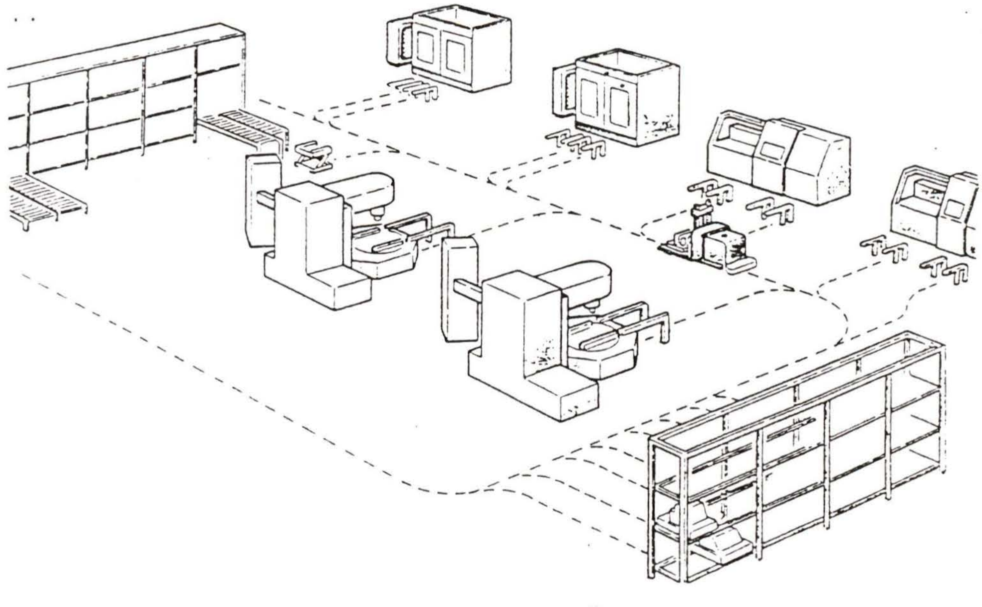
Chapter 3 presents the algorithm for fuzzy pattern clustering, method for clustering quality assessment and results from fuzzy cluster evaluation.

A description of a commercial traffic timing plan and simulation software. These are given in Chapter 4. A function that determines the combination factors of multi-objective traffic signal control objective function is proposed.

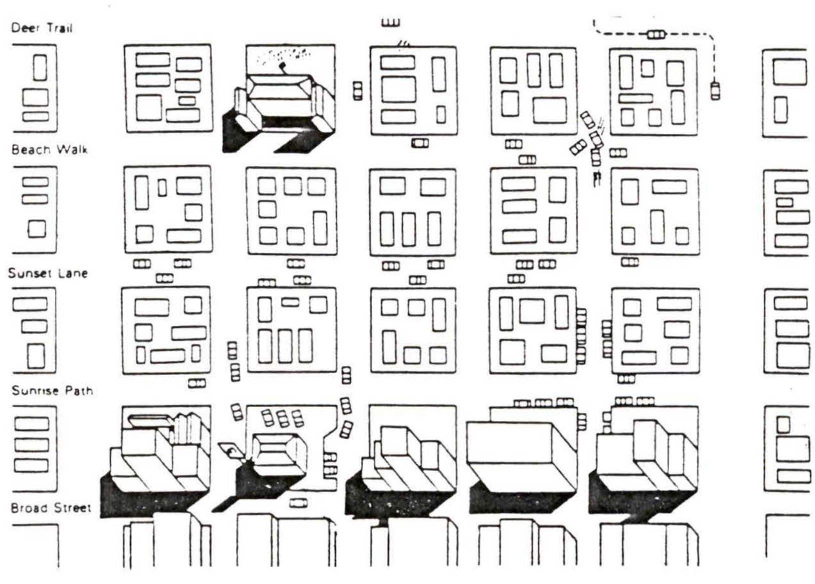
Chapter 5 focuses on several algorithms introduced for short-term traffic condition prediction.

The structure of a dedicated fuzzy-neural network and its use in dynamic traffic signal control are presented in Chapter 6.

Lastly, Chapter 7 summarizes the research carried out during this thesis work and the major contributions.



(a) Part Flow in A Flexible Manufacturing System



(b) Traffic Flow in A Urban Road Network

Figure 1.1: Resemblance of Production and Traffic Control Problems

Chapter 2

Intelligent Dynamic Traffic Signal Control – An Overview

2.1 Quantitative Representation of Traffic Flow Conditions

Today's road systems form a complex network for supporting various transportation needs. Efficient traffic flow is facilitated by coordinating and optimizing the control parameters at different signalized intersections. To successfully carry out this task, a good understanding of traffic flow condition at each intersection becomes essential. An appropriate mathematical representation of traffic flow conditions allows us to measure various traffic demands and to design adequate control schemes.

Due to the complexity of a traffic system and the dynamic nature of traffic demands, the knowledge and information related to traffic flow condition or traffic state are represented using multiple parameters, including traffic volume per time period, density, speed, occupancy, direction and so on. Traffic flow data represented using these parameters possess quantity, discreteness and uncertainty. The parameters tend to have a linear or nonlinear monatonic relation under unsaturated traffic conditions. For unsaturated traffic conditions, traffic volume (or simply volume) is normally used as the variable for representing traffic demand. The most effective and simplest representation of traffic condition is a hyper-point in the m-dimensional volume state space. Each of the m-dimensional space of the volume state represents the number of specific traffic movements in the traffic system. For instance, there exist three possible traffic movements for a one-lane road, in a given direction, at the intersection: (a) go through, (b) turn right, and (c) turn left. There are twelve movements in total at a four-way intersection.

The trajectory of hyper-point that represents traffic condition in the traffic volume state space presents a continuous curve. A simplified case with only two traffic movements, for ease of illustration, is shown in Figure 2.1, as a representation of traffic state. This representation is suitable for traffic flow prediction because the trend of traffic flow variation is well illustrated. However, in reality traffic conditions vary constantly, due to its stochastic, rather than deterministic nature. The traffic flow data can be collected only from the field in a discrete form. These data, representing traffic conditions, comprise the hyper-points and form clouds in the m-dimensional space as shown in Figure 2.2.

These clouds of hyper-points indicate the trend of traffic conditions (or quantity), as well as the uncertainty of traffic flow and discreteness of traffic data sampling. The first step to dynamically control a traffic system is to find the patterns of traffic flow. One way to accomplish this objective is to apply a statistical approach which eliminates the influence of random traffic demand variations. The traffic flow is also a time-varying event. An alternative can be carried out in two steps: (a) to classify all traffic flow conditions into a finite number of characteristic groups with representative traffic flow patterns; and (b) to identify the optimal control parameters for each representative pattern and to control traffic flow using the optimal control parameters of the closest representative traffic pattern. Identification of an appropriate traffic pattern group number and appropriate classification of traffic flow conditions into each group are two major technical challenges in this approach. In this work, we apply the fuzzy pattern clustering and classification methods to solve the problem.

With the fuzzy traffic pattern clustering and classification methods, we basically treat each cluster of traffic state cloud as a *characteristic group*, and use the density centers of traffic state cloud to represent the traffic patterns. The trajectories of various traffic pattern cluster centers thus represent the dynamic variation of the traffic volume (or traffic state) changes.

2.2 An Overview of the Intelligent Dynamic Traffic Signal Control System

In general, optimized traffic signal control parameters include cycle length, phase, offset. To accomplish dynamic traffic signal control that generates an optimal traffic signal control plan based upon on-line acquired traffic demands, an Intelligent Dynamic Traffic signal control System is developed in this work. The system combines a time-consuming mathematical model based optimal timing plan generation with efficient traffic pattern recognition mechanism, to produce a timing plan with quasi-optimal performance. The system has two major components: (a) the off-line part for traffic pattern clustering, training and analysis, as well as for generating the optimal timing plans for all recognized traffic patterns; and (b) the on-line part for implementing dynamic traffic signal control according to real traffic demands.

The intelligent dynamic traffic signal control system consists of several software and hardware building blocks, as illustrated in Figures 2.3 and 2.4. New function modules can be added to the existing system to replace existing modules without influencing other functional modules.

2.2.1 On-line Part Function Modules

The on-line part of the optimal dynamic traffic signal control system has three functional modules and two data bases. The function modules are: (a) traffic data acquisition; (b) short-term traffic condition prediction; and (c) optimal

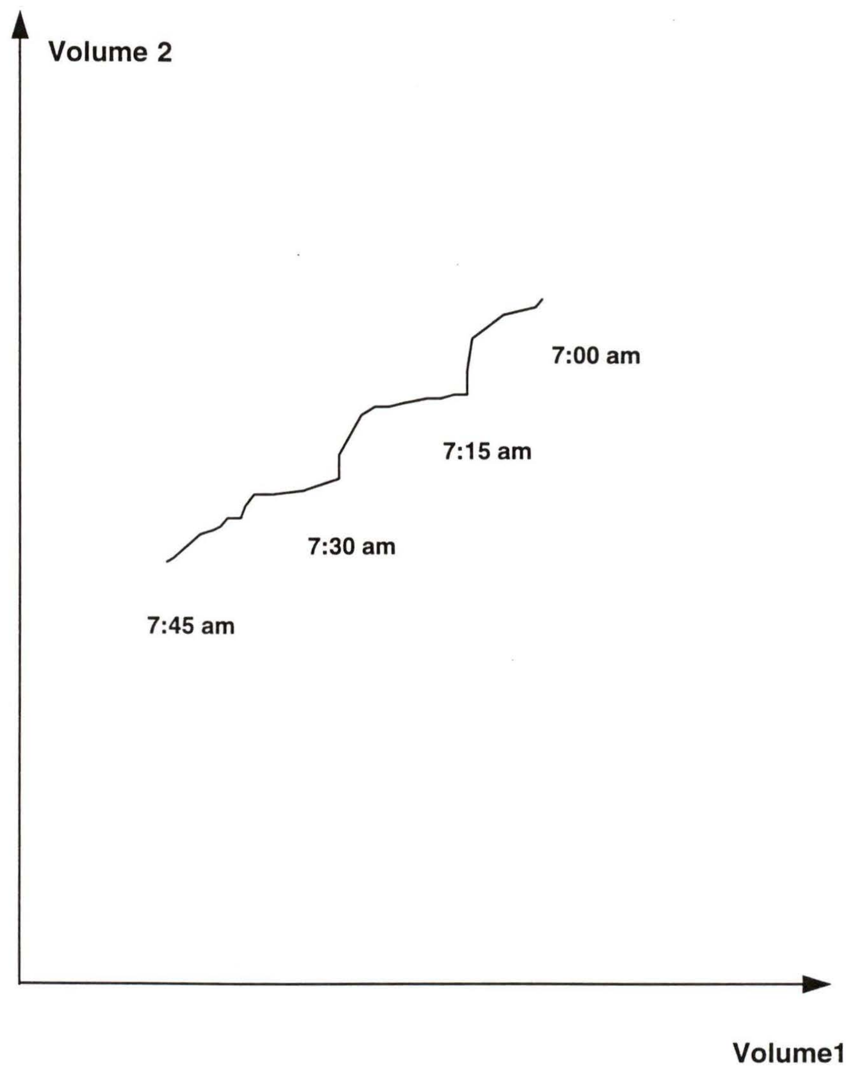


Figure 2.1: Representation of Traffic Volume Changes

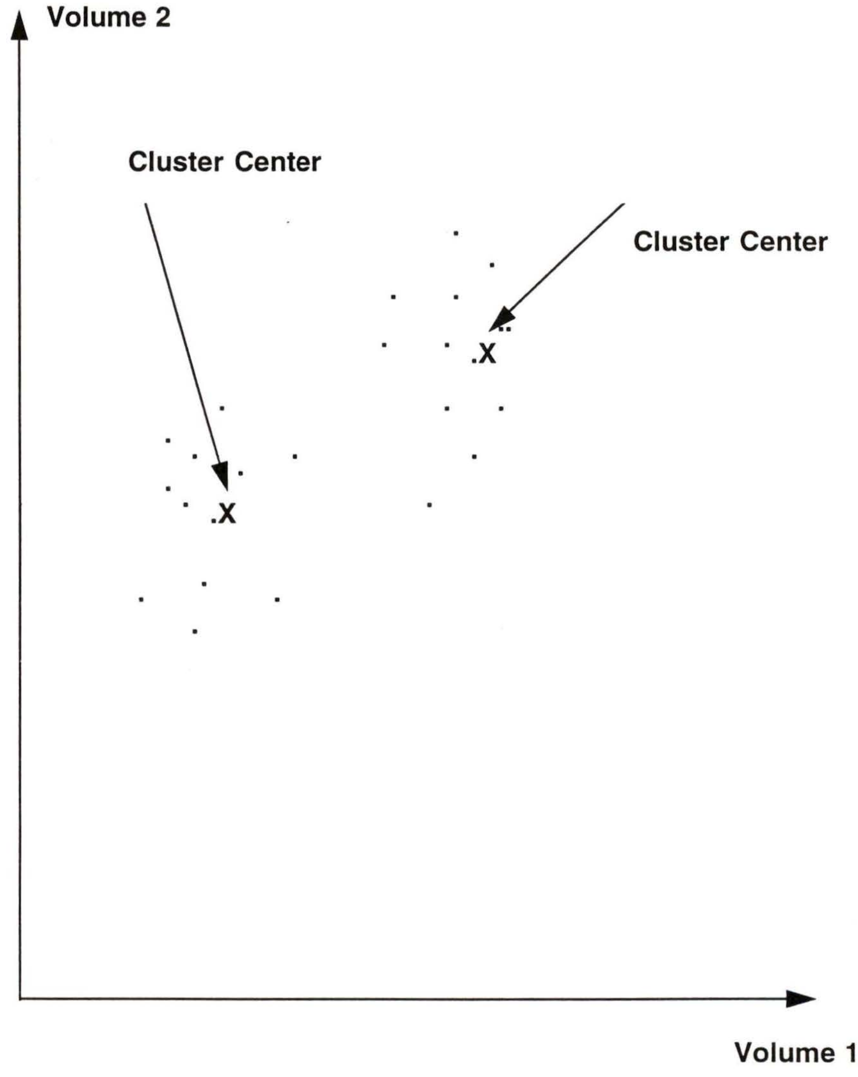


Figure 2.2: Representation of Traffic Flow Cloud

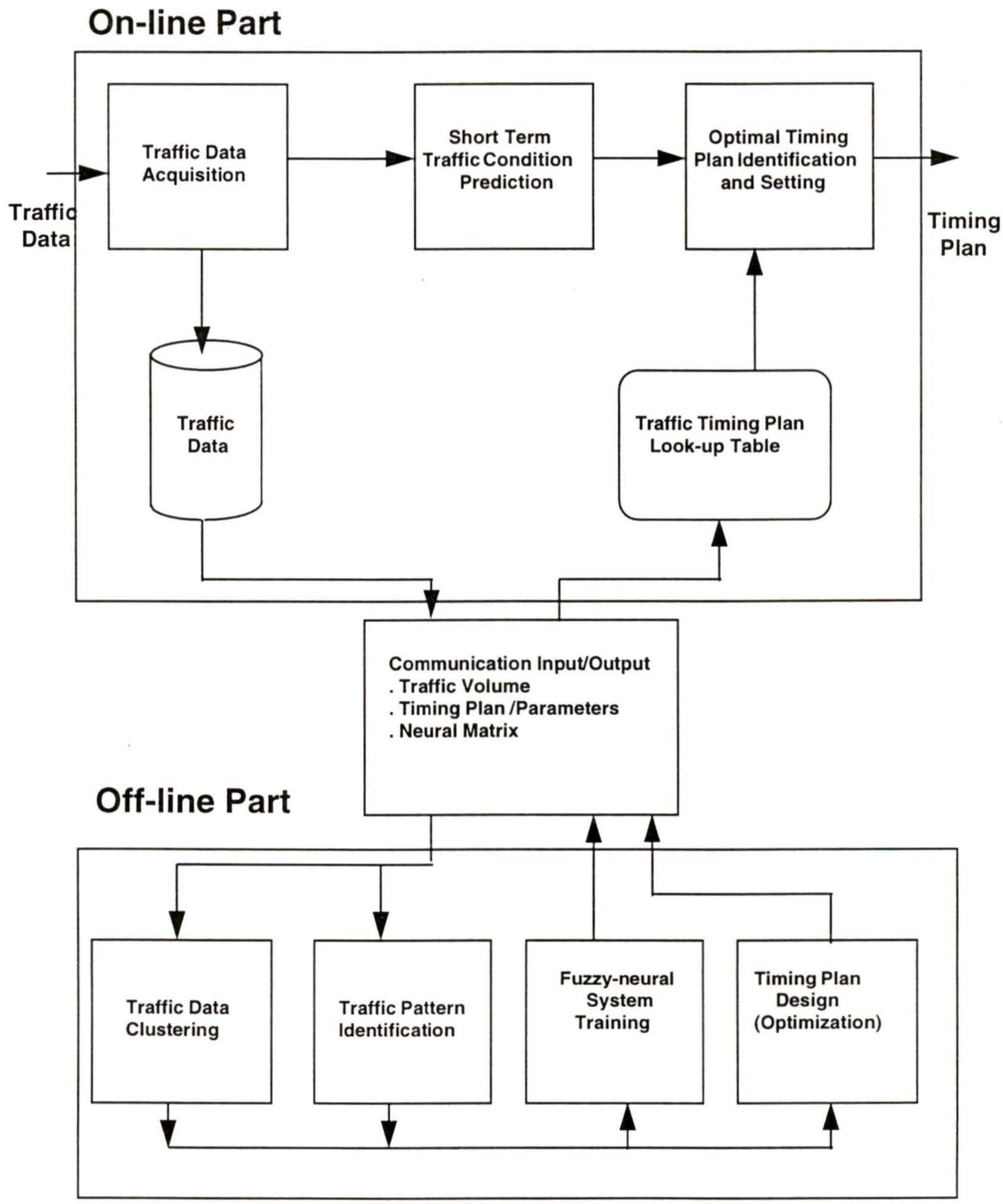


Figure 2.3: Functional Modules of the Intelligent Dynamic Traffic Control System (Software)

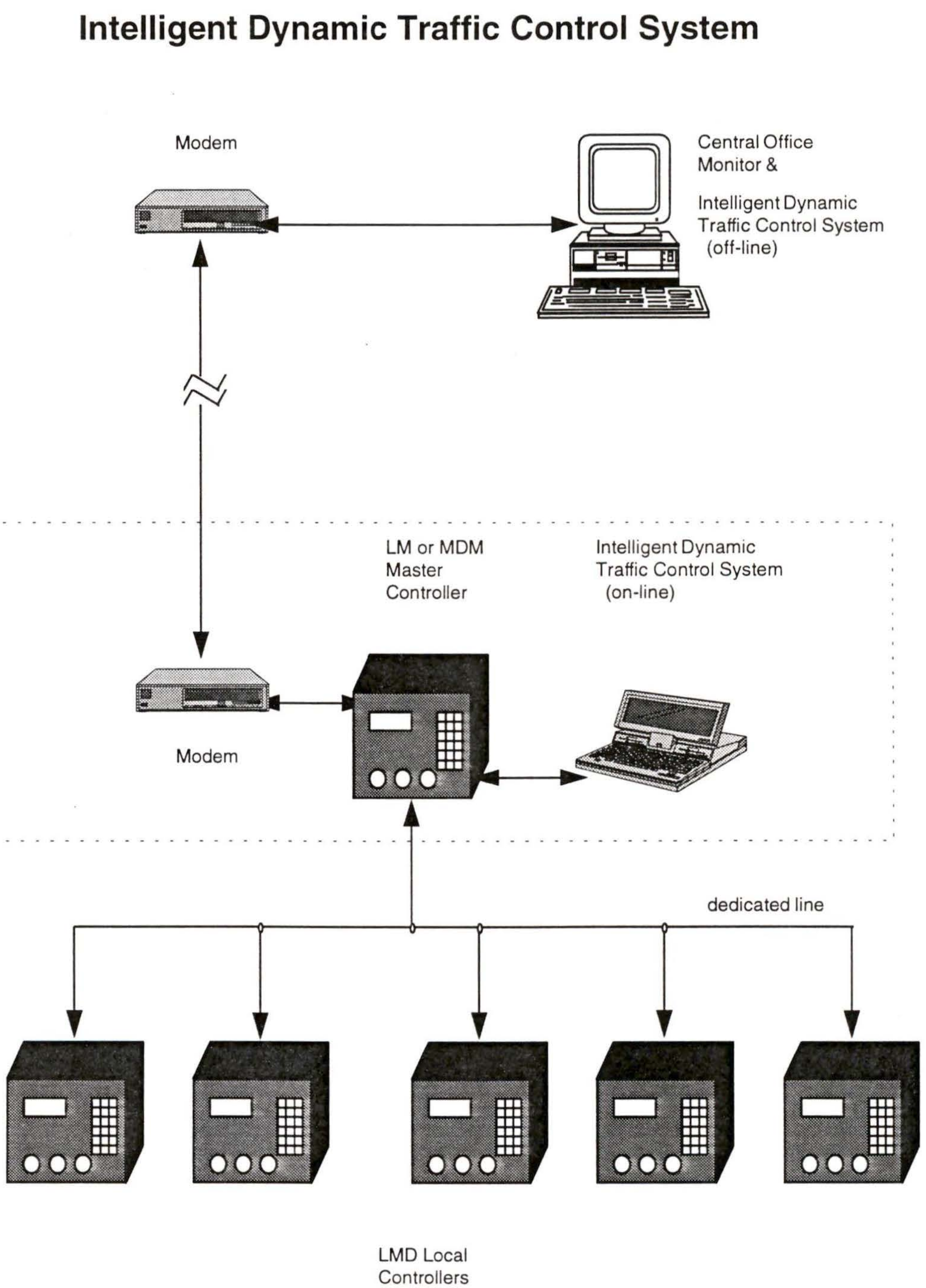


Figure 2.4: Functional Modules of the Intelligent Dynamic Traffic Control Structure (Hardware)

traffic timing plan identification and setting, and the data bases are: (a) traffic data recording, and (b) timing plan look-up table.

- **Traffic data acquisition**

  The traffic data acquisition unit collects traffic data from the field. Traffic data are collected either from inductance loops installed on road surfaces, or from other detecting devices, such as video camera. The traffic data are then transformed into a state space of traffic parameters such as volume, occupancy and speed.

- **Short-term traffic condition prediction**

  The traffic condition, identified by the traffic data acquisition and traffic pattern analysis system, represents the state of traffic flow (or demand) at the time of data acquisition. If a traffic timing plan based upon these data for the following time period were to be generated, it would introduce considerable error due to its time varying nature. To solve this problem and offset the time delay between traffic flow sensing and actual traffic signal control, a short-term traffic condition prediction mechanism is introduced in this work. This functional module analyses on-line collected traffic data for a short period of time to identify present traffic condition and interpret the trend of traffic flow variation. Historical traffic patterns are also used as a reference. Thus, combining inputs from these three sources, a good prediction of traffic flow condition for the next short period of time is generated. Two algorithms for the short-term traffic condition prediction are developed for different traffic conditions. These are:

  - Weighted filter prediction method
  - Adaptive trace prediction method

One of these two prediction methods is selected for use, based upon the actual condition of the traffic system. Detailed discussions of the method are given in Chapter 5.

- **Traffic timing plan identification and setting**

  The traffic timing plan identification module is a decision making unit. This subsystem searches through the knowledge base to identify the most suitable timing plan, based upon the predicted traffic condition.

  After the timing plan is selected, the specific number of the selected timing plan is passed to the traffic signal controller by the traffic timing plan setting unit. To control the traffic, the traffic signal controller will use the new control parameters specified by the timing plan. Several timing plans are generated and programmed into the traffic controller.

- **Traffic data recording**

  The dynamic traffic signal control system is continuously recording traffic flow conditions in several types of time intervals: 1 minute, 5 minutes, 10 minutes, or 15 minutes. These traffic data are transferred to the off-line part of the system, or the central database, on a daily basis. These collected data will be used for further training of the fuzzy-neural intelligent system, and for supporting the self-learning and self-adapting function of the intelligent dynamic traffic control system.

- **Timing plan look-up table** The pre-calculated timing plan parameters are stored in the timing plan look-up table. A timing plan will be sent to the traffic controller. when the new traffic pattern is identified using fuzzy-neural decision making. The traffic timing plan identification and setting module looks up the table, selects a group of control parameters and sends them to the traffic controller.

2.2.2 Off-line Function Modules

The off-line part of the system is composed of four functional modules. These are: input/output unit, clustering unit, timing plan design unit and control system training unit.

- **Input/output Unit**

  The functions of the input and output modules include:

  - Receiving traffic flow states detected by the traffic detection subsystem. For most applications inductive loops are used. The sensed traffic data includes traffic volume and occupancy. In this research traffic volume is measured using existing inductance loops.
  - Transferring the designed timing plans and other traffic signal control parameters to local traffic controllers.
  - Monitoring the state of traffic flow, if needed.

