

UC Berkeley

Earlier Faculty Research

Title

CARTESIUS: A Cooperative Approach to Real-Time Decision Support for Multi-Jurisdictional Traffic Congestion Management

Permalink

<https://escholarship.org/uc/item/60p449bh>

Author

Logi, Filippo

Publication Date

1999

UNIVERSITY OF CALIFORNIA,
IRVINE

**CARTESIUS: A Cooperative Approach
to Real-Time Decision Support
for Multi-Jurisdictional Traffic Congestion Management**

DISSERTATION

submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in Civil Engineering

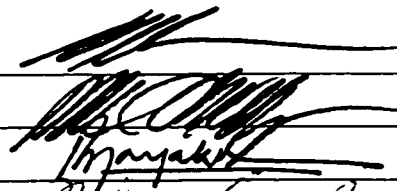
by

Filippo Logi

Dissertation Committee:
Professor Stephen G. Ritchie, Chair
Professor R. Jayakrishnan
Professor Michael G. McNally
Professor Wilfred W. Recker

1999

The dissertation of Filippo Logi is approved
and is acceptable in quality and form
for publication on microfilm:



Stephen G. Kirtley
Committee Chair

University of California, Irvine

1999

TABLE OF CONTENTS

LIST OF FIGURES	vi
LIST OF TABLES	vii
ACKNOWLEDGEMENTS	viii
CURRICULUM VITAE	ix
ABSTRACT OF THE DISSERTATION	xiii
1 Introduction	1
1.1 The Need for Integrated Management of Congested Facilities	1
1.2 Real-Time Incident Management	3
1.3 Distributed Problem Solving	4
1.4 Research Objectives	6
1.5 Research Contribution	6
1.6 Outline of the Dissertation	8
2 Integrated Traffic Management	9
2.1 Integrated Advanced Transportation Management and Information Systems	11
2.2 The Santa Monica Smart Corridor	13
2.2.1 The System Architecture	14
2.2.2 Incident Management	15
2.2.3 Critique	18
2.3 The City of Irvine Field Operational Test	19
2.3.1 Ramp Metering	20
2.3.2 Adaptive Signal Control	21
2.3.3 Data Management and Communication	22
2.3.4 Critique	23
2.4 Integrated Corridor Control in Glasgow	23
2.4.1 Signal Control	24
2.4.2 Traffic Diversion	27
2.4.3 Simulation Results	28
2.4.4 Critique	29
2.5 Coordinated Traffic Management in Munich	30

2.5.1	System Architecture	30
2.5.2	Critique	32
3	The Centralized Basic Component	34
3.1	Traffic Congestion Manager	35
3.1.1	The Knowledge Organization	37
3.1.2	Problem Characterization	38
3.1.3	Problem Response	43
3.1.4	Problem Monitor	50
3.1.5	The Test Site	50
3.2	The Evaluation Process	52
3.2.1	The Simulated Environment	55
3.2.2	Evaluation Results	58
4	Distributed Problem Solving and Functionally Accurate Cooperation	63
4.1	Distributed Problem Solving	63
4.2	Functionally Accurate, Cooperative Distributed Systems	65
4.2.1	Early Experimental Results: Speech Recognition	67
4.2.2	Explicit Goal Structures: Vehicle Monitoring	68
4.2.3	Multistage Negotiation: Distributed Constraint Satisfaction	73
4.2.4	More Sophistication: Aircraft Monitoring	75
4.2.5	Solution Quality	77
4.2.6	Nearly Monotonic Domains	84
5	The Distributed Architecture	87
5.1	The Architecture	87
5.1.1	Input/Output	88
5.1.2	Data Sharing and Agent Organization	90
5.1.3	The Rationales for a Functionally Accurate, Cooperative Approach	93
5.2	Problem Analysis	96
5.3	Problem Response	102
5.3.1	The Problem Response Algorithm	103
5.3.2	Strategies	112
5.3.3	Conditions	117
5.3.4	Termination Criteria	121
5.3.5	Near-Monotonicity	122
5.3.6	Computational Complexity	124
5.4	Current Problem Monitor	127
6	Control Algorithms: Software Design and Development	131
6.1	Signal Control	131
6.1.1	Signal Control on Diversion Routes	132
6.1.2	Signal Control at Oversaturated Intersections	137
6.1.3	Estimation of Average Green Time and Delay	141
6.2	Ramp Metering	143

6.2.1	Ramp Metering on Diversion Routes and at Congested On-Ramps .	144
6.2.2	Ramp Metering for Congested Freeways	145
6.2.3	Delay Computation	145
6.3	Traffic Diversion	147
6.4	Demand Computation	151
6.5	Treatment of Uncertainty	156
7	Performance Evaluation	161
7.1	The Test Site	162
7.1.1	Real-Time Connection with Traffic Management Centers	164
7.2	The Simulated Environment	165
7.3	Validation Results	167
7.3.1	Network Performance	167
7.3.2	Response Time	174
7.3.3	Effect of Signal Plans	177
8	Conclusions and Directions for Future Research	180
8.1	Conclusions	180
8.2	Future Research	182
8.2.1	Refinements to Problem Solving	182
8.2.2	Extensions	183
8.2.3	Application of Cooperative Approach to Other Domains	184
	BIBLIOGRAPHY	186

LIST OF FIGURES

2.1	The Santa Monica Freeway Smart Corridor	14
2.2	An intersection and a link in the graph (Z, J)	25
3.1	A schematic representation of input and output configuration	36
3.2	The structured collection of knowledge units	37
3.3	A subset of the frame system for problem diagnosis	41
3.4	The class hierarchy of problems	43
3.5	The search tree in the problem-solving process	49
3.6	The test site in Anaheim, California	51
4.1	A distributed, two-agent aircraft-monitoring scenario	67
4.2	A centralized goal tree	71
4.3	A distributed goal tree	72
4.4	Multistage negotiation: example of a communication network	74
4.5	Examples of global consistency sources of uncertainty for vehicle monitoring	78
4.6	The application of different global resolution strategies	81
5.1	The distributed architecture	88
5.2	Spillback at off-ramps	98
5.3	Spillback at signalized intersections	98
5.4	The capacity of a secondary congestion	101
5.5	The search trees in the distributed problem-solving process	108
5.6	The strategy class hierarchy	114
5.7	An example of redundant strategies	116
5.8	The local search tree	123
6.1	An eight-phase dual ring	135
6.2	The movement numbers and the critical movement search diagram	136
6.3	Demand and service rates	138
6.4	Ramp metering variables	144
6.5	The backbone and Δ -demand at an intersection	153
6.6	The currently expected demand and measured volumes at an intersection	155
7.1	The test site in Irvine, California	163

LIST OF TABLES

3.1	Basic module: simulated network performance	58
3.2	Basic module: response time	59
6.1	Measured percentage diversion	149
6.2	Modeled percentage diversion according to cause and delay	149
6.3	The backbone, Δ -demand, and expected current demand (veh/h)	153
7.1	Distributed system: incident characteristics in the test scenarios	168
7.2	Distributed system: simulated average and total travel time	169
7.3	Distributed system: simulated average and total traveled distance	170
7.4	Average travel time for the 405 NB scenarios	171
7.5	Average travel time for the 405 SB scenarios	172
7.6	Average travel time for the 5 NB scenarios	172
7.7	Average travel time for