- **Traffic data clustering** Among the daily, monthly and annual cyclic variation of traffic flow states, the daily traffic flow changes are the most important and are considered as the main base for traffic signal control. The task of traffic data clustering is the cluster of traffic patterns using the fuzzy pattern clustering algorithm. A number of traffic pattern groups (or clusters) and their centers (or primitives) are generated according to the traffic data acquired at different times of the day. The identified group center represents the traffic patterns of each group. The fuzzy data clustering method establishes a mapping between the sensed traffic volume and traffic pattern group number. This mapping is represented as a simple input and output data pair in the knowledge-based system.

- **Timing plan design**

The timing plans for controlling traffic with minimum delay are designed based upon the traffic patterns extracted from daily traffic data. These timing plans are generated using the commercial software tools such as TRANSYT 7F and PASSER II, or a new algorithm for optimal traffic signal control using optimal timing searching. The optimal timing plans for all identified representative traffic patterns are designed.

- **Learning of traffic flow patterns and their optimal control parameters**

  The representative traffic flow patterns, identified through fuzzy traffic data clustering and the optimal traffic signal control parameters for these patterns, which are generated through timing plan design, are used to update the knowledge bases of our intelligent system. This knowledge is represented as weighting matrices (or membership functions) of a fuzzy-neural network. The form of these weighting matrices and membership functions will be discussed in detail in the following chapters. Historical traffic data are used to train the fuzzy-neural networks and thus obtain traffic pattern knowledge. The trained membership functions of the fuzzy-neural intelligent system form the rules for optimal traffic signal control decision-making. These rules are fed to the traffic controllers to carry out the dynamic traffic signal control system in real time.

2.2.3 Communication between the On-line and Off-line Parts

The communication module provides the traffic data transmission channel between the on-line part and the off-line part. The first function is the uploading (input) of the sensed traffic data from roads to the off-line control center (or the in-house computer), to monitor traffic flow condition, learn the new traffic patterns, and to analyze the system performance. The second function is to download (output) the optimal timing plans and their traffic signal control parameters to the on-line control part in the dynamic traffic signal control system. These plans and parameters are generated for the identified traffic patterns, using commercial timing plan design programs.

Chapter 3

Traffic Pattern Recognition

As discussed in previous chapters, traffic pattern recognition, including traffic condition clustering and representative pattern identification, is a critical part for dynamic traffic signal control.

Traffic flow conditions for a traffic system, at different times, can be represented as many hyper-points in an m-dimensional volume space. The objective of traffic pattern recognition is to scientifically classify all of these points into characteristic groups, and to assign any given point to the group with a best fit. The centers of these groups should also be identified as the representative of the group. We call this group center – representative traffic pattern. The technical problem is how to group traffic data points and identify the group centers.

Due to the time variant and stochastic nature of traffic condition, the hyper-points representing traffic flow conditions tend to be scattered into the traffic volume space. These points form clouds, rather than definite geometric shapes with a clear boundary. Conventional mathematical methods that deal with a deterministic model therefore cannot be used to properly process these data points. The fuzzy pattern recognition has thus been adopted for the traffic pattern recognition, due to its capability to handle uncertainty. Specifically, a fuzzy pattern clustering algorithm is used to partition traffic volume data into proper groups and to find the centers of these groups. The classification is optimized to assure that the data points have a minimum average distance to their group center. These identified groups form a traffic pattern with characteristic groups and representative models – the group centers. The optimal timing plans that minimize total traffic delay for traffic conditions of all group centers are designed. These optimized timing plans will be stored in the traffic controllers as a look-up table to dynamically control traffic flow.

Among various fuzzy pattern recognition algorithms, the fuzzy c-Means clustering algorithm [1, 3] is the best known and most widely used. This algorithm is based upon a profound mathematical basis [2]. It also provides more information about the clustering, and converges faster than other fuzzy clustering algorithms [2].

3.1 Fuzzy Set Theory and Fuzzy Clusters

Fuzzy set theory was introduced in 1965 by Zadeh [43] as a new way of representing vagueness in the physical world. The theory is a generalization of conventional set theory, one of the basic structures underlying computational mathematics and models. Computational pattern recognition has played a central role in the development of fuzzy modeling theory because fuzzy interpretations of uncertain data form a very natural and intuitively plausible way to formulate and solve various pattern recognition problems.

The definitions of fuzzy set and fuzzy clusters [2] are briefly given below:

- **Fuzzy Set**

  Let X be a space of points (objects), with a generic element of X denoted by x. Thus, $X = \{x\}$. A fuzzy set (class) A in X is characterized by a membership (or characteristic) function $f_A(x)$ that associates with each point in X[1] a real number in the interval $[0, 1]$ [2], with the values of $f_A(x)$

[1] More generally, the domain of definition of $f_A(x)$ may be restricted to a subset of X

[2] In a more general setting, the range of the membership function can be taken to be s suitable partially ordered set P. For our purposes, it is convenient and sufficient to restrict the range of f to the unit interval. If the values of $f_A(x)$ are interpreted as truth values, the

at x representing the "grade of membership" of x in A. Thus, the nearer the value of $f_A(x)$ to unity, the higher the grade of membership of x in A. When A is a set in the ordinary sense of the term, its membership function can take on only two values 0 and 1, with $f_A(x) = 1$ or 0 according to whether x does or does not belong to A. Thus, in this case $f_A(x)$ reduces to the familiar characteristic function of a set A. (When there is a need to differentiate between such sets and fuzzy sets, the sets with two-valued characteristic functions will be referred to as ordinary sets or simply sets.)

- **Fuzzy c Partition**

  (a) X is any finite set; (b) n is the number of elements of a finite sets; (c) c is an integer, representing cluster number, $2 < c < n$; (d) V_{cn} is the set of real $c \times n$ matrices;. Fuzzy c-partition space for X is the set:

$$M_{fc} = \{U \in V_{cn} \; |u_{ik} \in [0,1] \;\; \forall i, k;$$

$$\sum_{i=1}^{n} u_{ik} = 1 \;\; \forall k;$$

$$0 < \sum_{k=1}^{c} u_{ik} < n \;\; \forall i; |\}$$

- **Fuzzy c Means Functionals**

  Let:

$$J_m(U, v) = \sum_{i=1}^{n} \sum_{k=1}^{c} (u_{ik})^m (d_{ik})^2 \tag{3.1}$$

where

$$U \in M_{fc}$$

---

latter case corresponds to a multivalued logic with a continuum of truth values in the interval [0, 1].

is a fuzzy c-partition of X;

$$v = (v_1, v_2, \ldots.. v_c) \in R^c \quad with \quad V_i \in R^p$$

is the cluster center or prototype of u_i, $1 \leq i \leq c$

$$d_{ik}^2 = (||x_k - v_i||_A)^2 = (x_k - v_i)^T A(x_k - v_i)$$

$||.||$ is any inner product induced norm on R^p; m is a weighting exponent, $m \in [1, \infty)$, and A is a $p \times p$ positive definite coefficient matrix.

Examination of J_m reveals that the measure of dissimilarity is $d_{ik} = ||x_k - v_i||$, the distance between each data point x_k and a fuzzy prototype v_i; the squared distance is then weighted by $(u_{ik})^m = (u_i(x_k))^m$, the mth power of x_k' s membership in fuzzy cluster u_i. Since each term of J_m is proportional to $(d_{ik})^2$, J_m presents a squared error clustering criterion, and solutions of

$$\min_{M_{fc} \times R^{cp}} \{J_m(U, v)\} \tag{3.2}$$

lead to the least-squared error stationary points of J_m. An infinite family of fuzzy clustering algorithms, one for each m–is obtained via the necessary conditions for solving the above equation solutions above. The basic theorem follows.

- **Theorem1: Picard iteration through necessary condition** Let $m \in (1, \infty)$, and X have at least $c < n$ distinct points, $\Rightarrow$ represents "if-then", and $\Phi = \{0\} \quad \forall k$ define the sets

$$I_k = \{i | 1 \leq i \leq c; d_{ik} = ||x_k - v_i|| = 0\} \tag{3.3}$$

$$\tilde{I}_k = \{i | i \in (1, 2, \ldots, c)\} - I_k$$

then $(U, v) \in M_{fc} \times R^{cn}$ may be a globally minimal for J_m only if

$$I_k = \Phi \Rightarrow u_{ik} = \frac{1}{[\sum_{j=1}^{c}(\frac{d_{ik}}{d_{jk}})^{\frac{2}{m-1}}]} \tag{3.4}$$

or

$$I_k \neq \Phi \Rightarrow u_{ik} = 0 \;\; \forall i \in \tilde{I}_k \;\; and \;\; \sum_{i \in I_k} u_{ik} = 1 \tag{3.5}$$

$$v_i = \frac{\sum_{k=1}^{n}(u_{ik})^m x_k}{\sum_{k=1}^{n}(u_{ik})^m \forall i} \tag{3.6}$$

.

3.2 Fuzzy c-Means (FCM) Algorithm

The fuzzy c-Means (FCM) clustering algorithm is a set-partitioning method based upon Picard iteration through necessary conditions (Theorem1) for optimizing a weighted sum of squared error objective function (J_m). Early work on FCM is summarized in Bezdek [2]. The FCM algorithm is described as follows:

Fuzzy c-Means (FCM) Algorithm

- **Step 1** Fix c, $2 \leq c \leq n$; let $X = \{x_1, ..., x_p\} \subset R^p$ be a finite data set containing at least $c < n$ distinct points; and let R^{cn} denote the set of all real $c \times n$ matrices. A nondegenerate fuzzy c-partition of X is conveniently represented by a matrix U:$[u_{ik}] \in R^{cn}$, The entries of which satisfy

$$u_{ik} \in [0, 1], \quad 1 \leq i \leq c, 1 \leq k \leq n$$
$$\sum_{i=1}^{c} u_{ik} = 1, \quad 1 \leq k \leq n$$
$$\sum_{k=1}^{n} u_{ik} > 0, \quad 1 \leq i \leq c$$

The set of all matrices on R^{cn} satisfying above condition is denoted by M_{fcn}. A matrix $U \in M_{fcn}$ can be used to describe the cluster structure of X by interpreting u_{ik} as the grade of membership of x_k in the ith cluster. For instance $u_{ik} = 0.95$ represents a strong association of x_k in cluster i, while $u_{ik} = 0.01$ represents a weak association. Other useful information about cluster substructure can be conveyed to identify the prototypes or cluster centers, $v = (v_1, ..., v_c)^T \in R^{cs}$, where, c is the number of clusters and s is the dimension of the fuzzy space. Here v_i is the prototype for class i, $1 \leq i \leq c, v_i \in R^s$. The best partitions U of X and the cluster center or representative, v_i for cluster i) is defined by minimizing the c-Means objective function J_m

$$\min_{U,v} J_m(U, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m |x_k - v_i|^2 \tag{3.7}$$

where $m \in [1, \inf)$, and $U^{(0)} \in M_{fcn}$ is initialized at step l, $l = 0, 1, 2, \ldots$

- **Step 2** Calculate the c fuzzy cluster centers $\{v_i^{(l)} | i \in 1, \ldots, c\}$ using

$$v_i^{(l)} = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m} \quad \forall i \tag{3.8}$$

- **Step 3** Update $U^{(l)}$ using

$$I_k = \Phi \Rightarrow u_{ik} = \frac{1}{[\sum_{j=1}^{c} (\frac{d_{ik}}{d_{jk}})^{\frac{2}{m-1}}]}$$

or

$$I_k \neq \Phi \Rightarrow u_{ik} = 0 \;\forall i \in \tilde{I}_k \;\; and \;\; \sum_{i \in I_k} u_{ik} = 1$$

- **Step 4** Compare $U^{(l)}$ with $U^{(l+1)}$ in a convenient matrix norm: If $||U^{(l+1)} - U^{(l)}|| \leq \varepsilon$ stop; otherwise, return to **Step 2**.

The fuzzy c-Means always converges to a local minimum of J_m from an initial guess of v_{ik}, different choices of initial v_{ik} might lead to different local minima.

3.3 Clustering Quality Assessment

The fuzzy c-Means algorithm produces c cluster centers in the traffic volume space. These centers represent points about which the traffic data are concentrated. The c-Means algorithms discussed previously can process the cluster fuzzy set with a given number of cluster c. It is also important to be able to assess the partition quality, and to find the best group number, c, so that the true traffic patterns can be identified. This forms the so called "cluster validity problem." The tool to assess partition quality, *validity functional* [1], is a function that generates a number for measuring the quality of the clustering based upon the outputs of FCM.

By evaluating the functional for a variety of c and m, one can determine the optimal values of these two parameters for which the corresponding clustering best identifies the class structure of the fuzzy data.

The quality of fuzzy clustering is measured by how closely the data points are associated to the cluster centers, and it is the membership functions that measure the level of association and classification. If the membership function value of a particular data point is significantly larger than the others, then this point is identified as the cluster center of the points in the subset.

Commonly used criteria for judging the group quality include the maximum partition coefficient $\sum_i (u_{ik})^2$ approach [1], the maximum classification entropy $-\sum_i u_{ik} log u_{ik}$ approach [2], the maximum proportion exponent $\max_i u_{ik}$ approach [39] or the minimum $\min_i u_{ik}/\max_i u_{ik}$ approach.

However, these criteria do not work well because they refer only to the membership values, not the data point positions in the volume space. Also the criteria refers to the group itself and not the relationship between the groups. The values that are used for quality judgment are sensitive to the values of c and m for any fuzzy data.

In this work a new criterion is introduced. it is based upon the principle that the closer the distance among the data points in the same groups, and the further the distance among the data points in different groups, the better the classification. The approach is straightforward and works well when the group numbers, c, is far smaller than the number of data points in volume sample space.

This partition coefficient criterion is based upon is the fuzzy cluster quality index, ID_f:

$$ID_f = 1 - \frac{\frac{1}{c}\sum_{i=1}^{c}\left[\frac{1}{N}\sum_{j=1}^{N_i}\sqrt{\sum_{k=1}^{M_i}(p_{jik} - p_{cik})^2}\right]}{\frac{1}{c}\sum_{i=1}^{c}\left\{\frac{1}{c-i}\sum_{l=i+1}^{c}\left[\frac{1}{N}\sum_{j=1}^{N_i}\sqrt{\sum_{k=1}^{L_k}(p_{jik} - p_{cik})^2}\right]\right\}} \tag{3.9}$$

where, c is the group number; p_{cik}, p_{clk} are the ith or the lth cluster center, the kth movement (note that k can be any movement in a considered system); p_{jik} is the jth point, the kth movement in the ith group; N_i is the point number in the ith group; M_i is the movement number in the ith group.

The index curve ID_f, defined in Eq 3.9, is shown in Figure 3.1 as original partition index. When group number is greater than six, the quality index ID_f curve becomes flat. In traffic control to avoid switching timing plan frequently, a small number of traffic pattern is preferred. To this end a modified, comprehensive index, ID, is introduced by

$$ID = \alpha ID_f + (1 - \alpha) ID_g \tag{3.10}$$

$$ID_g = \frac{c}{c_{max}} \tag{3.11}$$

where, ID_f is the original fuzzy cluster quality index; ID_g is the group number index; c is the cluster number: c_{max} is the maximum cluster number allowed by the traffic control hardware. This modified quality index incorporates the number of clusters into the clustering quality. α is a weighting factor that is determined by the capability of the traffic control hardware (how many different

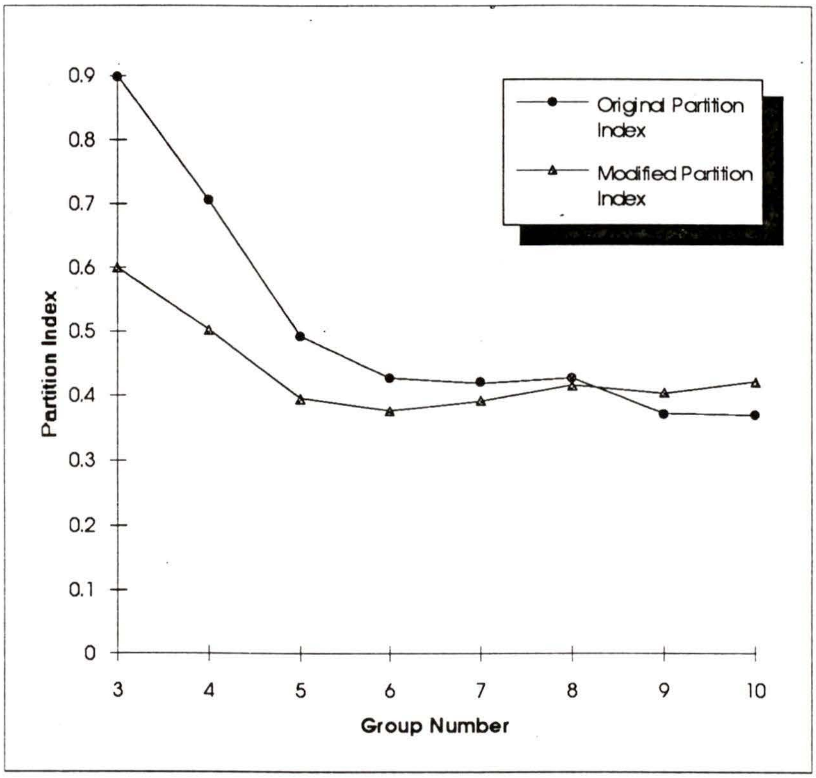


Figure 3.1: Explanation of Fuzzy Clustering Quality Assessment

traffic patterns the traffic controller can handle, and the stability of the traffic system under frequent timing plan switches).

The extensive tests with traffic data acquired from the BC highways, and consulting from experienced traffic experts lead to an optimal value of the weight factor, $\alpha = 0.6$.

3.4 Primitive Traffic Pattern Identification

The method for dividing the time of day into several time intervals of different traffic patterns is illustrated, using the traffic data from an example traffic system — the Duncan system.

The Duncan system is installed on the Trans Canada Highway at Duncan on Vancourver Island, B.C. The system has five intersections: Allenby, Boys, Trunk, Coronation, and James Streets as illustrated in Figure 3.4. Each intersection has a maximum of twelve movements, including through, left turn, and right turn in each direction, such as northbound through (NBT), southbound through (SBT), eastbound through (EBT), westbound through (WBT), northbound left turn (NBL), southbound left turn (SBL), eastbound left turn (EBL), westbound left turn (WBL), northbound right turn (NBR), southbound right turn (SBR), eastbound right turn (EBR), and westbound right turn (WBR).

The daily traffic volumes, measured from Allenby and Boys are given in Figure 3.3 and Figure 3.4, respectively.

Figure 3.5 and Figure 3.6 present the northbound and southbound through traffic passing the five intersections of the Duncan system in a day, respectively.

The traffic pattern identification problem is to divide the time of a day into several periods, each with a representative traffic volume pattern. The number of periods, starting and ending times, and representative traffic volume for each

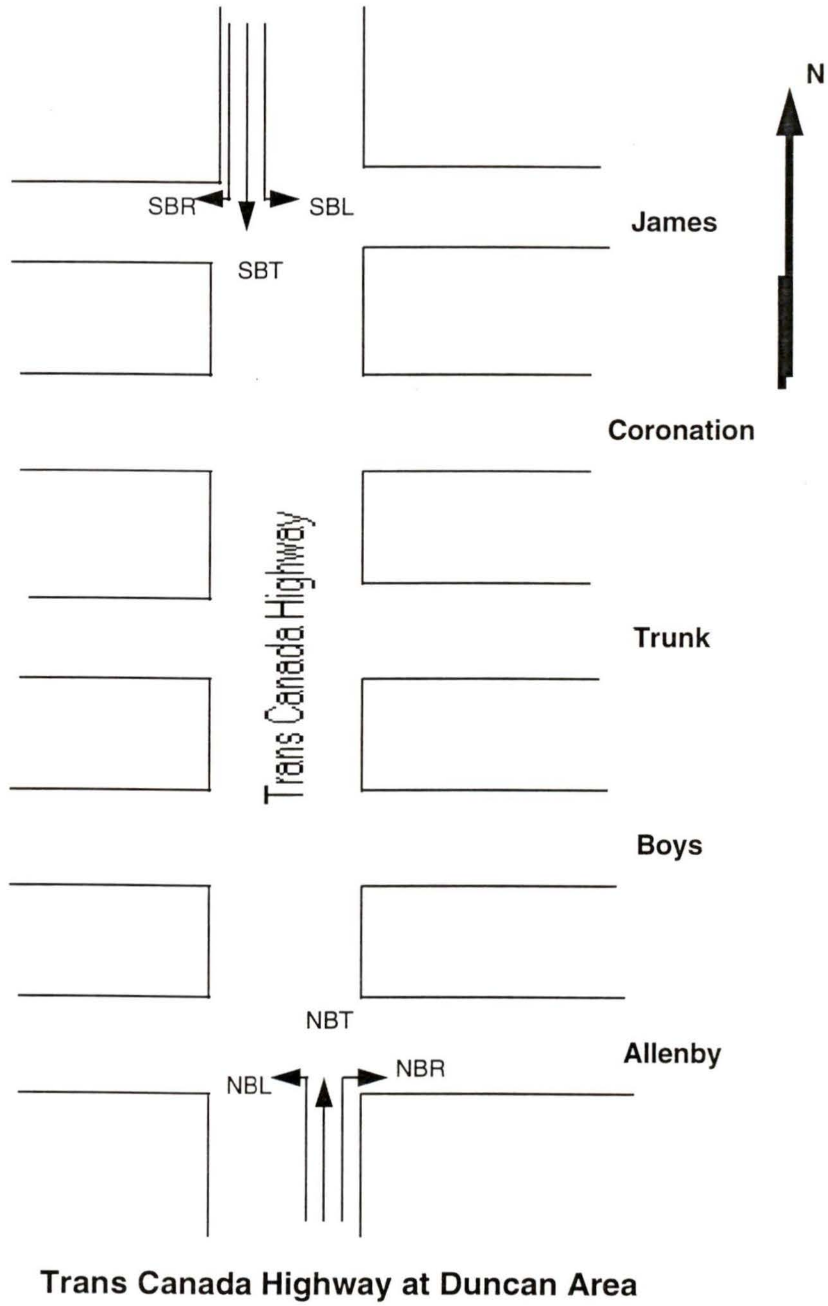


Figure 3.2: Duncan Traffic System Sketch

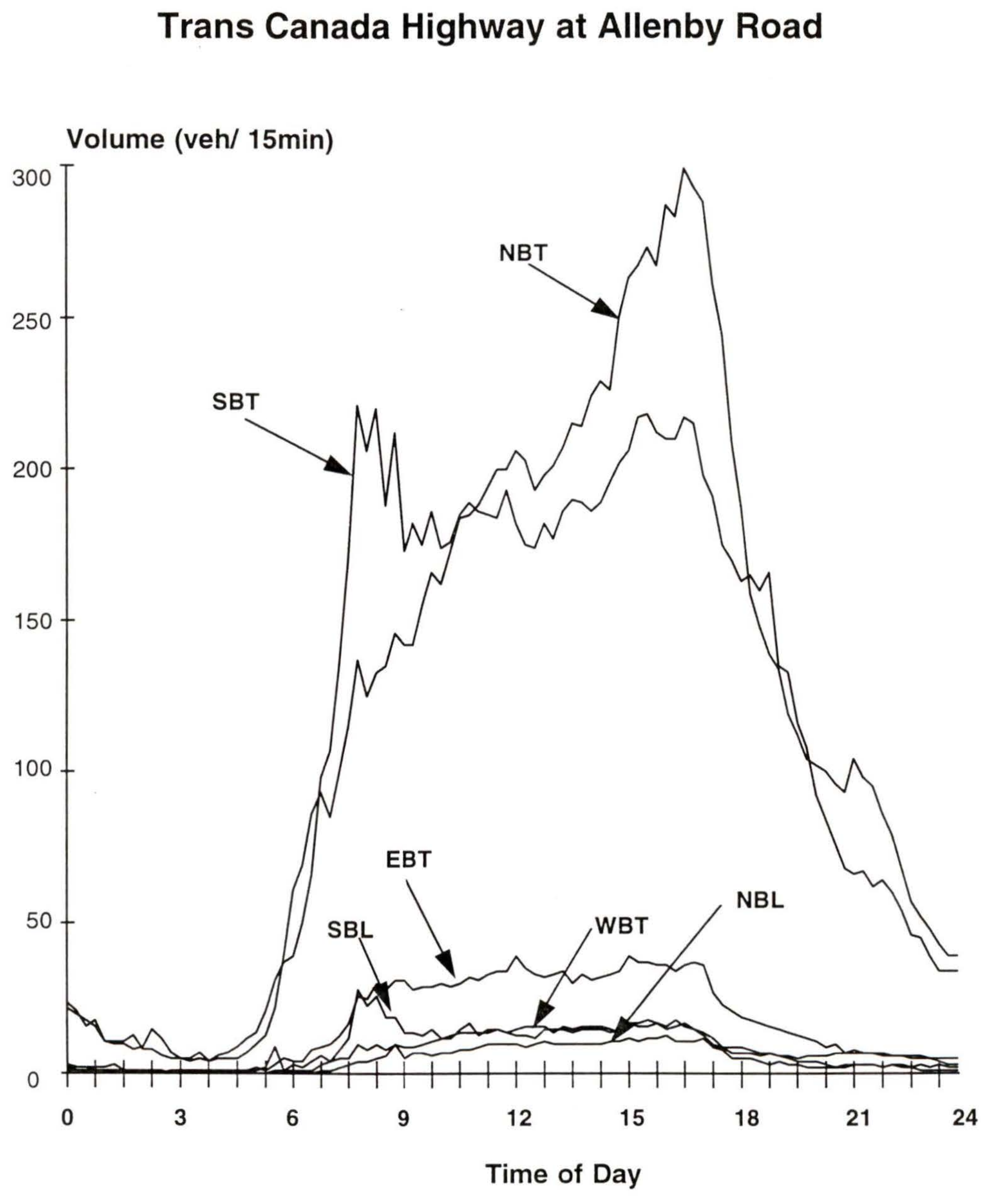


Figure 3.3: Intersection Traffic Volume at Allenby Street

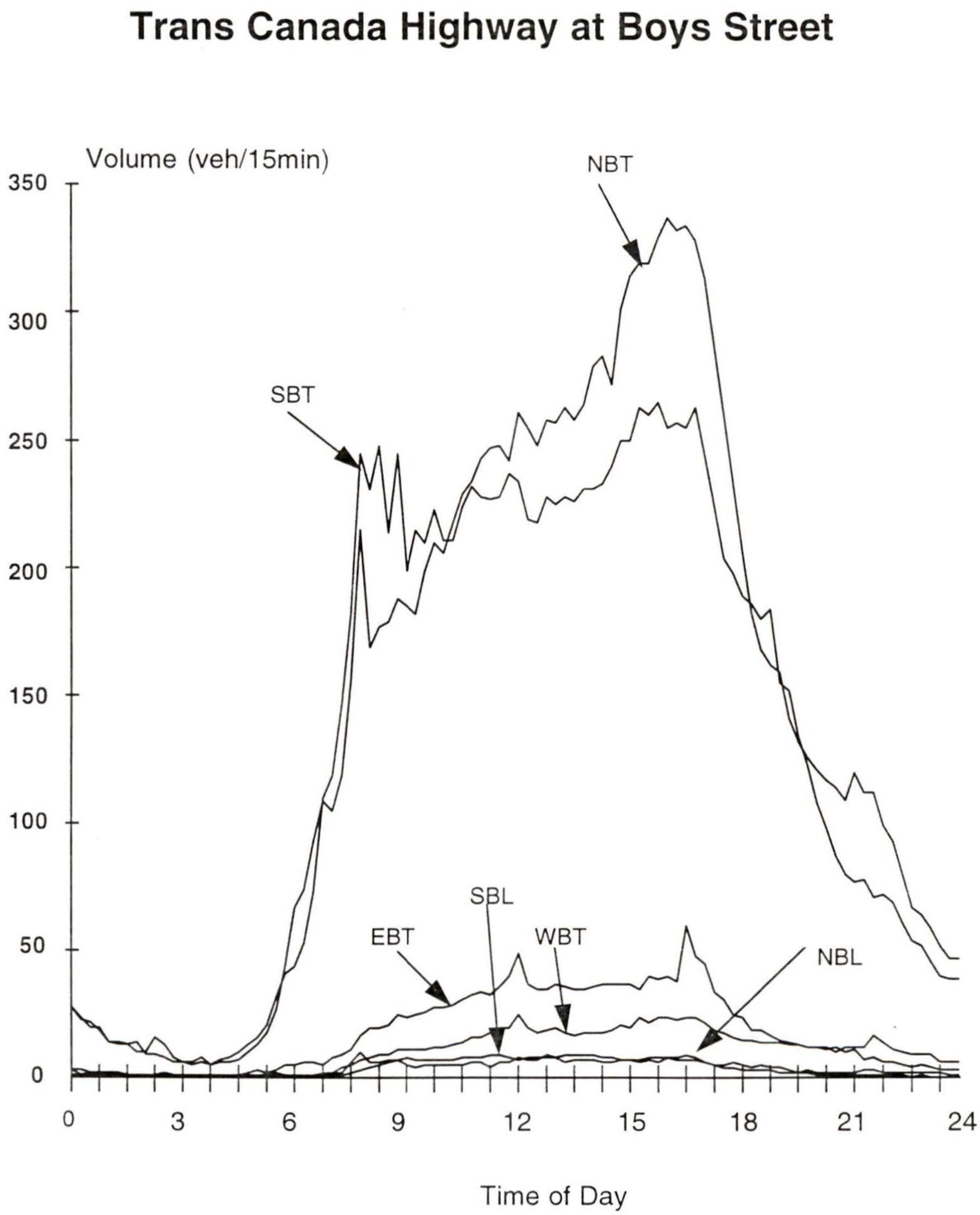


Figure 3.4: Intersection Traffic Volume at Boys Street

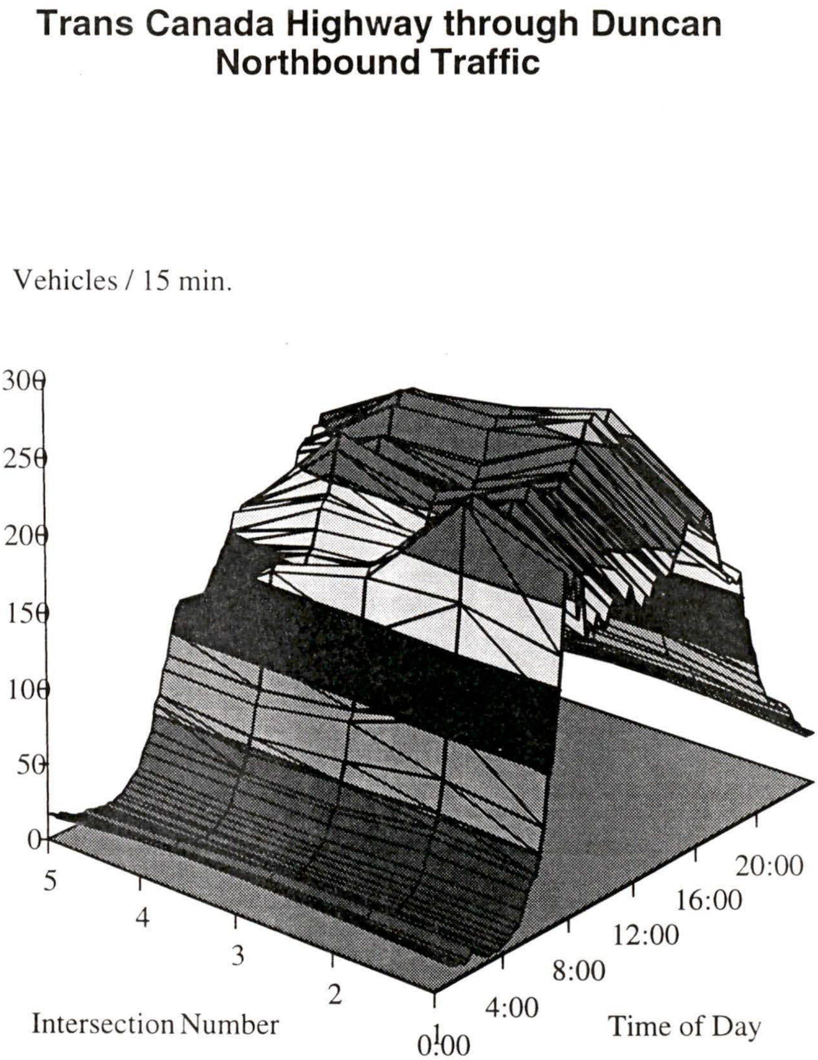


Figure 3.5: Duncan System Northbound Traffic Volume through Five Intersections 1. Allenby; 2. Boys; 3. Trunk; 4. Coronation; 5. James

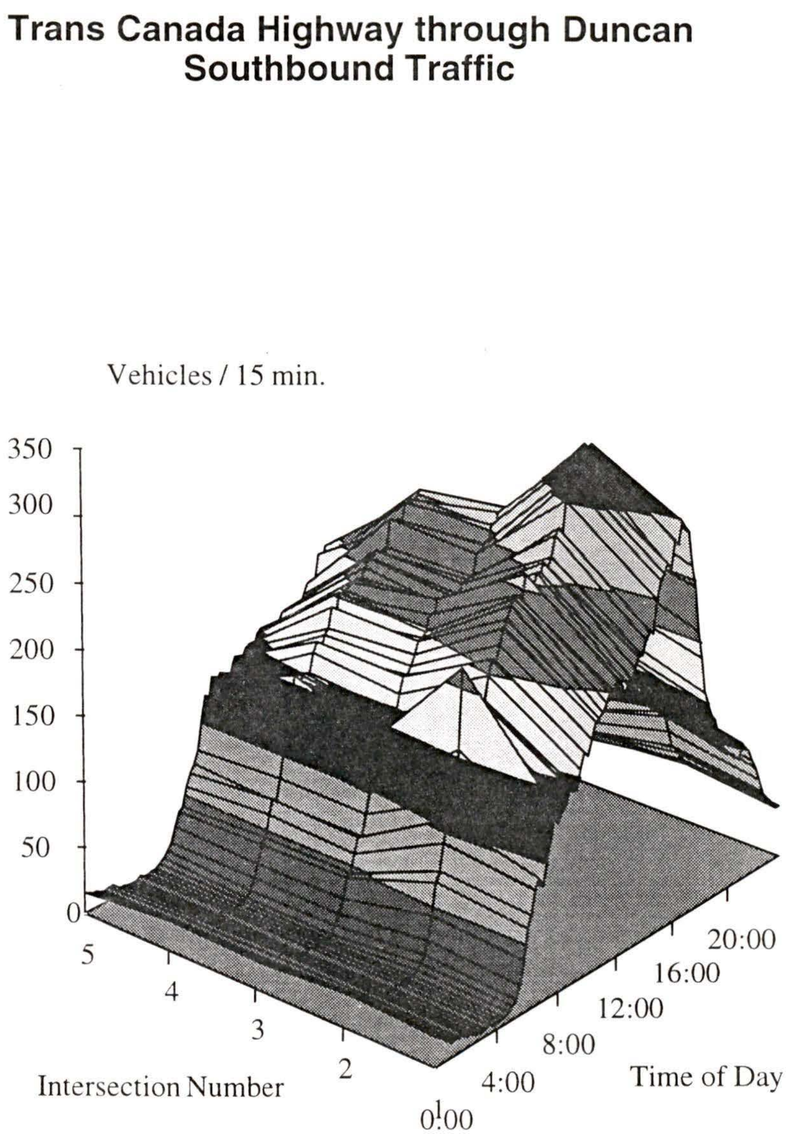


Figure 3.6: Duncan System Southbound Traffic Volume through Five Intersections 1. Allenby; 2. Boys; 3. Trunk; 4. Coronation; 5. James

movement are to be determined. Traditionally, this task is carried out manually by an experienced traffic engineer using the traffic volume data shown in Figures 3.3 – 3.6. This manual approach is highly inconsistent and inaccurate and often leads to over-estimation or under-estimation.

3.5 Automatic and Systematic Primitive Traffic Pattern Identification

The fuzzy pattern clustering method discussed previously is used in this work to improve the quality of traffic pattern identification. The method automates the labor intensive manual approach, and provides a systematic and scientific tool for the traffic pattern identification task.

The fuzzy pattern clustering system analyzes the traffic flow data, acquired from all involved intersections, and identifies the optimal traffic pattern clusters and cluster numbers.

A traffic pattern cluster or simply a traffic pattern is formed by the traffic volume points clustered as a group or a cluster. This cluster presents a piece of cloud in the m-dimensional traffic volume space. The fuzzy prototype of the cluster forms the representative model of each traffic pattern. The representative model, or fuzzy prototype is a m-dimensional point in the fuzzy space, of which each component represents a traffic movement. The starting and ending time of each traffic pattern, and the representative volumes of all movements are determined.

The first step of fuzzy traffic data clustering and traffic pattern identification is to represent the collected traffic volume data. Each traffic volume data point form a m-dimensional array (with m corresponding movements). The data collection time forms another important dimension, t, $(1 \leq t \leq T)$.

The collected traffic volume data are then stored in a $m \times T$ dimensional array, X:

$$\mathbf{X} = \left\{ \begin{array}{c} \mathbf{x_1} \\ \mathbf{x_2} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{x_T} \end{array} \right\} \qquad and \quad \mathbf{x_i} = \{x^i_1, x^i_2, \ldots, x^i_m\}$$

For the Duncan system example, five intersections, each with a maximum of 12 movements, are considered. A fuzzy space of dimension $5 \times 12 = 60$ is formed. The traffic volume are the average of summer of 1993. The traffic volume acquisition interval is 15 minutes, a total of 96 sample ($T = 96$) can be obtained for a day.

The collected data set, X, is then used as the fuzzy data input of the fuzzy pattern identification system, as illustrated in Figure 3.7. The fuzzy data X after being provided by the algorithm presented in section 3.2 and illustrated in Figure 3.7, lead to the fuzzy cluster time intervals, fuzzy cluster prototypes, fuzzy matrix, and hard partition matrix. The hard partition matrix specifies time intervals of all traffic patterns, and carries out defuzziness.

The optimal cluster number, ID^*, for the Duncan system is five, based upon the comprehensive clustering index values of different cluster numbers, as shown in Figure 3.8 -3.9.

$$ID^* = \min(ID_i) = 5$$

The defuzziness leads to five time intervals belonging to five different traffic patterns:

- 0000 - 0615 and 2245 - 0000 (group 1) (free operation)
- 0615 - 0730 and 1945 - 2245 (group 2)
- 0730 - 1000 and 1800 - 1945 (group 3)

- 1000 - 1445 and 1730 - 1800 (group 4)
- 1445 - 1730 (group 5)

The fuzzy clustering is carried out with an initial matrix, $U_{(0)}$, with random data from (0,1). The fuzzy vectors of the matrix must satisfy the non-correlative condition.

To illustrate the advantage of fuzzy traffic pattern classification, the total traffic delay of the fuzzy traffic pattern identification method-based traffic signal control, and the manual traffic pattern identification method-based traffic signal control, are compared. In both cases, an identical traffic signal control scheme (timing plan generation method) is used. The difference on traffic delay is introduced purely by the improvement on traffic volume data clustering.

The manual traffic pattern identification method produced the following traffic patterns for the Duncan traffic system.

- 7:00 am - 9:00 am (group 1)
- 9:00 am - 6:00 pm (group 2)
- 6:00 pm - 9:00 pm (group 3)
- 9:00 pm - 7:00 am (group 4) (free operation)

The traffic delays for these two cases are given in Table 3.1. The results from Table 3.1 show the total delay reduces 41 percent, the individual delay reduces 63 percent and the stops reduce 47 percent.

These time intervals and identical traffic patterns are illustrated in Figure 3.8-3.9, using the traffic volume diagrams at the Allenby and Boys Streets.

The developed fuzzy clustering program, *FUZZY-TRAFFIC* that runs in MS-Window environment, is used to determine traffic patterns, optimal timing

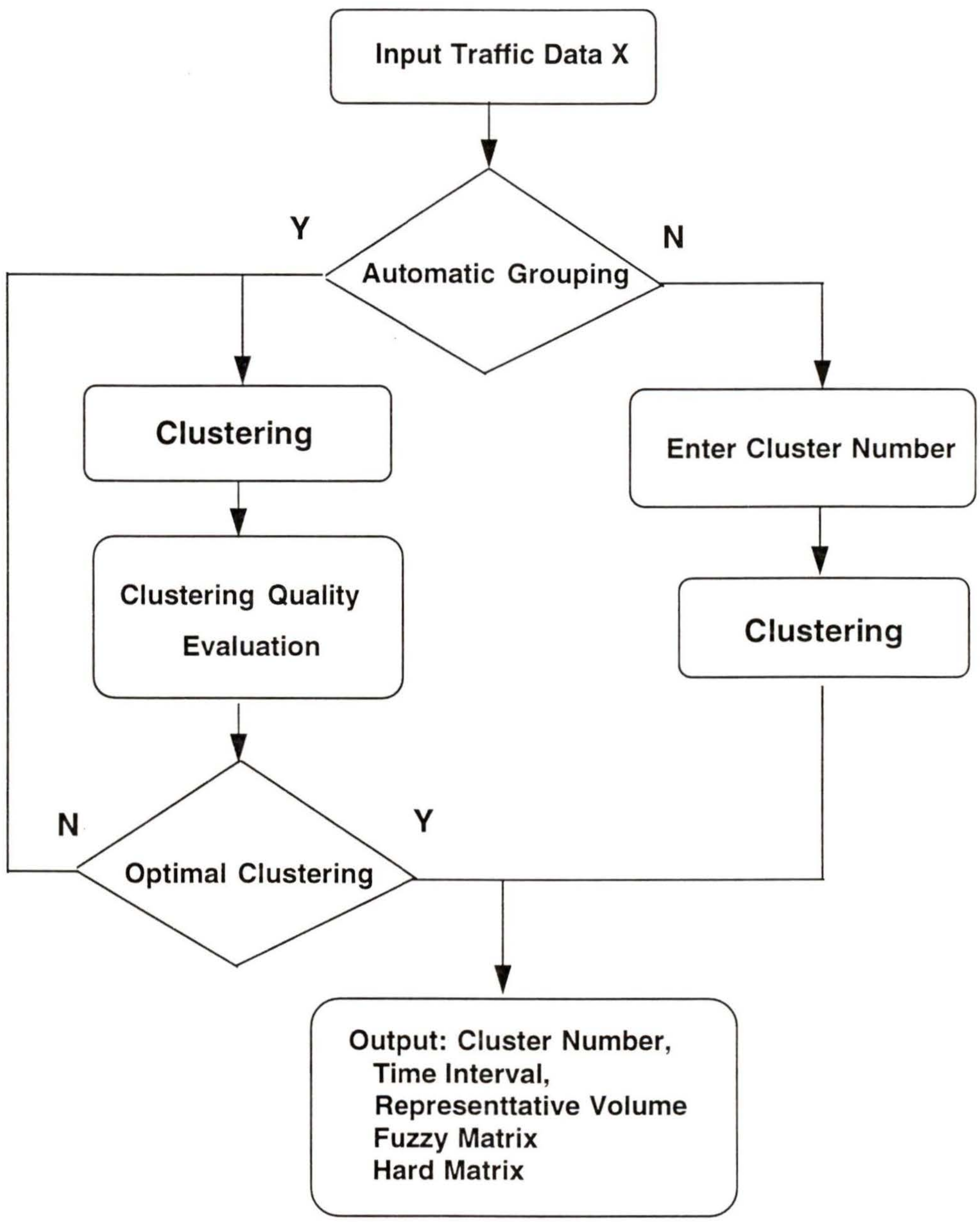


Figure 3.7: Fuzzy Pattern Identification Flow Chart

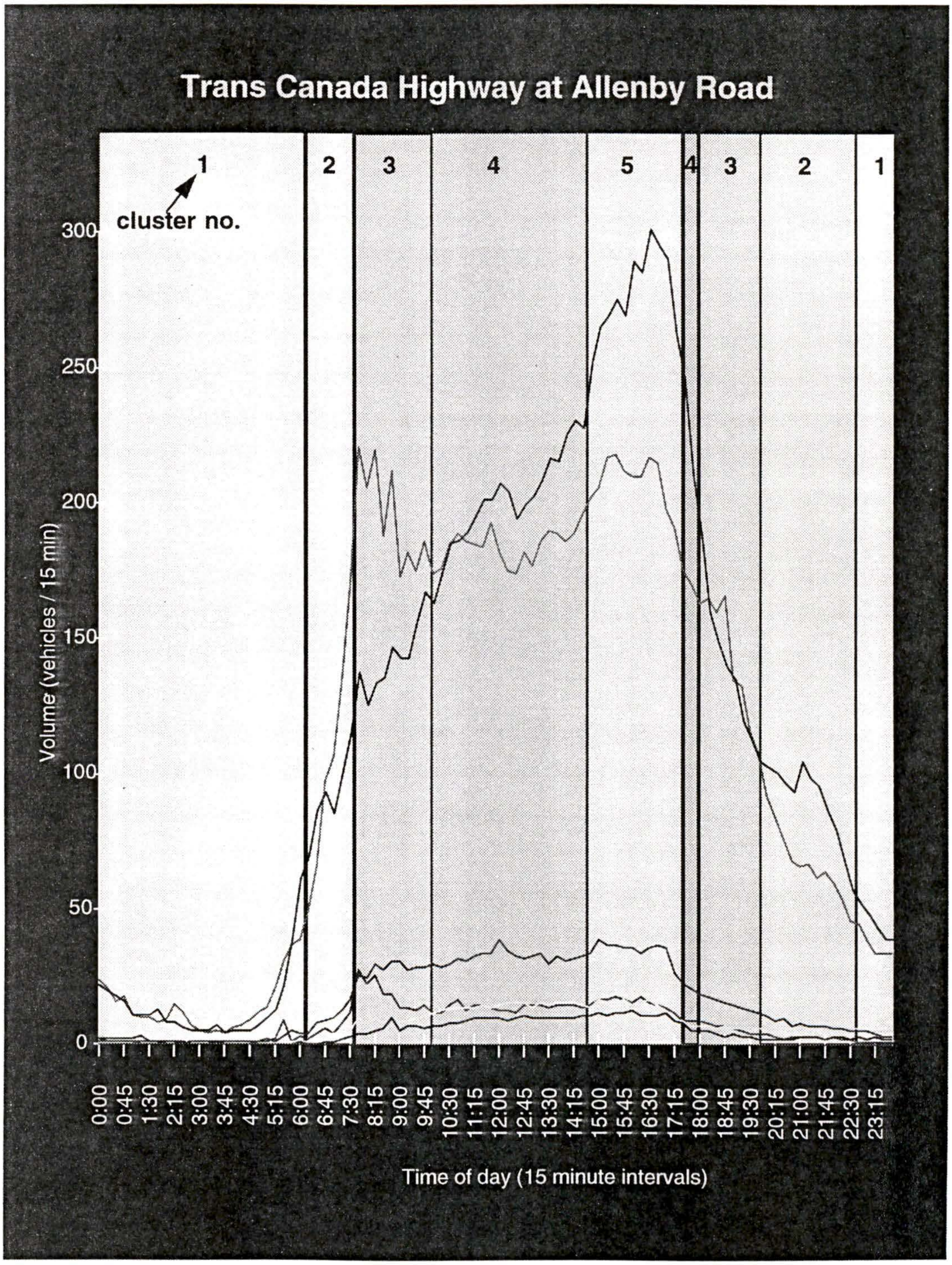


Figure 3.8: Time Interval Partitions in Allenby

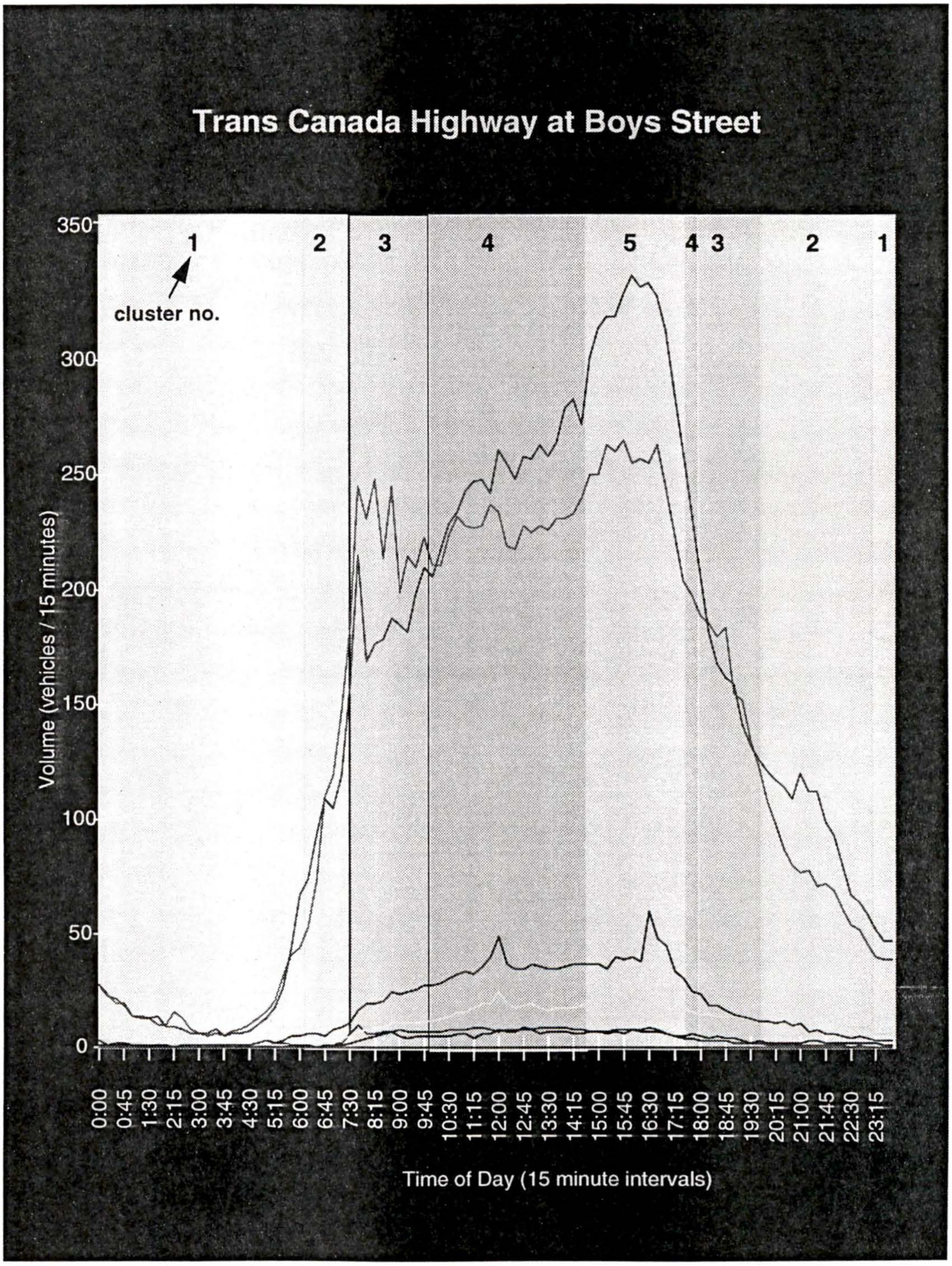


Figure 3.9: Time Interval Partitions in Boys

Table 3.1: Traffic Delay and Stops of Manual and Fuzzy Timing Plan

approach	Total Delay (veh-hr)	Indiv Delay (sec/veh)	Stops
Manual	68	2.76	5374
Fuzzy	40	1.03	2871

intervals, cluster prototypes, and a fuzzy cluster matrix for further use. The *FUZZY-TRAFFIC* gets traffic data from formatted traffic volume files and generates input matrix X. The output fuzzy matrix of the *FUZZY-TRAFFIC* is given in Appendix A. The flow chart of *FUZZY-TRAFFIC* is shown in Figure 3.7.

Chapter 4

Design of Optimal Timing Plan

The design of an optimal timing plan is aimed at finding the optimal time allocation to all involved conflicting movements of the traffic system. The time horizon is normally an hour (3600 seconds). The optimal timing plans can considerably improve traffic flow, fuel consumption, and vehicle operating costs.

The widely used commercial software *TRANSYT-7F* , TRAffic Network StudY Tool – Version 7. for designing optimal timing plan is adopted in this work. A program that carries out combined multiple objective evaluation, analyzes *THANSYT-7F* outputs, and searches for the optimal traffic signal control parameters is developed.

4.1 Description of TRANSYT-7F

The *TRANSYT-7F* is a widely used traffic flow analysis and optimization program. This program was originally introduced in UK and further developed by Transportation (McTrans), Transportation Research Center, University of Florida. The core of the *TRANSYT-7F* system is a function for analyzing the performance of the traffic system. The function is usually called traffic performance measures or traffic performance index (PI). The *TRANSYT-7F* program can be used for simple system evaluation, or for optimizing the system performance, using performance index function, PI. The performance index function is either a simple item or a linear combination of traffic delays and stops, fuel consumption (and optionally excessive maximum back of queue); or excess operating costs.

4.2 Traffic Flow Simulation

The optimization of the traffic timing plan by the *TRANSYT-7F* system is carried out by simulating various traffic flows in a signalized traffic system. The simulation is accomplished using an analytical model that represents real world events, such as traffic flow through the traffic system, stopped at intersections by red lights and restarted to flow by green lights. The optimization considers the following four essential timing plan elements, or optimization variables:

- *Cycle length* — The cycle length is the amount of time during which all movements at a signalized intersection are accommodated. *TRANSYT-7F* processes the cycle length based upon the following constraints:

  - The cycle length must be long enough to provide sufficient time for all phases, considering both vehicle and pedestrian requirements. The sum of these minimum phase lengths is the absolute lower limit of the cycle length.
  - The cycle length should be long enough to ensure that no movement is saturated. The degree of saturation should be less than 100 percent for all approaches at all intersections. This constraint generally leads to a higher value for the cycle length than the previous constraint.
  - The cycle length should not be too long to cause unacceptable high delays.
  - In a progressive traffic system the cycle length should be chosen to facilitate traffic progression.

- *Phase sequence* — A typical cycle may consist of two to eight phases, depending upon the number of traffic movements that require protection during their respective green periods. Phase sequences may consist of numerous combinations of protected and permitted movements.

- *Interval and phase lengths (splits)* — A split is an interval (a segment of the cycle), in which all signal displays, both traffic and pedestrian, are unchanged.

- *Offsets* — The offsets are the time difference from a system reference point to the beginning point of the cycle at all signal controllers in the system. Offsets are generally determined so that, to the maximum extent, traffic can flow through a number of signals without being stopped.

In *TRANSYT-7F*, an offset may be referenced to the start of any intervals in the cycle. Offsets are explicitly optimized by *TRANSYT-7F*.

4.3 Optimization Objective Function

The performance index or optimization objective function in *TRANSYT-7F* can be defined by a user in a number of ways. One of the most important parts of the objective function is the Disutility Index, DI, which is described by:

$$DI = \sum_{i=1}^{n}\{[W_{d_i} + KW_{s_i}] + U_i[W_{d_{i-1}} + KW_{s_{i-1}}S_{i-1}] + QB_i[W_q(q_i - qc_i)^2]\} \quad (4.1)$$

where,

n — the number of links;

d_i — delay on link i (of n links) and on an optional user-specified upstream input link, designated here as link $i-1$ (unit: vehicle-hour);

K — a user coded "stop penalty" factor to express the importance of stops relative to delay (note, if coded as "-1" in TRANSYT, the base W_{xi} are set internally so that the base DI is equivalent to excess fuel consumption);

S_i — stops on link i (and similarly for link $i-1$ (unit: stops/second);

W_{xi} — link specific weighting factors for delay (d) and stops (S) for link i (and $i-1$);

U_i — a binary variable which is "1" if link to link weighting has been established for link i, zero otherwise;

Q — a binary variable set by the user which if "1" includes the maximum back of queue penalty in the DI, or zero otherwise; B_i — a binary variable which is "1" if the maximum back queue (q_i) exceeds the user-specified storage capacity (the number of vehicles between two intersections), or zero otherwise;

W_q — a network-wide "penalty" applied to the excess queue "spillover,"

q_i — computed maximum back of queue on link i; and

qc_i — maximum back of queue "capacity" for link i.

4.4 Evaluation of the Objective Function

4.4.1 Calculation of Traffic Delay

One of the most important item in DI is the delay of vehicles in the traffic system. Delay represents indirect costs to the motorists in terms of lost time and a direct cost in terms of fuel consumption during idling. Excessive delay at signalized intersection reflects inefficiency in the signal timing. In a field study, delay is usually measured by periodically counting the number of vehicles queued at a signal, and integrating the counts over time, as illustrated in Figure 4.1.

The *TRANSYT-7F* can approximate only one such histogram (flow profile). The variation of arrivals from cycle to cycle, due to the randomness, of traffic flow needs to be considered particularly near or at saturation, when the influence of

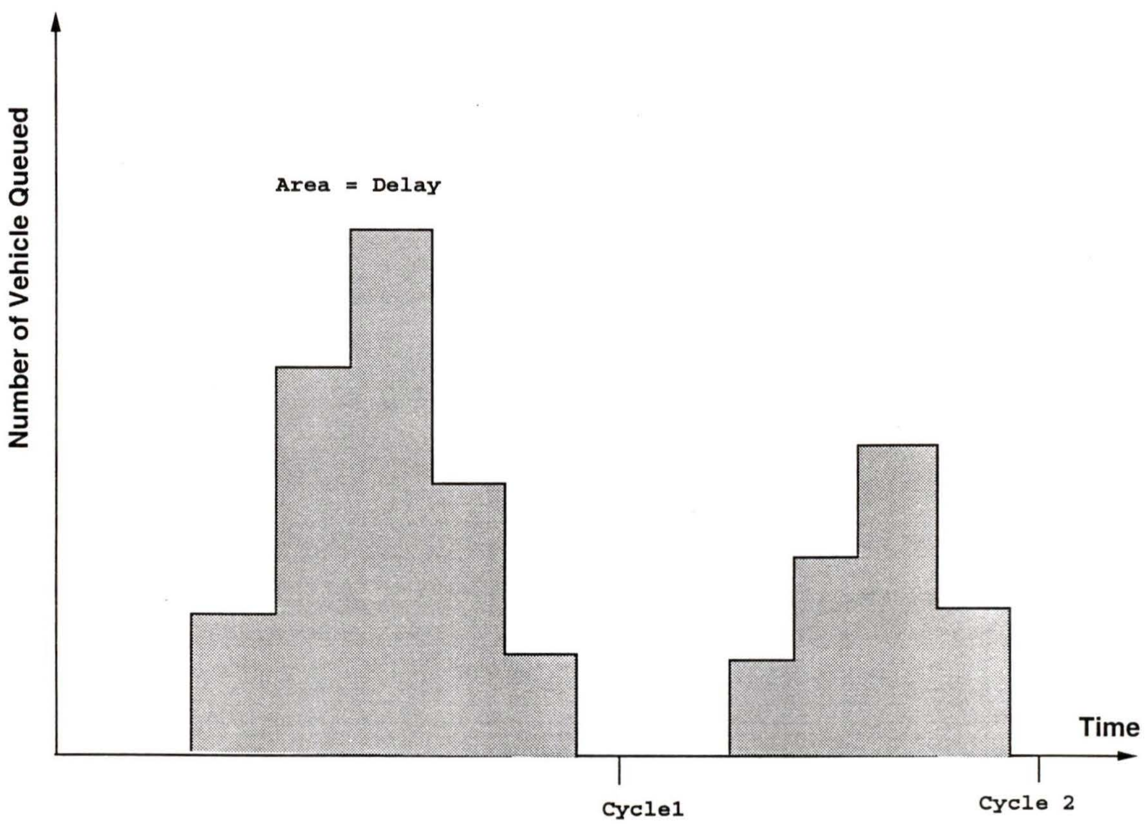


Figure 4.1: Typical Delay Measurement

arrival variation is more severe. For an average cycle, *TRANSYT-7F* estimates traffic delay by calculating the average delay, illustrated in Figure 4.2, where three vehicle traces through a series of intersections at prevailing cruise speed. Vehicle 1 experiences no delay, while vehicles 2 and 3 experience equal amounts of total delay with different number of stops.

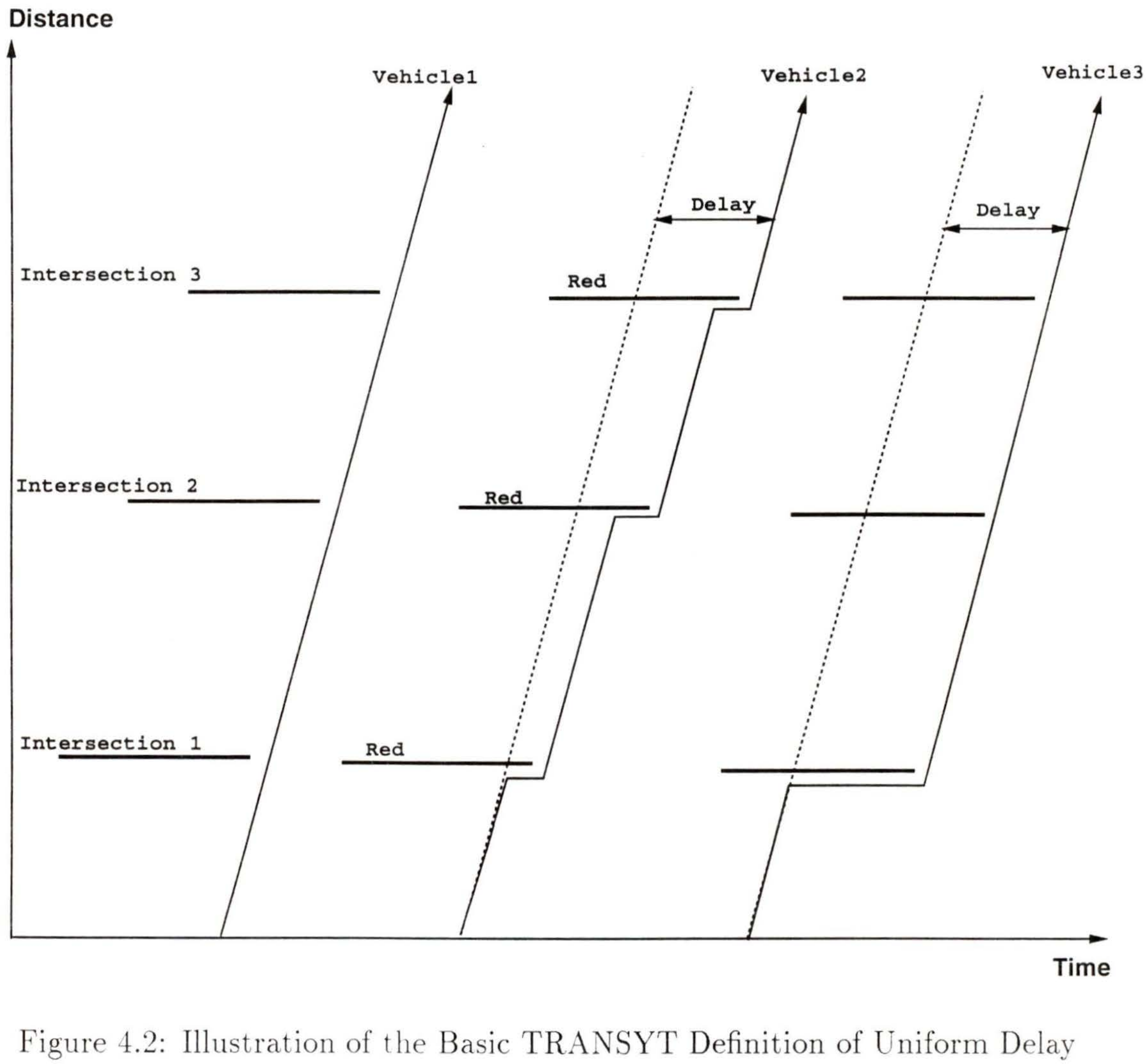


Figure 4.2: Illustration of the Basic TRANSYT Definition of Uniform Delay

While *TRANSYT-7F* does not explicitly trace the trajectories of individual vehicles for estimating these delay, it can, however, obtain macroscopic estimates of delays from the flow profiles. The delay is calculated based upon the flow rate and queue length relations, as illustrated in Figure 4.3. For a signal cycle starting at the beginning of red light, t_0, the inflow rate minus the outflow rate is equal to the inflow up until the time $t_1 + SLT$, where t_1 is the start of the green light and SLT is the vehicle start-up lost time. All arriving vehicles are being stopped and delayed during this time. Accumulating this queue yields the queue length curve depicted conceptually by the dashed line. The traffic rate and queue length are represented and calculated discretely using a fixed small time interval — step, for ease of calculation.

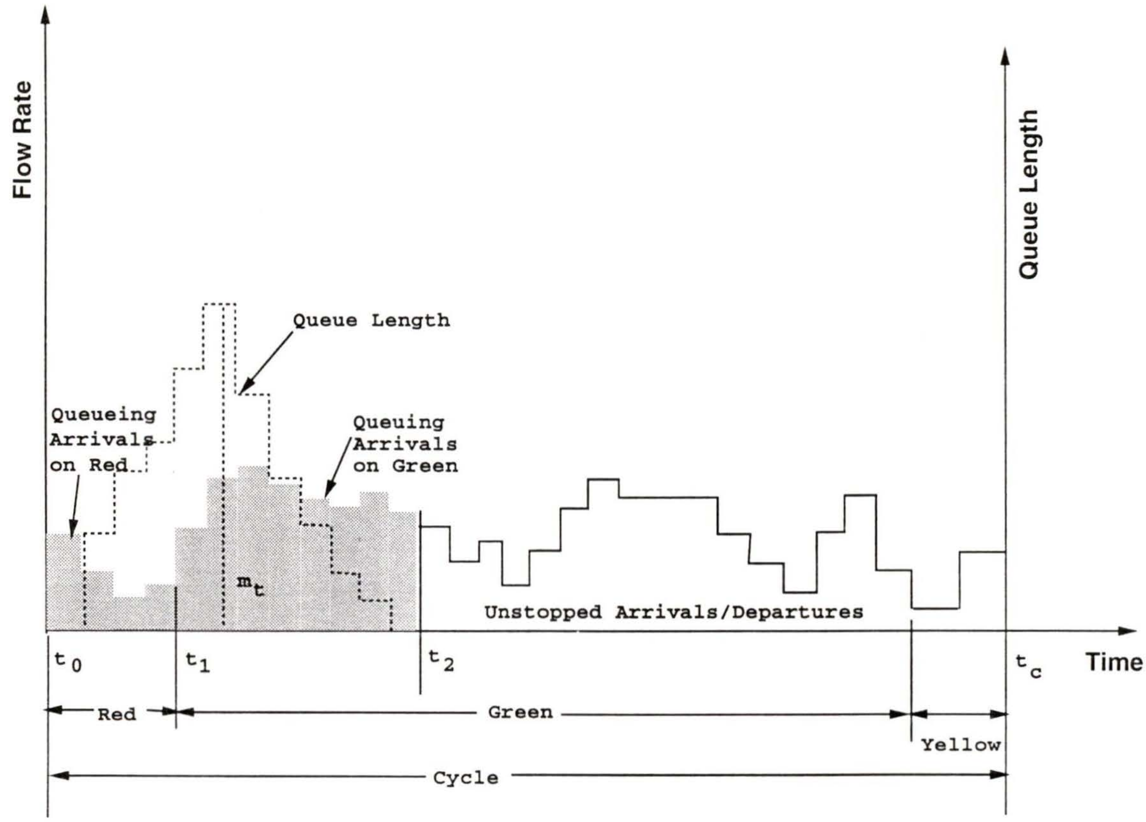


Figure 4.3: Derivation of Uniform Delay

Once traffic begins to move at time, $t_1 + SLT$, vehicles continue to join the end of the queue, but the queue is discharging at a higher rate at the front of the queue. At t_2 the queue has dissipated entirely and no further delay occurs in this cycle. The uniform delay is calculated by integrating the area under the dashed line. which has the effect of averaging the queue length over the cycle. The following equations provide the estimate to the queue length. m_t. for step. t. *TRANSYT-7F* estimates uniform delay by counting the simulated queue length in every step. The uniform delay is the average queue length times the cycle length:

$$D_u = \sum_t^N \frac{m_t}{N} \tag{4.2}$$

where,

D_u — uniform delay (vehicle-hour);

m_t — queue length in vehicles during step t (vehicle);

N — number of steps in the cycle ($t_c/\delta t$);

δt — step time interval (seconds);

t_c — cycle length (seconds)

Delay is also introduced by the random arrivals of vehicles. *TRANSYT-7F* computes the combined effect of random delay and saturation delay, D_{rs} (vehicle-hour), using an experience formula by:

$$D_{rs} = \left[\left[\frac{B_n}{B_d}\right]^2 + \left[\frac{X^2}{B_d}\right]\right]^{1/2} - \frac{B_n}{B_d} \tag{4.3}$$

where,

X — degree of saturation;

$B_n = 2(1 - X) + XZ$

$B_d = 4Z - Z^2$

$Z = (2x/V) * (60/T)$;

v — volume on the link; and

T — period length. normally 60 minutes for unsaturated conditions.

The total delay in vehicle-hour. D. is calculated by:

$$D = D_u + D_{rs} \tag{4.4}$$

4.4.2 Calculation of Traffic Stops

As shown in Figure 4.2. the two vehicles incurring delay are also stopped. Vehicle 2 has three stops and vehicle 3 has one stop, although both are delayed for the same amount of time. In general, the number of stops is related to the length of delay, but not necessarily proportional. *TRANSYT-7F* assumes that delayed vehicles are also stopped.

Figure 4.4 shows a typical time-space arrival/queue/departure condition at an intersection (a *TRANSYT-7*) node of a road (a *TRANSYT-7F* link) under the uniform delay assumption, a vehicle stops instantaneously either at the stopline or at the back of the queue, and restarts instantaneously when the light turns to green.

The spatial and time relation of a queue stopped at a red light is illustrated in Figure 4.4. The vehicles approach the intersection at a same speed. The first vehicle, $vehicle_1$, stops at the stop line. The following vehicles stop at the end of the queue, indicated by the point a and locus of point A for all vehicles, L(A). When the traffic light turns green all vehicles start to move after a short delay of SLT (start lost time), at the time specified by point B and the locus of B, L(B).

This module is based upon instant stop and restart. The actual stop and restart of a vehicle represents a decelerate and acceleration period, or a partial stop, as indicated by the dash curves at points A and B.

TRANSYT-7F takes into account the partial stops by calculating the effective stops using a filtering algorithm. Studies suggested that the short delay can be expressed as fractions of stops for the vehicles affected. Since the traf-

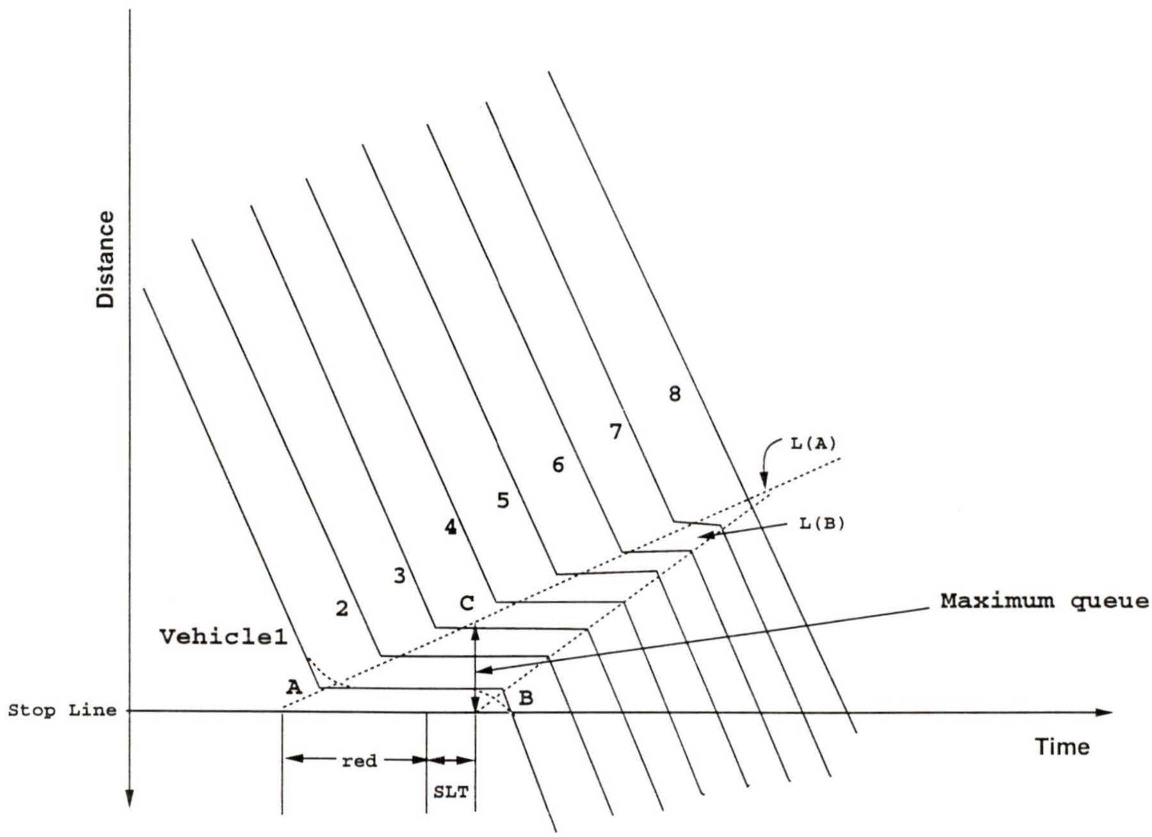


Figure 4.4: Derivation of Stops and Maximum Back of Queue

fic model actually uses the flow profiles (Figure 4.3) instead of the conceptual representation shown in Figure 4.4, the number of stops can thus be counted by the length of delay. The empirical studying conducted by the Transport and Road Research Laboratory (TRRL) in the United Kingdom produced the relationship between the percentage of stops and length of delay, as shown in Figure 4.5. *TRANSYT-7F* has a built-in function to calculate the length of delay based upon the percentage of the vehicle stops.

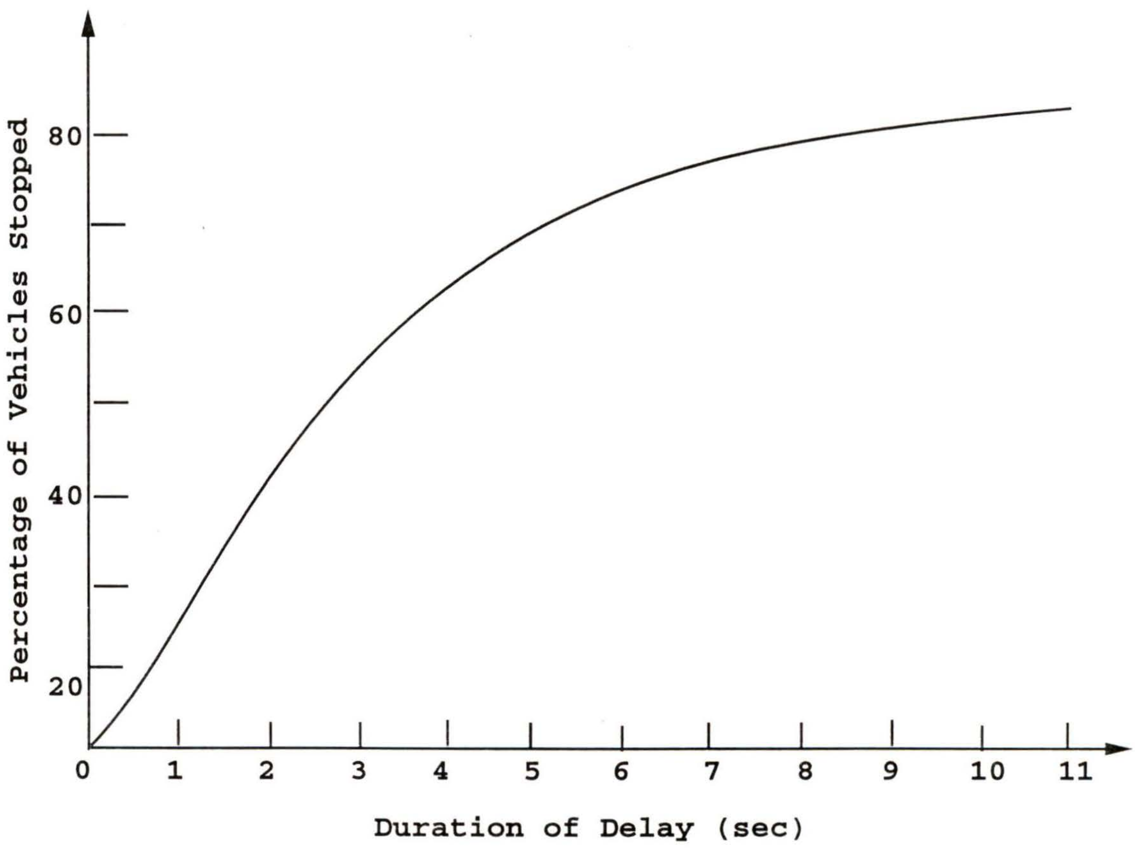


Figure 4.5: Reduction of Stops as a Function of Length of Delay

TRANSYT-7F calculates the number of stops by multiplying the percentage of vehicles stopped by the number of vehicles that leave the stopline. When a link volume (or length of a queue) exceeds the link capacity, and the stopped size of the vehicle storage buffer at the intersection, not all vehicles that arrive on green can leave on the same cycle. Since *TRANSYT-7F* simulates only one cycle, the number of stops calculated, based on the above multiplication, does not take into consideration the extra vehicles that arrive but could not leave during the green. These extra vehicles are assumed to experience "full stops." Thus, the total stops under saturated conditions are obtained by summing the number of the extra vehicles. The addition of the extra vehicles on the number of stops calculated is shown in Figure 4.6. At the present time, *TRANSYT-7F* cannot accurately take into account the random and saturation stops. These types of stops might be considered.

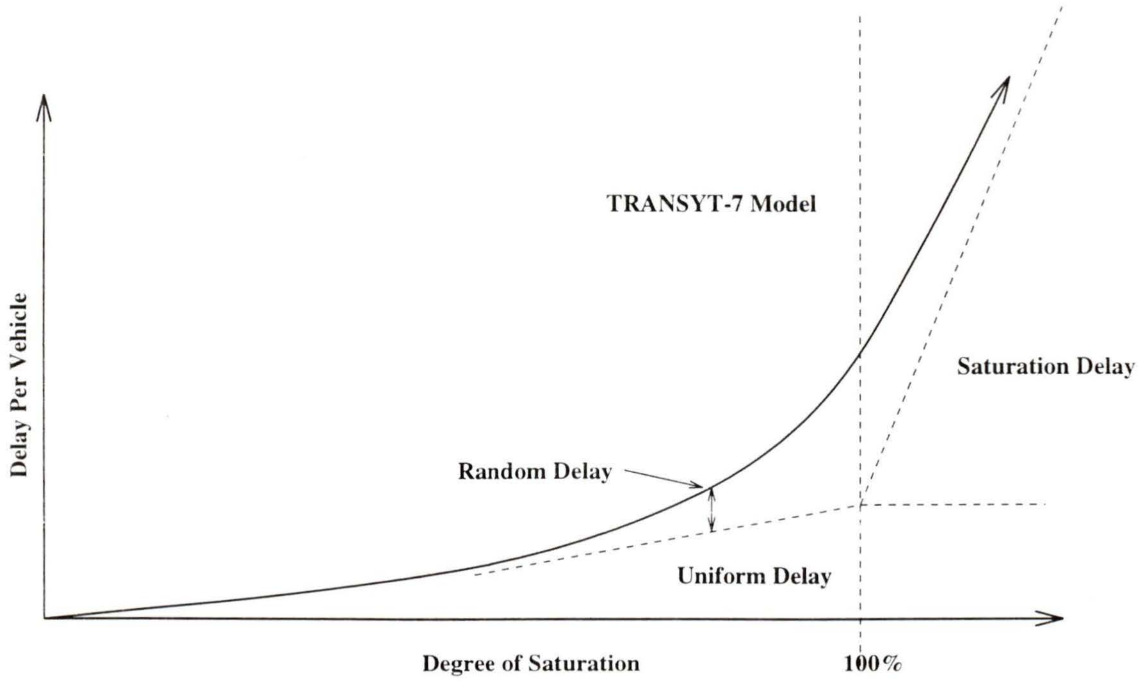


Figure 4.6: Degree of Saturation

4.4.3 Maximum Back of Queue

Conceptually, the queue is the vertical "distance" between $L(A)$ and $L(B)$ in Figure 4.3. *TRANSYT-7F* actually reports point C, which is the maximum back of the queue, in terms of the number of vehicles. This figure is more informative than the maximum queue length because we are primarily concerned with whether the queue will be too long and will "spillover" into the upstream intersection. *TRANSYT-7F* does not consider this matter explicitly, and disregards the effect of spillover. *TRANSYT-7F* does, however, report a queue capacity value for each link in the traffic network. By comparing the maximum back of queue value with the queue capacity value, one can easily identify where a spillover may occur. The queue capacity value function may be defined on a link-by-link basis, and the program will calculate the queue capacity value for the selected link. If one does not specify explicit queue length value, the program will calculate the queue capacity value, QC_i, by:

$$QC_i = \frac{P_{qi} \times L_i}{N_i \times H} \tag{4.5}$$

where,

P_{qi} — the proportion of the link "length" as a percentage of total link length with a given specify acceptable storage capacity (the default is 80 percent);

L_i — length of link i in feet (m);

N_i — the number of lanes on link i, derived from the coded saturation flow rate;

H — average vehicle spacing in the queue, normally 25 ft . or 7.6 m

4.5 A New Traffic signal control Optimization Formulation

The *TRANSYT-7F* timing plan design program, as a macroview cycle-based, traffic flow simulation and optimization program, identifies the optimal control parameters using a mutiple-objective optimization.

The optimization problem is formulated using a linear combination of total delay, stop, and queue length as the objective function; it is called performance index. These competitive objectives are weighted using three conflicts to be determined by user.

The optimization leads to minimum traffic delay for vehicles on the highways. However, this minimized traffic delay is achieved by maximizing the bandwidth controlled by the parameters. Often, the bandwidth is unnecessarily wide to achieve the minimum delay in the artery direction. This may cause unacceptable traffic delay in the cross direction. If we can find the minimum useful bandwidth in the artery direction based upon the actual traffic platoon, then the traffic delay minimization will thus achieve good performance for both artery and cross directions. The minimum useful bandwidth, BW^*, can be calculated by:

$$BW^* = (\frac{VC}{S} + l) \times F_{dis} \tag{4.6}$$

where, V — vehicle volume (vehicle/hour); C — cycle length (sec); S — saturated flow rate (vehicle/hour); l — lost time (sec) , and F_{dis} — platoon disperse factor.

This bandwidth value is then fixed when the traffic delay is minimized. The optimal problem is changed to a single objective problem with an additional equation constraint. Better overall performance of the traffic system can be thus achieved.

The optimization is implemented by a dedicated C program, ATPCP, that interfaces with the *TRANSYT-7F* timing plan generation program.

Chapter 5

Short-term Traffic Demand Prediction

5.1 A Brief Overview

The optimal timing plan identification method. introduced in the previous chapters. senses real traffic demand and identifies the best-fit traffic pattern with stored optimal traffic signal control parameters. The method has significantly reduced the traffic plan lead times and made dynamic traffic control possible.

However, traffic signal control should be based upon the traffic demand in the next instant (in terms of control period) rather than the past demand sensed from road.

A fast and efficient short-term traffic demand prediction algorithm is introduced in the work to allow traffic signal control to catch up the dynamic variation of traffic demands. The traffic condition at each moment is represented by a hyper-point in the m-dimensional traffic volume space. where m is the number movements of the traffic system. The locus of this hyper-points forms a continuous curve of time. In this work. most recently measured traffic data are regarded as *current data*. and the average of all collected traffic data are regarded as *historical data.*

The acquired traffic data are composed of a dominant (or basic) part and a random part. The dominant part normally varies from zero to thousands vehicles per hour. and the random part varies from zero to hundreds of vehicles per hour.

Trend prediction has been researched for several decades, and no efficient traffic demand prediction method has been found so far. Most of the previously developed traffic trend prediction methods are based upon a fixed mathematical model and the model parameters determined using historical traffic data by the least-square fitting. These methods share two major shortcomings, (a) requirement of extensive computation when the mathematical model is complex, thus having no time for implementing dynamic traffic signal control, (b) equal weights on all historical data with no emphasis on more influential, recently acquired data [6].

To overcome these shortcomings, a method considering both average traffic demand (historical data) and instant traffic demand (current data) using the weighted average method is introduced, Figure 5.1 illustrates how to obtain the predicted traffic demand from the historical and current data.

In Figure 5.1, the dashed curve represents the average (or historical) traffic demand obtained by averaging all historical traffic demand. The average traffic demand at 7:45 am can be retrieved directly. The current demand that is sensed right before the control instant is plotted as the solid curve with 7:00 am, 7:15 am, and 7:30 am as three sample points. The traffic demand at 7:45 am, if determined purely based upon the current demand data, using the three point weighted average method, will be at point $X_{c(t+1)}$. We call $X_{c(t+1)}$ the weighted-average prediction. The calculation of the three point weighted average method is frequently used in forecasting, and is explained in section 5.2. To better predict traffic variation, the historical demand and the current demand are both considered. Their influences to the further demand are counted by a vector sum:

$$X_{p(t+1)} = \alpha X_{c(t+1)} + \beta X_{h(t+1)} \tag{5.1}$$

$$\alpha + \beta = 1 \tag{5.2}$$

$$0 \leq \alpha, \beta \leq 1 \tag{5.3}$$

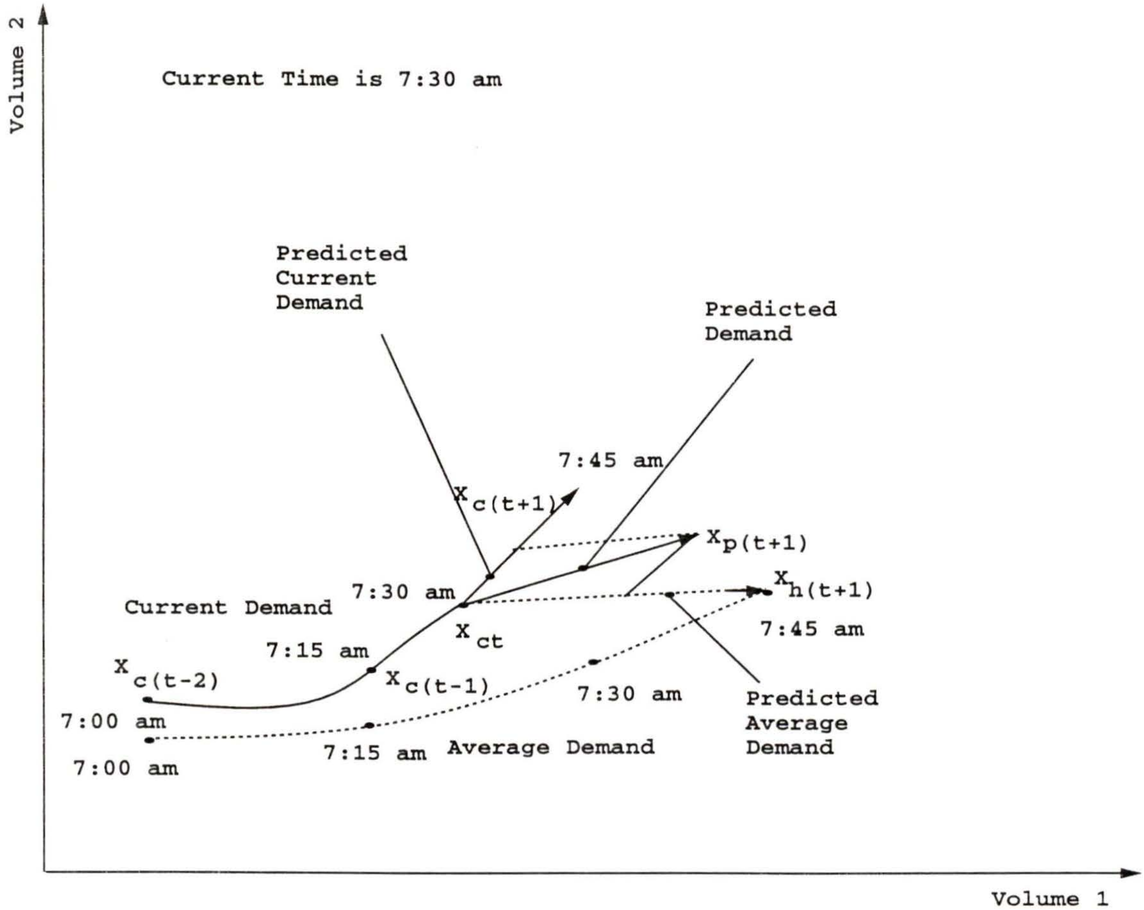


Figure 5.1: Prediction Algorithm Description

where, $X_{h(t+1)}$ is the historical demand at 7:45 am; $X_{c(t+1)}$ is the simple weighted-average prediction of current traffic demand at 7:45 am. The vector $X_{p(t+1)}$ is the modified predicted traffic demand, and $X_{p(t+1)}$ is also a hyper-point in the traffic volume space. α and β are two convex weighting factors to be discussed in the following sections.

5.2 Calculation of Current Traffic Demand

As discussed presently the modified predicted traffic demand $X_{p(t+1)}$ is the weighted vector sum of the two vector $X_{c(t+1)}$ and $X_{h(t+1)}$. The simple weighted-average prediction, $X_{c(t+1)}$, is determined using the three point weighted average method for trend forecasting:

$$X_{c(t+1)} = \zeta^2 X_{ct} + \zeta\eta X_{c(t-1)} + \eta^2 X_{c(t-2)} \tag{5.4}$$

$$\zeta + \eta = 1 \tag{5.5}$$

$$\zeta \geq \eta \geq 0 \tag{5.6}$$

where,

$X_{c(t+1)} = \{x_{ci(t+1)} | i = 1, 2, \ldots, n \text{ (n — movement number)}\}$, and the $x_{ci(t+1)}$ is the predicted traffic volume of the ith movement at the $t+1$ moment;

$X_{ct} = \{x_{cit} | i = 1, 2, \ldots, n\}$, the x_{cit} is the measured traffic volume of the ith movement at the t moment;

$X_{c(t-1)} = \{x_{ci(t-1)} | i = 1, 2, \ldots, n\}$, the $x_{ci(t-1)}$ is the measured traffic volume of the ith movement at the $t-1$ moment;

$X_{c(t-2)} = \{x_{ci(t-2)} | i = 1, 2, \ldots, n\}$, the $x_{ci(t-2)}$ is the measured traffic volume of the ith movement at the $t-2$ moment.

In this work the weighting factors, ζ and η, are chosen to be 0.78 and 0.22 accordingly, due to the heavy influence of "recent" current traffic data point.

This is based upon the selection of $\zeta^2 = 0.6$, $\zeta\eta = 0.3$, $\eta^2 = 0.1$. The decreasing values of the three coefficients allow more weight being put on the recent traffic data. This chose of weighting factor is commonly used in forecasting and is recommended by [4]

5.3 Weighting Factors for Traffic Demand Prediction

The short-term traffic demand prediction model, presented in Eq. (5.3), is a linear combination of two terms. One term includes the recently measured traffic data, and the other is the historical average of the traffic volume at the time of prediction. The weighting factors, α and β, are determined using two alternative methods:

Fixed values — The method is lock of flexibility for varying influences of current and historical traffic data, but works better under large traffic volume noise.

An extensive test has been carried out to identify the best α and β values, based upon the traffic data of five representative days (from the Ministry of Transportation and highways, B.C.). The test data come from five intersections contain 96 sample data points. The test based upon these 28,800 traffic samples demonstrates that 0.7 and 0.3 are the best values for α and β, respectively. A comparison of prediction errors of different α and β values is illustrated in Table 5.1. The case where, $\alpha = 0$ and $\beta = 1$ represents the average prediction method. Variation of α and β is not sensitive around the optimum.

Adaptive weighting factors — Another method for predicting the short-term traffic demand is to use two adaptive factors, α and β, in the linear combination of current demand prediction and the average demand prediction. Instead of two fixed values, the values of α and β are determined based upon the differences between the sensed, real traffic volume at present time t, X_{ct} and the current demand prediction, X_{pt}, and the average demand prediction, X_{ht}. If X_{pt} is closer to X_{ct}, α value increases, and if X_{ct} is closer to X_{ht}, β value increases. The method is superior when the traffic volume changes following a stable trend, but functions less well in the presence of large traffic noise. The approach is adopted in this work.

Table 5.1: Fixed Weight Factor Test

α	β	Total Predicted Error (vehicle)
0.8	0.2	225,794
0.7	0.3	201,364
0.6	0.4	217,553
0.5	0.5	257,779
0.0	1.0	454,371

The weighting factors are calculated by:

$$\alpha = \frac{|x_{pit} - x_{hit}|}{|x_{cit} - x_{hit}| + |x_{cit} - x_{pit}|} \tag{5.7}$$

$$\beta = \frac{|x_{pit} - x_{cit}|}{|x_{cit} - x_{hit}| + |x_{cit} - x_{pit}|} \tag{5.8}$$

where, x_{pit} the measured traffic volume of the ith movement at time t; x_{hit} the averages traffic volume from historical data of the ith movement at time t; x_{cit} the predicted traffic volume of the ith movement at time t.

5.4 Testing Results

5.4.1 Fixed Weighting Factor Method

In addition to its simplicity, the fixed weighting factor method presents better performance for constantly varying traffic volume and large noise, due to its emphasis on the traffic average rather than last prediction result as adaptive weighting do. This advantage can be illustrated by the traffic flow simulation of the Duncan system at the Boys Road intersection (north bound through) in the fall of 1992, shown in Figure 5.2.

The total and average errors with respect to real traffic demand (sensed) of the simple average prediction method, based upon only historical average traffic data, and the fixed weighting factor prediction method are listed in Table 5.2. In Figure 5.2 the total errors are the difference between real (or measured) volumes and predicted volumes. The average error is:

$$average(\frac{real\ volume - predicted\ volume}{real\ volume})$$

. The fixed weighting factor method demonstrated considerable error reduction:

$$Error\ Reduction = \frac{average\ prediction\ error - fixed\ weghting\ prediction\ error}{average\ prediction\ error} = 27\%$$

Table 5.2: Prediction Errors of Historical Average and Fixed Weighting Factor Prediction (Based upon 14683 Vehicles at This Movement in A Day)

Prediction Method	Total Error (vehicles)	Average Error	Improvement (%)
Average	3248	22.4 %	1
Fixed Weight	1848	16.3 %	0.73

5.4.2 Adaptive Weighting Factor Method

The adaptive weighting factor method considers the performance of the demand prediction at the previous instant. The quality of the last prediction is used as a factor. The fitness of the average traffic prediction and current traffic prediction to the real traffic demand at previous instant is used to decide their contributions to the actual prediction in the next instance. However, when large noise present and no definite fitness can be determined. the method may not work well. The adaptive weighting factor method is illustrated by the traffic flow simulation of the Duncan system at the Boys Road intersection (north bound through) in the fall of 1992. shown in Figure 5.3.

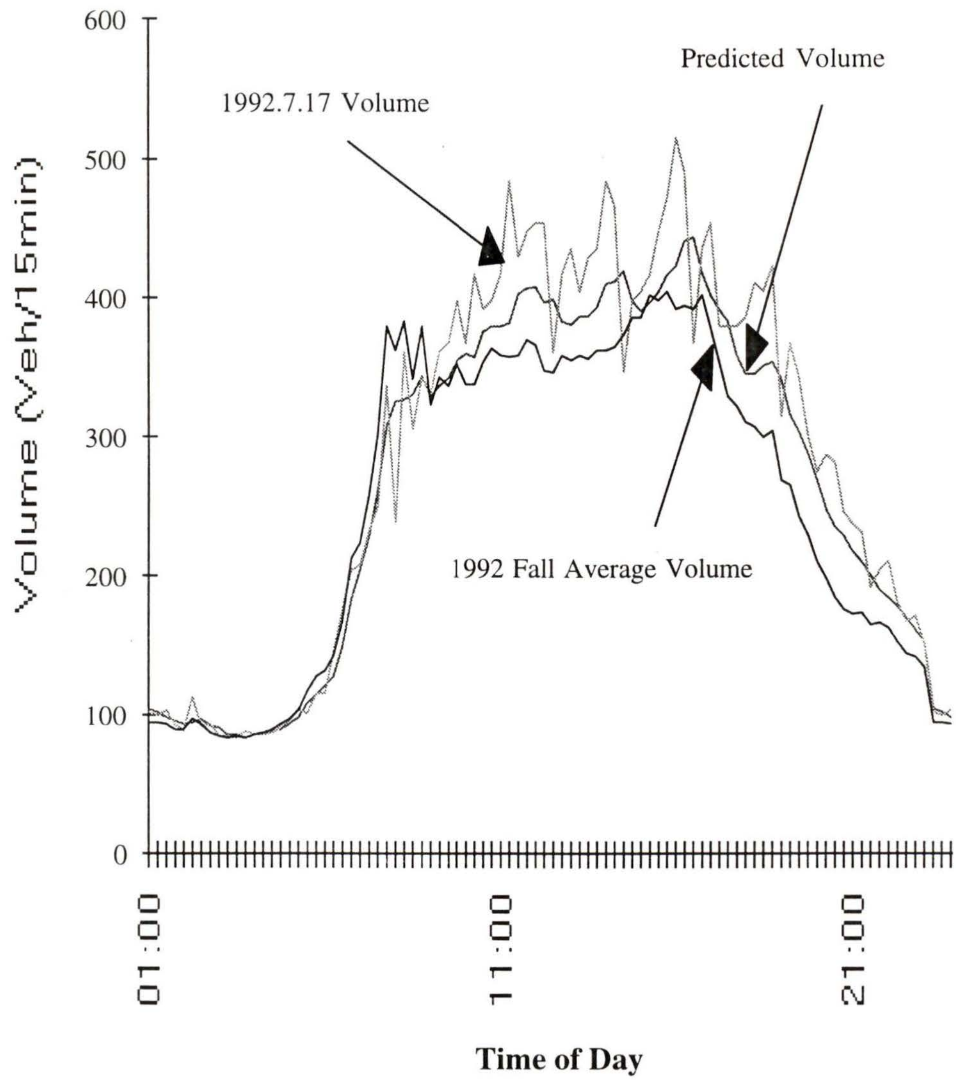


Figure 5.2: Fixed Weighting Traffic Flow Prediction Simulation I (Duncan at Boys Road., Northbound through)

The total and average errors with respect to real traffic demand (sensed) of the simple average prediction method and the adaptive weighting factor prediction method are listed in Table 5.3. The adaptive weighting factor method demonstrated considerable error reduction to be 37 percent

Table 5.3: Comparison between Historical Average and Adaptive Weight Prediction

Prediction Method	Total Error (vehicle)	Average Error	Improvement (%)
Average	3248	22.4 %	1
Adaptive Weight	1884	14.1 %	0.63

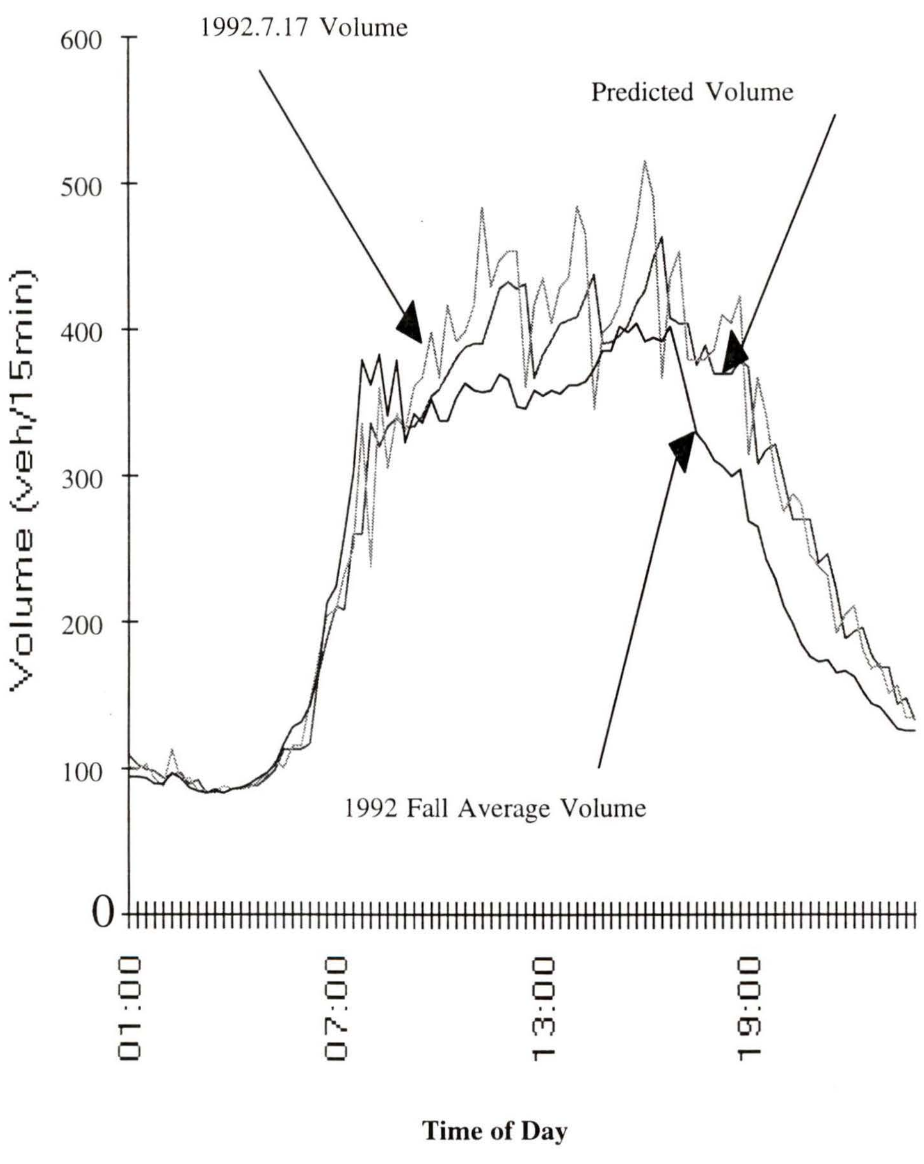


Figure 5.3: Adaptive Weighting Traffic Flow Prediction Simulation I (Duncan at Boys Road., Northbound Through)

Chapter 6

Fuzzy-Neural Dynamic Traffic Decision Making

6.1 An Introduction to Fuzzy-Neural Networks

6.1.1 Artificial Neural Networks (ANN)

An artificial neural network (ANN) is a mathematical model that has been developed to emulate the human neuron system. An ANN consists of nodes that associate with values, and links that associate with weights. The nodes are arranged in layers that include an input layer (the lowest layer), an output layer (the top layer) and several hidden layers. These layers are inter-connected by links. The inputs to the lower layer nodes propagate to higher layer nodes through weighted links.

Each node in the network is a nonlinear, analog computing unit. An intermediate node sums N weighted inputs using simple calculations and passes the result up to higher layers. The weights attached to the links of the network are determined by giving a set of inputs and outputs, and running a network training program. The ANN automatically adjusts the weights during the training to match the outputs from the top layer to the given outputs. A typical three layers ANN is illustrated in Figure 6.1.

In recent years, ANN has found many successful industrial applications, due to its superior capabilities in computation and pattern recognition. With its massive parallel processors with simple computation logic, an artificial neural network or a neural computer can process a large amount of data quickly, considerably improving the computation speed for many application. An ANN with

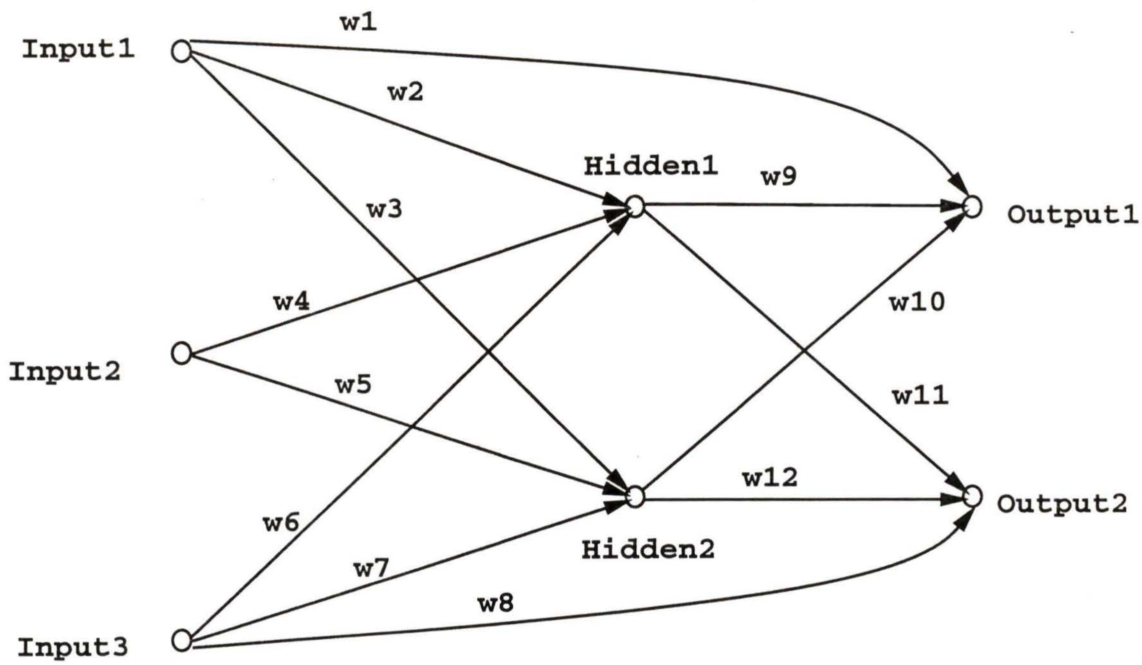


Figure 6.1: A Typical Three Layers ANN

its training algorithm also demonstrates a "learning" capability. The network, once trained, using given inputs and outputs, will be able to generate reasonably good results for other inputs without requiring an explicit model of the physical system. This capability is particularly useful for those engineering tasks such as hand character recognition and image pattern recognition, where perfect modeling of the system is difficult.

In dynamic traffic signal control, the real-time control decision-making task requires fast data processing speed. In addition, the dynamic behavior of a traffic system is very complex and difficult to model in an explicit form. These properties of dynamic traffic signal control make it a good application of ANN, with the exception of its requirement of a good fuzzy knowledge reasoning.

In reality, dynamic traffic demand has an uncertain component, as discussed in Chapter 3. Fuzzy pattern clustering and recognition allow us to carry out fuzzy knowledge reasoning, based upon uncertain traffic data to identify the optimal traffic signal control parameters. The fuzzy system and the ANN system, if combined as a fuzzy-neural system, will overcome the drawbacks of each pure system which make the development of the intelligent dynamic traffic signal control system very difficult. On the other hand, a pure neural system will not be able to provide explicit fuzzy pattern classification, or to handle the fuzziness and uncertainty of a traffic system. Furthermore, the combined fuzzy-neural system can support fuzzy knowledge reasoning that provides more functions to the intelligent system, even though this capability waste not used in this work.

6.1.2 Fuzzy-Neural Networks

The fuzzy decision-making mechanism applies a number of membership function as the reasoning rules. The difficulty in using fuzzy logic inference lies in the determination of the multiple dimensional fuzzy membership functions. Unlike the one-dimension fuzzy problem, it is very difficult to determine the multiple dimensional membership functions manually, based upon the statistical data and experience. Moreover the determined membership functions can not be adjusted according to changing traffic conditions.

The fuzzy-neural network developed in this work is determined by obtaining the fuzzy membership functions, by fitting the system inputs and outputs without requiring any knowledge on the form of these membership functions. Two types of fuzzy neural networks are used today. One uses fuzzy neurons to substitute the weighting elements [12], where $f_1(x), f_2(x), \ldots, f_n(x)$ are fuzzy membership functions as shown in Figure 6.2. The other embeds neural networks into the fuzzy reasoning system [36], as shown in Figure 6.3.

In this work, the second type of fuzzy-neural networks is adopted. The system embeds neural networks into the fuzzy reasoning graph. At each state (or node) of the graph a neural network is used, as shown in Figure 6.3. Each

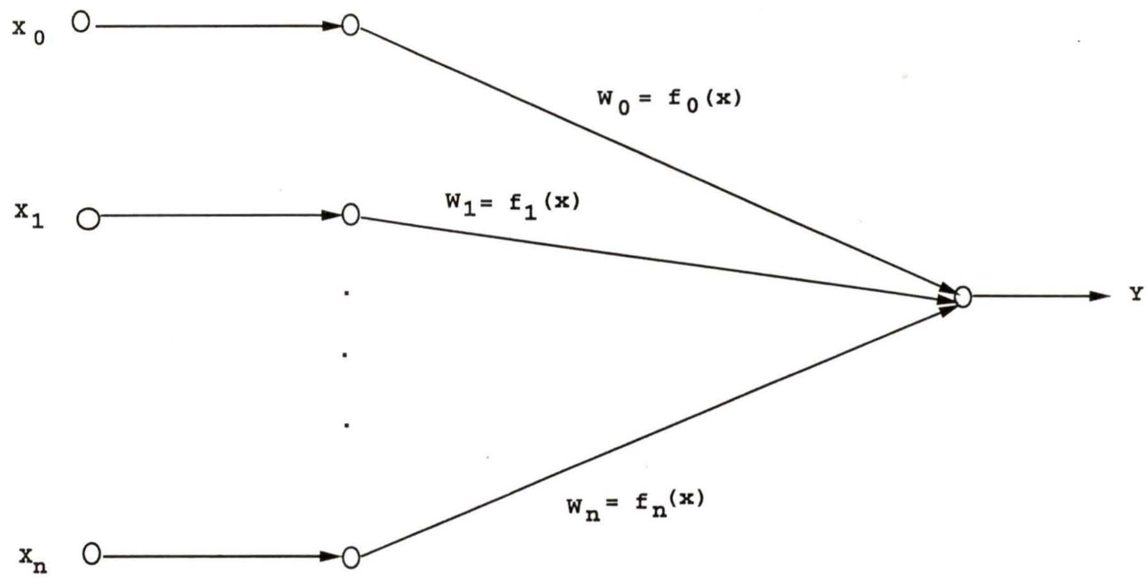


Figure 6.2: Fuzzy-neural Network with Fuzzy Neurons Substituting the Weighting Elements

layer of the neural network sums N weighted inputs and produces outputs at next layer through a nonlinear function:

$$y_i = f(\sum_{i=0}^{N-1} w_i x_i) \tag{6.1}$$

where, f is a nonlinear function in the form of hard limits, threshold logic, and sigmoid. The input-output relation of an ANN node and the form of the stated mapping functions are illustrated in Figure 6.4.

6.1.3 A Taxonomy of Neural Nets Used for Pattern Classification

Several types of ANN structure have been introduced for different applications. A taxonomy of ANNs used for pattern classification is presented in Figure 6.5 [16].

The taxonomy divides various types of ANN into two groups: those with binary and those with continuous valued inputs. Each group is further classified into nets that either require or do not require supervision during their training.

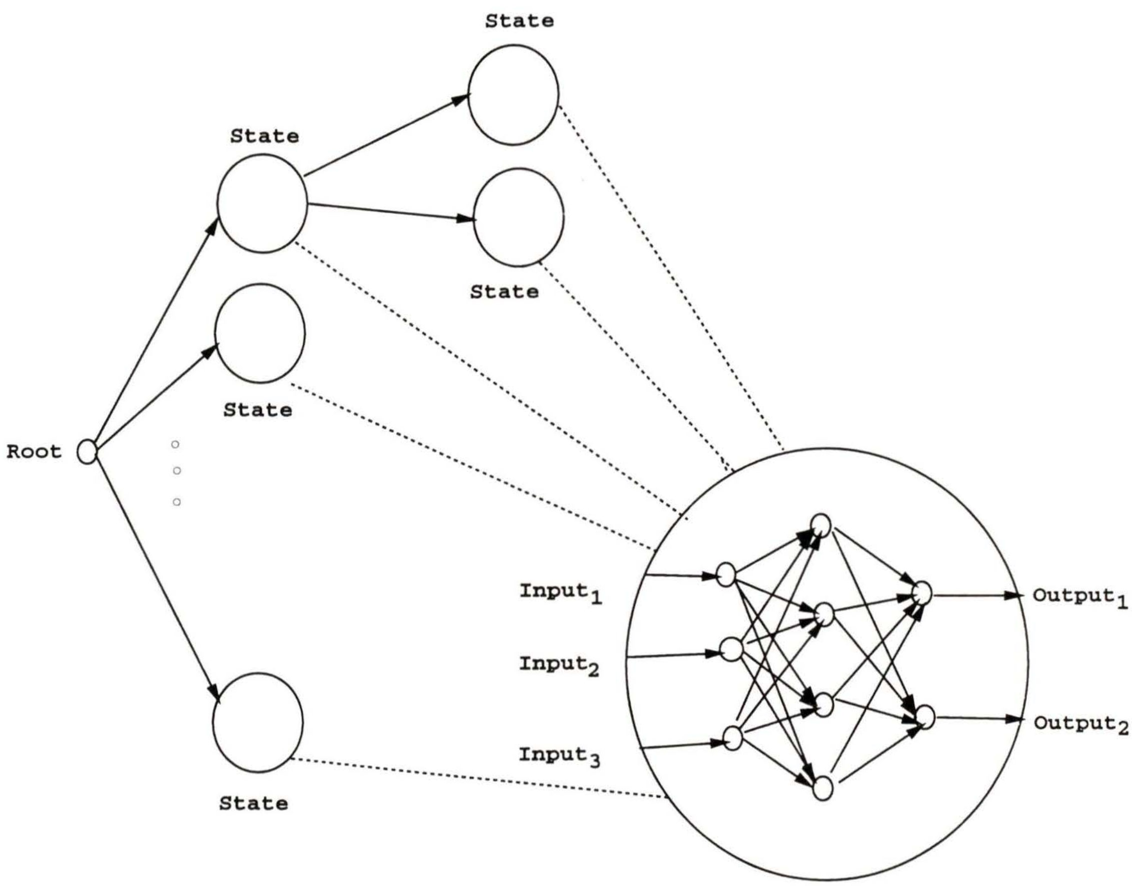


Figure 6.3: Fuzzy-neural Network Which Embed the Neural Network into Fuzzy Reasoning System

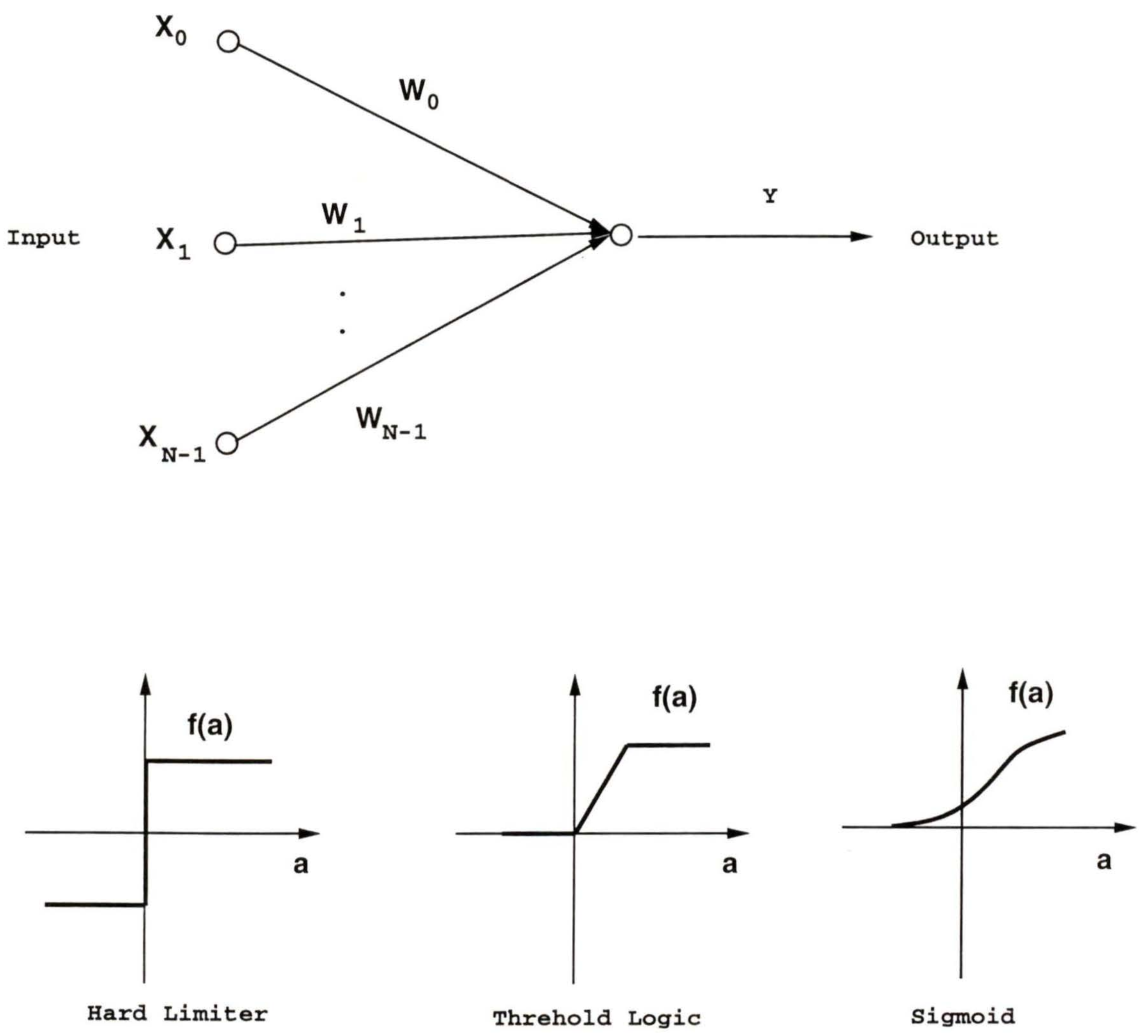


Figure 6.4: Computational Element or Node Which Forms a Weighted Sum of N Inputs and Passes the Result through a Nonlinearity

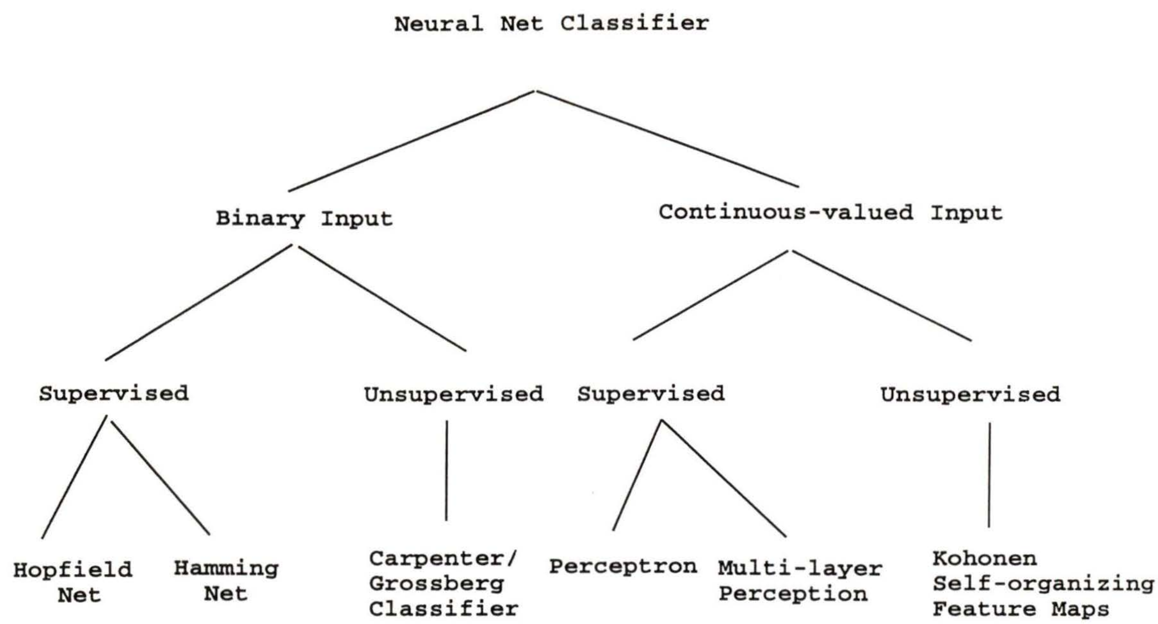


Figure 6.5: A Taxonomy of Six Types of ANNs Used as Classifier

ANN requiring supervised training includes the Hopfield net and Hamming net for binary inputs and perceptron nets [30] for continuous inputs. During the supervised training, information specifying the correct class of new input patterns is provided to determine the weight of the network. On the other hand, the ANN trained without supervision, such as the Kohonen's feature map forming nets [15], require no information concerning the correct class during the training, and the nets are used as vector quantifiers to form better clusters.

6.1.4 Continuous Input and Supervised Neural Network

In the intelligent dynamic traffic signal control approach proposed in this work, the traffic system provides continuous traffic volume inputs, and the fuzzy traffic pattern recognition function module provides the mapping from clustered traffic volume inputs to the best-fit traffic pattern output. The input-to-output relations provide ideal information for training the neural network. The neural network with continuous input and supervised training, namely the multilayer perception nets, is adopted in this work to form the fuzzy-neural traffic pattern matching function module.

The boundary of pattern region division is associated with the number of ANN layers. A neural net with more layers normally supports decision regions with more complex shapes, with more accurate mapping functions. The one-layer net can only process half plane boundary divided by hyperplane. The two-layer net can process convex open or closed region, and the three layer net is considered to be able to handle most mappings [22], and is used in this work due to its simplicity. The neural decision region is illustrated in Figure 6.6.

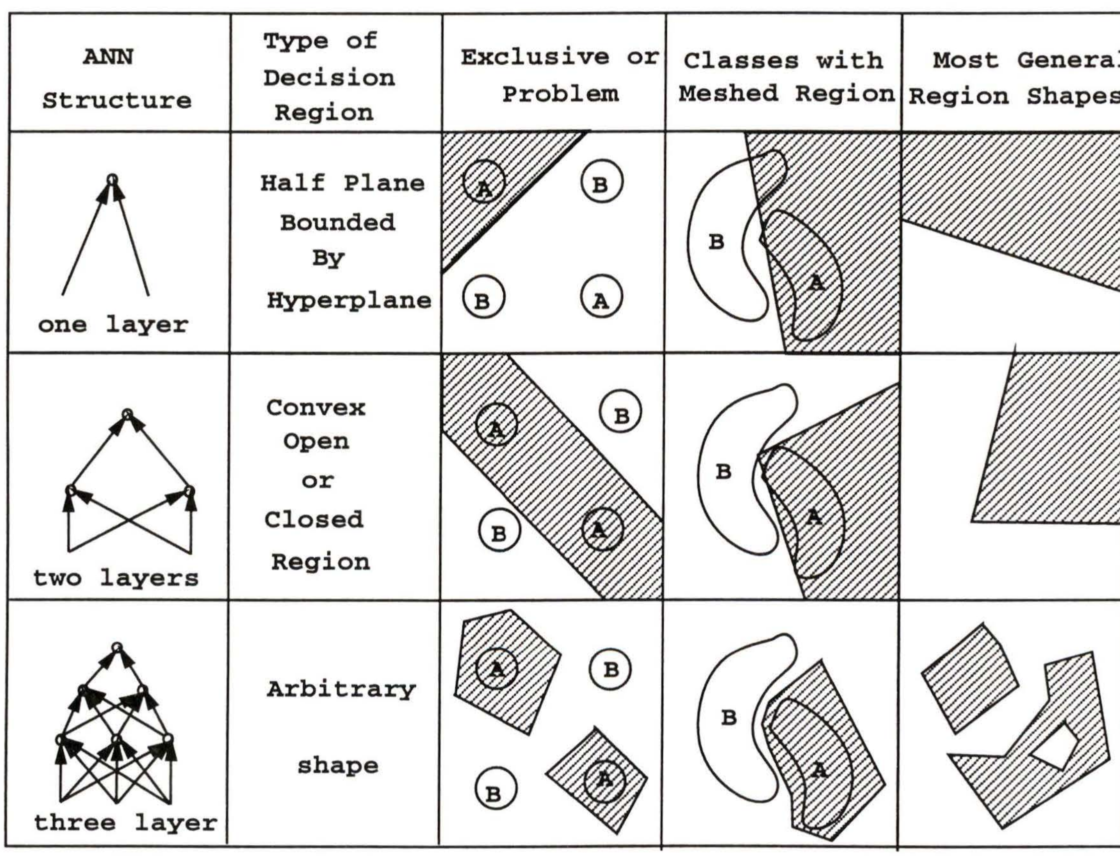


Figure 6.6: Types of Decision Regions

6.2 ANN Training and The Back-propagation Algorithm

The supervised training of the three-layer neural network is carried out by minimizing the mean square difference between the given input-output relations that are determined by the fuzzy traffic pattern classification, and the input-output relations determined by the network. For each input, the difference of the two types of output is calculated by the forward propagation of the network. This difference is minimized through many iterations of the network by adjusting the weights using the gradient search optimization scheme.

The net is trained by initially specifying small random weights, w_i, and internal thresholds for each output, y_i. The training is carried out using all available input-output relations, until the weights of the network finally converge to fixed values and the mean square difference of outputs are reduced to an acceptable value.

The gradient search-based backpropagation training algorithm requires a continuous differential function between the input and output of each neural network layer. The sigmoid logistic nonlinearity, given in Eq. (6.2), satisfying this requirement well represents the nonlinear input-output relations of the physical system.

The multi-layer neural net has many advantages over the single-layer neural net and overcomes many limitations of the single-layer neural net. The multi-layer neural net has found more applications today, due to the introduction and improvement of the effective training algorithm — the back-propagation training algorithm [32].

The input-output relation of a specific neural net layer is represented as:

$$y_j = f_j(\alpha) = \frac{1}{1 + e^{-(\alpha - \theta)}} \tag{6.2}$$

where, $\alpha = \sum_{i=1}^{N} w_{ij} x_i$; x_i $(i = N)$ are N lower layer inputs; w_{ij} are the weights on the links from lower layer nodes j ($i = 1, \ldots, N$) to the upper node j; N is the number of input of the network layer; $f_j(\alpha)$ is the network layer output at node jth; θ is called the *bias*, and $\theta = 1$.

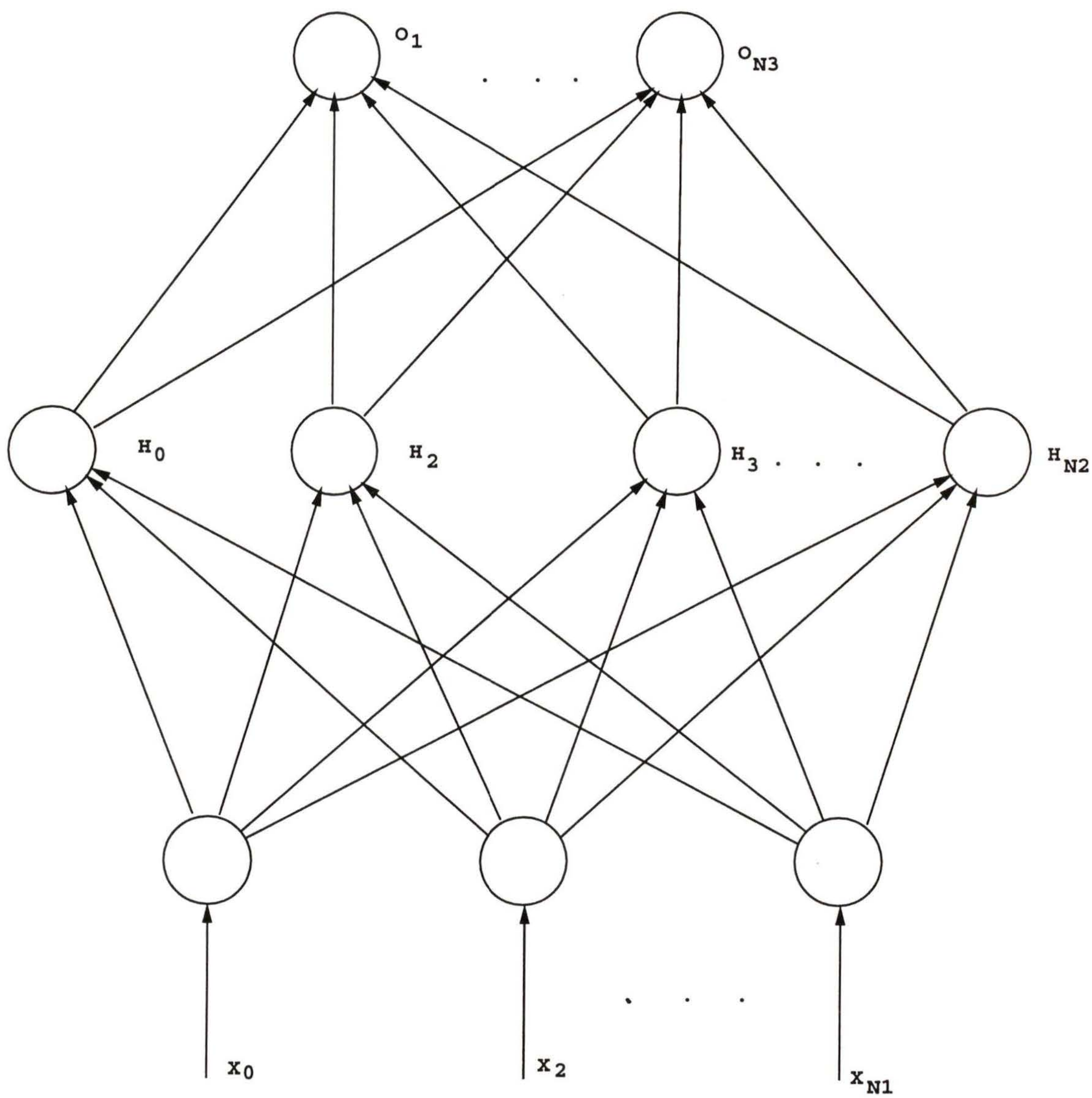


Figure 6.7: Structure of A Three Layer Neural Network Structure

Given the nonlinear layer mapping function of Eq. (6.2), and a set of input-output pairs applied to the outer (first and third) layers of the network, the weights of a three-layer neural network can be determined using the Back-propagation Algorithm, through the procedure to be discussed.

Before the introduction of the algorithm, a number of variables used in the algorithm are defined as follows:

Let N_1 be the number of nodes at the input (or first) layer, and N_3 be the number of nodes at the output (or third) layer of the three layer neural net. N_1 and N_3 are the number of inputs and outputs to the neural net respectively. Similarly, N_2 represents the number of nodes at the hidden (or second) layer of the network, and its value is chosen by the user to achieve best system performance. In general, a larger hidden layer results in a more capable network for more complex nonlinear system modeling. It also leads to a more complex network which is slow in speed and difficult to train [28, 33, 37]. As shown in Figure 6.7, each of the input and hidden layers has an extra node, x_0 and $y_0^{[i]}$, used for thresholding, where the $y_j^{[i]}$ is the ith layer the jth node output. The nodes at these two layers are thus indexed from 0 to N_1 and N_2 respectively. A set of training data is composed of two vectors, $X = [X_1, \ldots, X_{N_1}]^T$ and $Y = [Y_1, \ldots, Y_{N_3}]^T$. X_j are given inputs to the first layer of the network, and Y_j are the desired outputs from the third layer of the network. The second and the third layer outputs are expressed as $y_j^{[2]}$ and $y_j^{[3]}$, respectively.

1. Assign initial weights of the network. Each weight should be set randomly to a number between -0.1 and 0.1.

$$w_{ij}^{[1]} = random(-0.1, 0.1) \quad \forall i = 0, ..., N_1, j = 1, ..., N_2 \tag{6.3}$$

$$w_{ij}^{[2]} = random(-0.1, 0.1) \quad \forall i = 0, ..., N_2, j = 1, ..., N_3 \tag{6.4}$$

where, $w_{ij}^{[1]}$ and $w_{ij}^{[2]}$ are the weights of between layer one to layer two and layer two to layer three accordingly.

2. Assign the activation level at the thresholding nodes.

$$X_0 = 1.0 \tag{6.5}$$

$$y_0^{[2]} = 1.0 \tag{6.6}$$

3. Choose an input-output pair. Let the input vector is $X = [X_1, \ldots, X_{N_1}]^T$ and the target output vector is $Y = [Y_1, \ldots, Y_{N_3}]^T$.

4. Propagate the input values from the nodes in the input layer to the nodes in the hidden layer using the activation function as follows:

$$y_j^{[2]} = \frac{1}{1 + e^{-\sum_{i=0}^{N_1} w_{ij}^{[1]} X_i}} \quad \forall j = 1, ..., N_2 \tag{6.7}$$

Note that i ranges from 0 to N_2. $w_{0j}^{[1]}$ is the weight for the hidden unit j (its propensity to fire irrespective of its inputs). X_0 is always 1.

5. Propagate the hidden values from the nodes in the hidden layer to the nodes in the output layer, using the activation following function:

$$y_j^{[3]} = \frac{1}{1 + e^{-\sum_{i=0}^{N_2} w_{ij}^{[2]} X_i}} \quad \forall j = 1, ..., N_3$$

Again, the weight $w_{0j}^{[2]}$ for output node j plays a role in the weighted summation. $y_0^{[2]}$ is always 1.

6. Compute the error of the nodes in the output layer, denoted $\delta_j^{[2]}$. This error is calculated based on the network's actual output ($y_j^{[3]}$) and the target output (Y_j).

$$\delta_j^{[2]} = y_j^{[3]}(1 - y_j^{[3]})(Y_j - y_j^{[3]}) \quad \forall j = 1, ...N_3 \tag{6.8}$$

7. Compute the errors of the nodes in the hidden layer, denoted $\delta_j^{[1]}$.

$$\delta_j^{[1]} = y_j^{[2]}(1 - y_j^{[2]}) \sum_{i=1}^{N_3} \delta_i^{[2]} w_{ji}^{[2]} \quad \forall j = 1, ..., N_2 \tag{6.9}$$

8. Adjust the weights between the hidden layer and output layer. The value of the learning rate, η, is chosen to be 0.35 empirically.

$$\triangle w_{ij}^{[2]} = \eta \cdot \delta_j^{[2]} \cdot y_i^{[2]} \quad \forall i = 0, ..., N_2, \ j = 1, ..., N_3 \tag{6.10}$$

9. Adjust the weights between the input layer and the hidden layer.

$$\triangle w_{ij}^{[1]} = \eta \cdot \delta_j^{[1]} \cdot X_i \quad \forall i = 0, ..., N_1, \ j = 1, ..., N_2 \tag{6.11}$$

10. Go to step 4 and repeat. When all input-output pairs have been presented to the network, one epoch is completed. Repeat steps 4 to 10 for as many epochs as desired.

The stated weight update rule is based upon the derivation of the activation function. The derivation of these weight update rules is discussed by Rumelhart *et al.* [31], and beyond the scope of the thesis.

One of the major obstacle to the wide use of connections learning networks in real-world applications is the slow convergent speed of the current training algorithms. At present, the widely used Back Propagation Training algorithm runs mostly faster than other earlier training algorithms. However the algorithm is still not fast enough as we expected.

In this work, several fast network training algorithms are tested and compared. These include the Cascade-Correlation Learning Architecture [33], the

Autoregressive Backpropagation Algorithm [31], and the Quick Learning Algorithm [9]. The latter is adopted in this work due to its simplicity and superior convergence speed. The Quick Learning Algorithm is modified from the regular back-propagation algorithm. A momentum term is added and the weight adjustment formula is given in Eq. (6.12) and Eq. (6.13), in the steps 9 and 10 of the stated training algorithm. These addition can smooth the weight change. The weights are updated by:

$$\Delta w_{ij}^{[2]}(t+1) = \eta \cdot \delta_j^{[2]} \cdot y_i^{[2]} + \alpha \Delta w_{ij}^{[2]}(t) \tag{6.12}$$

$$\Delta w_{ij}^{[1]}(t+1) = \eta \cdot \delta_j^{[1]} \cdot y_i^{[1]} + \alpha \Delta w_{ij}^{[1]}(t) \tag{6.13}$$

where, $y_i^{[2]}$, X_i, $\delta_j^{[1]}$, and $\delta_j^{[2]}$ are measured at time $t+1$. $\Delta w_{ij}^{[1]}(t)$ and $\Delta w_{ij}^{[1]}(t)$ are the changes of the weight experienced during the previous forward-backward iteration of the training. If α is set to be about 0.9, learning speed can be improved. In our experiments the best convergence results are obtained when the value of α is set to *zero* for the first several, then increased to 0.9 for the rest of the training. This process first gives the algorithm some time to find a good general search direction, and then accelerates the search in that direction [9].

The Quick Back Propagation algorithm converges significantly faster than the conventional Back Propagation algorithm. Given a fixed number of iteration, the Quick Back Propagation algorithm can converge to a smaller error. The error is calculated by:

$$error = \sum_{i=1}^{N_3} (Y_j - y_j^{[3]})^2 \tag{6.14}$$

The capability of error reduction of the two algorithms is illustrated in Table 6.1 through a test example. The "Exp. Value" means experimental total error values and the "Rel. Value" means relative error values in Table 6.1.

Table 6.1: The Comparison of Convergence Results

Number of Iterations		200	600	1000	2000	3000	5000
Back prop error reduction	Exp. Value	.42	.25	.10	.041	.023	.012
	Rel. Value	1	.59	.23	.098	.055	.029
Quick B P error reduction	Exp. Value	.028	.010	.0057	.0018	.0013	.00096
	Rel. Value	.19	.023	.014	.0043	.0031	.0023

The supervised training of a sixty input and six output, three layer network took seven hours on a Sun Sparc I Workstation, using the standard back propagation algorithm, while the same task took only 1.5 hours, using the Quick Back Propagation algorithm.

6.3 Traffic Flow Pattern Matching

In this work, dynamic traffic signal control is accomplished by recognizing traffic flow patterns from sensed traffic data, and applying pre-stored optimal traffic signal control parameters of the corresponding traffic flow pattern on-line to control traffic.

A traffic pattern can be described as a "characteristic group" of traffic flow conditions. The characteristic group, if described in a fuzzy traffic volume state space, can be represented by the position and shape of fuzzy data cluster in the form of a fuzzy cloud. Based upon the fuzzy clustering method, discussed in Chapter 3, the centers of the recognized traffic patterns (or the positions of the patterns) can be calculated.

The aim of fuzzy-neural network training is to obtain the weight matrices of the neural network which replace the "difficult-to obtain" multiple dimensional fuzzy membership functions. The weight matrices hold the fuzzy traffic pattern contours (or shapes), based upon the information obtained in fuzzy clustering.

The fuzzy-neural system is used to carry out the traffic flow pattern matching in real time.

The introduced method is tested using a real traffic system. The traffic system under testing is at a section of the Trans Canada Highway in the Duncan area, as described in Chapter 3. The traffic data used in this test are the average traffic volumes over the fall of 1992. A three-layer neural network is trained for traffic flow pattern matching. Traffic volume and associated fuzziness numbers used in the fuzzy clustering form a input-output pair for training the network. A set of sample input-output pairs is given as input parameters of the network, with the traffic volumes on the input side, and the fuzziness numbers calculated by fuzzy clustering on the output side. The number of input nodes of the network is determined by the number of maximum movements of the modeled traffic system. The number of output nodes is determined by the number of acceptable traffic patterns. The number of samples depends upon the number of acquisition time intervals of a day. If the acquisition time interval is 15 minutes, then 96 input-output data pairs are formed for a day.

The Duncan section on the Trans Canada Highway, has five intersections each with twelve movements. This results in sixty input nodes ($5 \times 12 = 60$). The traffic flow at this section is best represented as five traffic patterns, thus five output nodes are used in this test. The training data on the output side are the fuzzy output matrix obtained from the fuzzy clustering calculation. The matrix is 96×5 in dimension, and contains the fuzzy numbers associated with corresponding sample inputs, as illustrated in the Appendix A.

The number of hidden layer nodes is determined empirically. When the input nodes are fewer than thirty six, thirty hidden layer nodes are used. When more than thirty six input nodes are used, the number of hidden nodes are selected as half of the number of the input nodes. The approach works well for the test system. The structure of the network is illustrated in Figure 6.8.

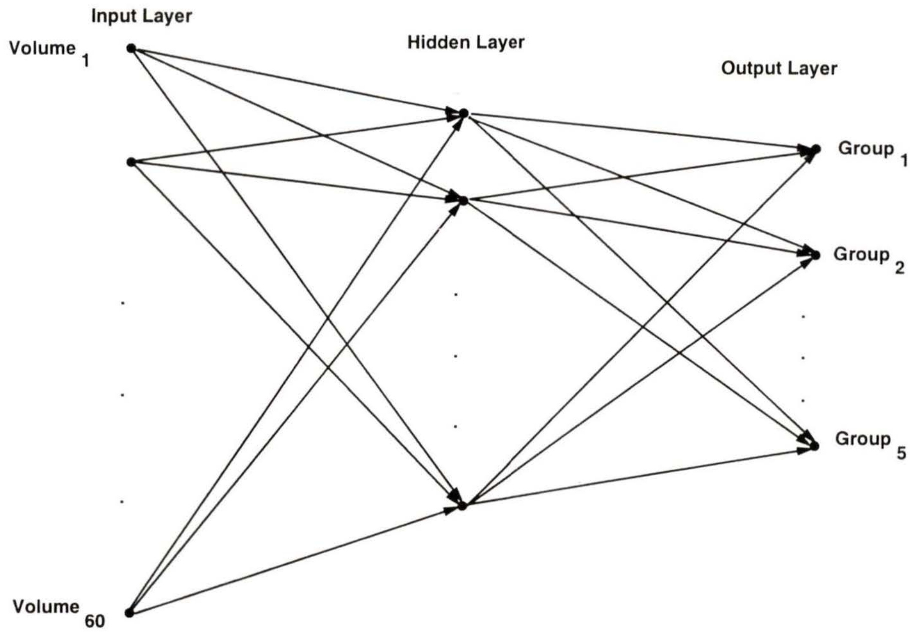


Figure 6.8: A Fuzzy-neural Network Used in Traffic Dynamic Control

The Quick Back-propagation neural network training is implemented in an off-line fuzzy-neural network training program, Traffic-Neural, on a 486-DX66 PC. It takes 7 hours for a new traffic system with random initial neural matrix, and 0.5 hour for a previous trained traffic system, to converge to acceptable criterion.

The neural matrix obtained from the training is then transmitted to the on-line part of the intelligent traffic signal control system to make traffic control decisions. The output node of the network, which has the largest fuzzy number, indicates the corresponding group is the best-fit traffic pattern.

6.4 Test Results

6.4.1 Test on Pattern Selection Consistency

The ability of the trained fuzzy-neural system to produce correct pattern matching results for the training data is tested with 100 percent correctness. The pattern matching time, using an on-line decision-making function, is less than 0.01 ms on a 486-DX66 PC. The test results are given in Appendix B, which is consistent to the fuzzy clustering output matrix given in Appendix A.

6.4.2 Test on Dynamic Traffic Flow Variation Response

The performance of the dynamic traffic signal control approach introduced in this work is compared to the static traffic signal control approach, following these steps:

- select a typical traffic flow data set.
- determine the fuzzy traffic patterns using the static approach, including the cluster number, cluster centers, and time period interval for each cluster using the fuzzy traffic pattern cluster program.
- design the timing plans of each traffic pattern, using the $TRANSYT-7F$ program for each period of time.
- increase the daily traffic volume by 30 percent.
- calculate the total delay of each sampling interval using the corresponding timing plans, given by the static control timing plans. The timing plans will not change according to the traffic volume variation in the static approach.
- determine the pattern number for each sampling using the dynamic traffic signal control approach with the fuzzy-neural system. The timing plans will change according to the traffic volume variation, due to the changed pattern numbers in the dynamic approach.
- calculate the total delay of each sampling using the timing plans that are dynamically selected based upon the calculated pattern number.

- compare the total delay caused by the two approaches.

Figure 6.9 presents a comparison of the different total delay for vehicles in the traffic system when traffic volume increases by 30 percent, using the dynamic and static traffic signal control methods. The traffic volume data are acquired from the Highway 99, Whalley section. In the figure, only those time intervals with different timing plan selection are included. Significant total delay reduction is obtained during between 12:30pm and 1:30pm. Modest reduction is accomplished during 9:00 am to 9:45 am, and 7:00 pm to 8:00 pm. This distribution of delay reduction is because traffic volume is near its peak from 12:00pm to 1:30pm, and the green time is not contained by the minimum green time. While in low volume period the green time is contained by minimum green time for the safety of pedestrians. Reducing peak hour total traffic delay is critical for traffic signal control.

6.4.3 Robustness and Stability of the Approach

To apply the proposed dynamic traffic signal control method to real traffic control, the robustness and stability of the dynamic traffic signal control system becomes important. When traffic condition information is incomplete, the fuzzy-neural network should still be able to select correct (or near correct) traffic patterns and timing plans. The introduced approach proves this task can be fulfilled through the test on incomplete traffic condition data.

Suppose two traffic detectors suddenly failed in the NBT movement of the Boys intersection on the Trans Canada Highway, and the sensed traffic volumes of the two affected channels turn to zero, the fuzzy-neural system is still able to select the same traffic patterns as before, although the fuzziness number will change slightly, as shown in Table 6.2. The system will lead to the same traffic pattern and timing plan associated with the maximum fuzzy output number

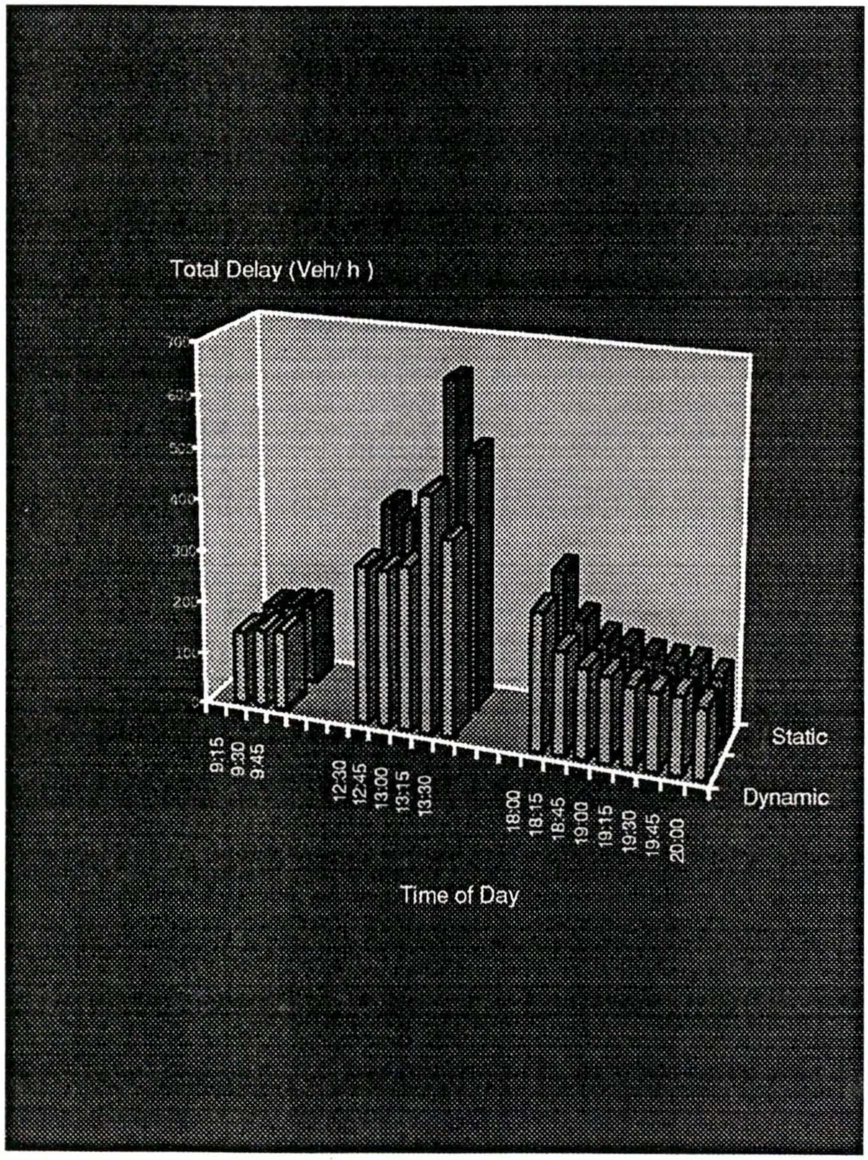


Figure 6.9: Total Traffic Delay for Dynamic and Static Traffic Signal Control

Table 6.2: Fuzziness Numbers as Output of Pattern Matching

Time	Volume	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
14:00	231	0.000	0.000	0.000	0.001	0.005	**0.992**
14:00	0	0.000	0.001	0.018	0.012	0.001	**0.973**

(boldface).

We have also tested an extreme case where 10 percent of the traffic flow sensors fail to function. The traffic pattern matching result only shifts from the "correct" pattern to its closest pattern with small changes to the optimal control parameters.

In addition, the proposed dynamic control system is bounded to one of the pre-stored timing plans. No matter what type of traffic flow inputs are provided, the system guarantees a set of usable control parameters.

Chapter 7

Conclusions

In this work, an approach for dynamic traffic signal control, based upon a fuzzy-neural intelligent system and traffic delay minimization, is introduced.

This research applies the general *quantitative intelligent system approach* [7] to the traffic control problem. The research is aimed at maximizing the efficiency of signalized traffic systems without new road construction, maintaining safe traffic flow through the traffic system, minimizing total traffic and individual vehicle delays, and reducing air and noise pollutions.

This approach consists of three major parts: (a) automated traffic flow pattern identification using fuzzy pattern clustering, (b) optimization of traffic control parameters (timing plan design) for identified traffic flow patterns, and (c) dynamic traffic signal control by real-time traffic flow monitoring, traffic flow pattern matching using the fuzzy-neural system, and execution of stored optimal control parameters.

The traffic flow pattern identification is carried out using fuzzy pattern clustering and matching techniques. A mathematical model is first introduced to quantify the fuzzy traffic conditions. The traffic condition at a moment is expressed as a hyperpoint in a m-dimensional traffic parameter space. Similar traffic conditions show as clouds of hyperpoints. The quantified traffic condition description allows the "characteristic groups" of traffic flow conditions to be recognized as traffic patterns, using the fuzzy clustering methods. These traffic patterns are closely studied. The optimal timing plans of these traffic patterns, which contain the optimal signal control parameters, are generated using commercial software through extensive optimization. The method has been adopted by the B.C. Ministry of Transportation and Highways in their present operations, and achieved overwhelming performance.

Dynamic traffic control is accomplished using the fuzzy traffic pattern clustering/matching methods and traffic plan optimization. The task is carried out in two steps: off-line learning and on-line control. The off-line learning part identifies all representative traffic patterns based upon previously collected traffic data, designs a timing plan for each identified traffic pattern using traffic delay minimization, and trains a fuzzy-neural system using the traffic pattern – optimal timing plan pairs generated. The on-line control part senses traffic flow in real-time, matches the sensed traffic flow condition with the best fitted traffic pattern, assigns the optimal timing plan of the matched traffic pattern to related traffic controllers dynamically. A method for short-term traffic flow condition prediction is also developed to offset the short delay in traffic flow condition sensing, and quasi-optimal traffic signal parameter updating at the controller. The approach makes dynamic traffic signal control of a corridor traffic system with quasi-optimal performance possible. The system is self-adaptive and capable of carrying out self-learning to varying traffic demands. Computer simulation and prototype testing using real traffic data have demonstrated significant traffic delay reduction. The B.C. Ministry of Transportation and Highways is in the process of implementing the introduced method.

The research directly contributes to static and dynamic traffic control research and practice. It also extends the research and applications of the *quantitative intelligent system approach*, and benefits the research on intelligent scheduling and planning for time and facility conflict activities. The research on developing a hybrid fuzzy-neural system combines the reasoning ability of a fuzzy system and the learning ability of a neural network, which is critical for a self-learning and self-adaptive, intelligent system.

Future Work

For further development of the intelligent, dynamic traffic control system, the following issues are to be addressed:

- The presently used *TRANSYT 7* timing plan generation program is a static traffic timing plan optimization software. It determines traffic control parameters based upon the macro-view average of traffic flow. The micro-view, cycle-based traffic flow cannot be handled by the method. A more adequate timing plan optimization method is needed to the *TRANSYT 7* program.

- The dynamic traffic control system analyzes traffic condition, using only one multiple-input-output rule. For complex traffic flow states, a multiple level, fuzzy-neural reasoning system is needed.

- The off-line learning and on-line control parts of the system now reside on two separated computers. The communication between these two computers is troublesome due to the poor service of public telephone lines connecting the two computers. Since the on-line traffic data transmission and decision making only takes 3-4 seconds during the 5-15 minute control time interval, the off-line part and on-line part can be merged together on a single computer with a multi-task operation system, thus reducing the influence of communication errors and resource cost, and increasing system reliability.

- Limited by the timing plan optimization algorithm, the present intelligent dynamic traffic control system can only be used for an artery traffic control system. The system can be extended to control a network traffic system if an effective network traffic optimization algorithm is made available.

Bibliography

1. Bezdekk, J. C.: 1973, "Cluster Validity with fuzzy Sets," *Journal Cybern.*, **3**, pp. 58-73.

2. Bezdekk, J. C.: 1975, "Mathematical Models for systematics and Taxonomy," *Proceeding 8th International Conference on Numerical Taxonomy*, G. Estabrook, (Ed.), San Francisco, CA: Freeman, pp. 143-166.

3. Bezdekk, J. C.: 1981, *Pattern Recognition with Fuzzy objective Function Algorithm*, Plenum, NY, USA.

4. Bedworth, D.D., and J. E. Baibey: 1987, *Integrated Production Control Systems*, 2E, John Wiley and Sons.

5. Chen, K. and R. French: 1991, "Organizational Response to Intelligent Vehicle -Highway Systems," *SAE Technique Paper Series 1991 – Proceedings of the Future Transportation Technology Conference and Exposition*, Portland, OR, USA, Aug. 5-7, pp. 1-11.

6. Davis, G. J. Hang: 1993, "Filtering and Prediction on Freeways Using Markov Models," *Proceedings of the 1993 Pacific Rim Transportation Technique Conference*, Seattle, ASCE, NY, USA, pp. 43-49.

7. Dong, Z.: 1994, *Artificial Intelligence in Optimal Design and Manufacturing*, PTR Prentice Hall, Englewood Cliffs, NJ, USA.

8. Manzur Elahi, S., A. Essam Radwan, and K. Michael Goul: 1990, "Knowledge-based System for Adaptive Traffic Signal Control," *Trans-*

portation Research, Record 1324, pp. 115-122.

9. Fahlman S. E.: 1988, "Faster Learning Variations on Back Propagation: An Empirical Study," *Proceeding of the 1988 Connectionist Models Summer School*, Morgan Kaufmann.

10. French, R., L. Schaeffer, and Norman, M.: 1991, "Global Challenge of Intelligent Vehicle/Highway Systems (IVHS) Standards," *Proceeding of the 6th International Pacific Conference on Automotive Engineering*, Seoul, South Korea, Oct. 28 - Nov. 1, pp. 1435-1443.

11. Groover, M. P. and E. W. Zimmers, Jr.: 1984, *CAD/CAM: Computer-Aided Design and Manufacturing*, Prentice Hall, Englewood Cliffs, NJ, USA.

12. Hunt, P. B., D. I. Robertson, R. D. Bretherton and R. I. Winton: 1989, "SCOOT — A Traffic Responsive Method of Coordinating Signals" *Transport and Road Research Laboratory (TRRL)*, Report 1014, August.

13. Keller, J. M., D. J. Hunt: 1985, "Incorporating Fuzzy Membership Functions into the Perceptron Algorithm," *IEEE Trans. Anal. Machine Intelligent*, Vol. PAMI, **6**, pp. 693-699, November.

14. Jeach, J.L.: 1985, *Statistical Analysis of Measurement Errors*, John Wiley and Sons.

15. T. Kohonen: 1984, *Self-organization and Associative Memory*, Springer-Verlag, Berlin, Germany.

16. Lippmann, R. P. : 1987, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, pp. 4-22, April.

17. Lowrie, P. R.: 1990, "SCATS Sydney Co-ordinated Adaptive Traffic System: A Traffic-Responsive Method for Controlling Urban Traffic," *Road*

and Traffic Authority of New South Wales, Sydney, Australia.

18. McShane, W. R. and R. P. Rooess: 1990, *Traffic Engineering*, Prentice Hall, Englewood Cliffs, NJ, USA.

19. Michalopoulos, P.G.: 1991, "Vehicle Detection through Video Image Processing: The Autoscope System," *IEEE Transactions on Vehiclar Technology*, **40**, 1, pp. 21-29.

20. Miller, A. J.: 1963, "Settings for Fixed-Cycle Traffic Signals," *Operations Research Quarterly*, **14**.

21. Miller, A. J.: 1968, "The Capacity of Signalized Intersections in Australia," *Australia Road Research Board*, ARRB Bulletin 3.

22. Minisky, M. and S. Papert: 1969, *Perceptrons: An Introduction to Computational Geometry*, MIT Press, MA, USA.

23. Nagui M. R: 1990, "A Cycle-Based Model for Platoon Delays at Signalized Intersections," *Chicago III: Urban Transportation Center*, June.

24. Nagui, M. R.: 1991, "Cycle-by-Cycle Analysis of Congested Flow at Signalized Intersections," *ITE Journal*, March, pp. 33-36.

25. Nathan, H. C.: 1991, "Impacts of Smart Signals on Intersection Capacity and Level of Service," *Highway Capacity and Level of Service*, Brannolte (Ed.) Balkema, Rotterdam.

26. Newell, G. F.: 1965, "Approximation Methods for Queues with Application to the Fixed-Cycle Traffic Light," *SIAM Review*, **7**.

27. Michalopoulos, P. G., R. D. Jacobson, A. A. Craig and C. B. James: 1992, "Field Deployment of Autoscope in the FAST-TRAC ATMSATIS Program," *Journal of Traffic Engineering and Control*, September.

28. Pomerleau, D.: 1989, "ALVINN: An Autonomous Land Vehicle in A Neural Network," *Advances in Neural Information Processing System I*, D. Touretzky, (Ed.), San Mateo, Morgan Kaufmann, CA, USA, pp. 305-313.

29. Reilly, W. R. and C. C. Gardner: 1977, "Technique for Measuring Delay at Intersections," *TRRL 644*.

30. Rosenblatt, R.: 1959, *Principle Neurodynamics*, Spartan Books, NY, USA.

31. Rumelhert, D., J. McClelland, *et al.*: 1986, *Parallel Distributed Processing*. Cambridge, MIT Press, MA, USA.

32. Rumelhart, D. E., G. E. Hinton and R. J. Williams: 1986, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, D.E. Rumelhart and J.L. McClelland (Eds), **1**: Foundations. MIT Press, MA, USA.

33. Sejnowski, T. J. and C. R. Rosenberg.: 1987, "Parallel Networks That Learn to Pronounce English Text," *Comlex Systems*, **1**, 1, pp. 145-168.

34. Shin, Y.C., A. H. Abodelmonem, and S. Kumara: 1992, "Neural Networks in Manufacturing and Robotics," *Proceeding of the Winter Annual Meeting of The American Society of Mechanical Engineers*, Anaheim, CA, USA. Nov., pp. 8-13.

35. Swedish National Road Administration: 1991, "LHOVRA: A Traffic Signal Control Strategy for Isolated Junctions," *SNRA Publication*, 52E, Borlange.

36. Takagi, H., I. Hayashi: 1991, "NN – Driven Fuzzy Reasoning," *International Journal Approximate Reasoning*, **5**, 3, pp. 192-212.

37. Tesauro, G. and T. J. Sejnowski: 1989, "A Parallel Network That Learns to Play Backgammon," *Artificial Intelligence*, **39**, 3.

38. Webster, F. V.: 1958, "Traffic Signal Settings," *Road Research Technical Paper No. 39*, Road Research Laboratory, Her Majesty's Stationary Office.

39. Winndham, M. P.: 1981, "Cluster Validity for Fuzzy Clustering Algorithms," *Fuzzy Sets System*, **5**, pp. 177-185.

40. Vincent, R. A. and C. P. Young: 1986, "Self-Optimising Traffic Signal Control Using Microprocessors – The TRRL MOVA Strategy for Isolated Intersections," *Traffic Engineering and Control*, **27** (7/8), July/Augest, pp. 385-387.

41. Vincent, R. A. and J. R. Peirce: 1991, "MOVA: Traffic Responsive, Self-Optimising Signal Control for Isolated Intersections," *TRRL Research Report 170.*

42. Xue, D., and Z. Dong: 1994, "Developing A Quantitative Intelligent System for Implementing Concurrent Engineering Design," *Journal of Intelligent Manufacturing*, **5**, 1994, pp. 107-116.

43. Zadeh, L.A.: 1965, "Fuzzy Sets," *Information and Control*, **8**, pp. 338-352.

Appendix A

Fuzzy Clustering Output

Time	$Cluster_1$	$Cluster_2$	$Cluster_3$	$Cluster_4$	$Cluster_5$
0000	0.997	0.002	0	0	0
0015	1	0	0	0	0
0030	1	0	0	0	0
0045	1	0	0	0	0
0100	1	0	0	0	0
0115	1	0	0	0	0
0130	0.999	0.001	0	0	0
0145	0.999	0.001	0	0	0
0200	0.999	0.001	0	0	0
0215	0.999	0.001	0	0	0
0230	0.998	0.001	0	0	0
0245	0.997	0.002	0	0	0
0300	0.996	0.003	0.001	0	0
0315	0.996	0.003	0.001	0	0
0330	0.996	0.003	0.001	0	0
0345	0.996	0.003	0.001	0	0
0400	0.996	0.003	0.001	0	0
0415	0.996	0.003	0.001	0	0
0430	0.997	0.002	0	0	0
0445	0.999	0.001	0	0	0
0500	0.999	0	0	0	0
0515	1	0	0	0	0
0530	0.993	0.006	0.001	0	0
0545	0.949	0.044	0.004	0.002	0.001
0600	0.813	0.167	0.013	0.005	0.002
0615	0.483	0.479	0.026	0.009	0.004
0630	0.094	0.873	0.023	0.007	0.003
0645	0.016	0.966	0.014	0.003	0.001
0700	0.016	0.954	0.024	0.005	0.002
0715	0.023	0.806	0.145	0.019	0.007
0730	0.009	0.128	0.812	0.039	0.011
0745	0.004	0.025	0.83	0.116	0.024
0800	0.002	0.015	0.965	0.015	0.003
0815	0.004	0.026	0.9	0.056	0.013
0830	0.001	0.01	0.976	0.01	0.002
0845	0.002	0.014	0.907	0.066	0.012
0900	0	0.002	0.993	0.004	0.001
0915	0	0.002	0.99	0.007	0.001
0930	0.001	0.008	0.923	0.061	0.007
0945	0.002	0.011	0.731	0.238	0.018

Time	$Cluster_1$	$Cluster_2$	$Cluster_3$	$Cluster_4$	$Cluster_5$
1000	0.002	0.013	0.549	0.412	0.024
1015	0.002	0.011	0.337	0.62	0.03
1030	0	0.001	0.022	0.965	0.011
1045	0.001	0.003	0.052	0.921	0.024
1100	0	0.001	0.019	0.969	0.011
1115	0	0.001	0.009	0.983	0.007
1130	0	0	0.004	0.986	0.01
1145	0	0.001	0.011	0.97	0.018
1200	0	0.001	0.012	0.968	0.018
1215	0	0.001	0.008	0.979	0.012
1230	0	0.001	0.014	0.974	0.011
1245	0	0.001	0.006	0.987	0.006
1300	0	0	0.004	0.986	0.009
1315	0	0	0.004	0.982	0.013
1330	0	0.001	0.006	0.975	0.019
1345	0	0.001	0.005	0.97	0.024
1400	0	0.002	0.015	0.835	0.148
1415	0	0.002	0.011	0.891	0.096
1430	0	0.002	0.014	0.864	0.119
1445	0.001	0.003	0.017	0.537	0.442
1500	0.001	0.003	0.015	0.526	0.456
1515	0	0.002	0.011	0.258	0.728
1530	0	0.001	0.003	0.039	0.958
1545	0	0	0.002	0.026	0.971
1600	0	0.001	0.002	0.017	0.98
1615	0	0	0.002	0.013	0.985
1630	0	0.001	0.005	0.03	0.963
1645	0	0.001	0.003	0.021	0.974
1700	0	0	0.001	0.008	0.99
1715	0	0.001	0.004	0.059	0.936
1730	0.001	0.004	0.023	0.573	0.398
1745	0.001	0.003	0.031	0.927	0.038
1800	0.002	0.013	0.345	0.608	0.032
1815	0.002	0.017	0.823	0.143	0.015
1830	0.001	0.011	0.958	0.026	0.004
1845	0.001	0.008	0.965	0.023	0.003
1900	0.003	0.044	0.916	0.031	0.006
1915	0.01	0.21	0.71	0.056	0.014
1930	0.015	0.515	0.412	0.046	0.013
1945	0.017	0.625	0.304	0.042	0.012

Time	$Cluster_1$	$Cluster_2$	$Cluster_3$	$Cluster_4$	$Cluster_5$
2000	0.012	0.849	0.114	0.018	0.006
2015	0.005	0.968	0.022	0.004	0.002
2030	0.004	0.979	0.013	0.003	0.001
2045	0.005	0.971	0.019	0.004	0.001
2100	0.004	0.976	0.016	0.003	0.001
2115	0.001	0.997	0.002	0	0
2130	0.001	0.996	0.002	0.001	0
2145	0.001	0.998	0.001	0	0
2200	0.004	0.992	0.003	0.001	0
2215	0.033	0.953	0.01	0.003	0.001
2230	0.207	0.759	0.023	0.007	0.004
2245	0.48	0.484	0.023	0.008	0.004
2300	0.628	0.342	0.019	0.007	0.003
2315	0.817	0.165	0.012	0.004	0.002
2330	0.911	0.078	0.007	0.003	0.001
2345	0.911	0.078	0.007	0.003	0.001

Appendix B

Trained Fuzzy-neural Network Output

Time	$Cluster_1$	$Cluster_2$	$Cluster_3$	$Cluster_4$	$Cluster_5$
0000	0.9896	0.0079	0.0032	0.0009	0.0004
0015	0.9924	0.0050	0.0038	0.0009	0.0003
0030	0.9948	0.0046	0.0025	0.0011	0.0003
0045	0.9959	0.0044	0.0025	0.0009	0.0003
0100	0.9970	0.0033	0.0022	0.0010	0.0002
0115	0.9907	0.0027	0.0026	0.0009	0.0002
0130	0.9976	0.0024	0.0023	0.0009	0.0002
0145	0.9974	0.0019	0.0032	0.0009	0.0002
0200	0.9976	0.0025	0.0022	0.0010	0.0002
0215	0.9979	0.0023	0.0019	0.0012	0.0002
0230	0.9979	0.0027	0.0020	0.0012	0.0002
0245	0.9981	0.0020	0.0022	0.0011	0.0002
0300	0.9981	0.0020	0.0023	0.0010	0.0002
0315	0.9983	0.0018	0.0023	0.0011	0.0002
0330	0.9983	0.0018	0.0019	0.0012	0.0002
0345	0.9983	0.0019	0.0019	0.0011	0.0002
0400	0.9983	0.0020	0.0019	0.0011	0.0002
0415	0.9983	0.0023	0.0017	0.0011	0.0002
0430	0.9981	0.0025	0.0019	0.0011	0.0002
0445	0.9978	0.0027	0.0021	0.0010	0.0002
0500	0.9972	0.0034	0.0020	0.0009	0.0002
0515	0.9951	0.0044	0.0022	0.0009	0.0004
0530	0.9869	0.0134	0.0018	0.0009	0.0006
0545	0.9512	0.0424	0.0027	0.0009	0.0008
0600	0.8249	0.1495	0.0031	0.0010	0.0011
0615	0.4763	0.4934	0.0036	0.0014	0.0015
0630	0.0776	0.9091	0.0034	0.0023	0.0030
0645	0.0275	0.9422	0.0079	0.0031	0.0043
0700	0.0266	0.9278	0.0248	0.0025	0.0039
0715	0.0174	0.7934	0.1225	0.0018	0.0079
0730	0.0063	0.1274	0.8576	0.0019	0.0189
0745	0.0024	0.0314	0.8224	0.0932	0.0137
0800	0.0060	0.0149	0.9573	0.0204	0.0053
0815	0.0057	0.0172	0.9191	0.0372	0.0076
0830	0.0047	0.0177	0.9716	0.0070	0.0103
0845	0.0028	0.0053	0.9459	0.0538	0.0108
0900	0.0029	0.0155	0.9497	0.0218	0.0069
0915	0.0023	0.0116	0.9629	0.0266	0.0092
0930	0.0012	0.0101	0.9019	0.0937	0.0101
0945	0.0010	0.0089	0.7208	0.2965	0.0093

Time	$Cluster_1$	$Cluster_2$	$Cluster_3$	$Cluster_4$	$Cluster_5$
1000	0.0008	0.0146	0.5612	0.4132	0.0079
1015	0.0007	0.0054	0.3269	0.6209	0.0109
1030	0.0006	0.0022	0.0733	0.8864	0.0182
1045	0.0006	0.0033	0.0570	0.9335	0.0117
1100	0.0005	0.0025	0.0304	0.9572	0.0110
1115	0.0003	0.0047	0.0283	0.9398	0.0169
1130	0.0003	0.0013	0.0051	0.9664	0.0320
1145	0.0006	0.0005	0.0136	0.9710	0.0219
1200	0.0004	0.0010	0.0078	0.9697	0.0188
1215	0.0003	0.0029	0.0036	0.9821	0.0156
1230	0.0003	0.0055	0.0088	0.9914	0.0045
1245	0.0004	0.0009	0.0103	0.9907	0.0076
1300	0.0004	0.0006	0.0068	0.9647	0.0290
1315	0.0003	0.0007	0.0088	0.9604	0.0337
1330	0.0004	0.0005	0.0095	0.9635	0.0354
1345	0.0003	0.0008	0.0102	0.9427	0.0474
1400	0.0002	0.0013	0.0041	0.8582	0.1436
1415	0.0003	0.0011	0.0022	0.9338	0.1019
1430	0.0002	0.0008	0.0079	0.8278	0.1308
1445	0.0001	0.0009	0.0067	0.5312	0.4528
1500	0.0002	0.0008	0.0038	0.5371	0.4374
1515	0.0002	0.0003	0.0045	0.2847	0.7252
1530	0.0001	0.0003	0.0070	0.0678	0.9106
1545	0.0002	0.0003	0.0040	0.0526	0.9409
1600	0.0001	0.0003	0.0037	0.0084	0.9868
1615	0.0001	0.0003	0.0022	0.0060	0.9933
1630	0.0001	0.0001	0.0047	0.0011	0.9974
1645	0.0001	0.0002	0.0042	0.0024	0.9957
1700	0.0001	0.0006	0.0033	0.0122	0.9803
1715	0.0002	0.0004	0.0037	0.0930	0.9104
1730	0.0001	0.0026	0.0050	0.5628	0.4155
1745	0.0003	0.0021	0.0354	0.8832	0.0380
1800	0.0009	0.0035	0.3475	0.5938	0.0150
1815	0.0011	0.0179	0.8079	0.1619	0.0112
1830	0.0018	0.0118	0.9371	0.0569	0.0077
1845	0.0022	0.0077	0.9486	0.0604	0.0082
1900	0.0029	0.0662	0.9020	0.0150	0.0076
1915	0.0029	0.2001	0.7123	0.0243	0.0040
1930	0.0033	0.5256	0.4205	0.0157	0.0038
1945	0.0036	0.6353	0.2980	0.0194	0.0031

Time	$Cluster_1$	$Cluster_2$	$Cluster_3$	$Cluster_4$	$Cluster_5$
2000	0.0046	0.8640	0.1011	0.0147	0.0026
2015	0.0039	0.9471	0.0489	0.0139	0.0024
2030	0.0042	0.9656	0.0362	0.0133	0.0022
2045	0.0050	0.9639	0.0286	0.0132	0.0024
2100	0.0038	0.9667	0.0358	0.0118	0.0026
2115	0.0050	0.9840	0.0211	0.0077	0.0028
2130	0.0059	0.9894	0.0090	0.0086	0.0025
2145	0.0091	0.9842	0.0103	0.0080	0.0019
2200	0.0139	0.9815	0.0080	0.0064	0.0017
2215	0.0389	0.9719	0.0034	0.0067	0.0012
2230	0.2054	0.7638	0.0078	0.0029	0.0009
2245	0.5009	0.4795	0.0075	0.0019	0.0007
2300	0.6196	0.3492	0.0056	0.0014	0.0009
2315	0.8064	0.1531	0.0057	0.0013	0.0007
2330	0.9174	0.0677	0.0048	0.0011	0.0006
2345	0.9174	0.0677	0.0048	0.0011	0.0006

Vita

Surname: Wu

Given Names: Jian

Place of Birth: Beijing, China

Date of Birth: December 11, 1955

Educational Institutions Attended:

University of Victoria	1991 to 1995
China Geoscience University	1985 to 1987
Chengdu Geology College	1978 to 1981

Degrees Awarded:

B.A.	Chengdu Geology College	1981
M. A. Sc.	China Geoscience University	1987

Honours and Awards:

Publications:

Partial Copyright License

Title of Thesis/Dissertation:

Dynamic Traffic Signal Control Using A Self-learning, Fuzzy-neural Intelligent System

Author: JIAN WU

[redacted]

signature

Apr. 21, 1995

Date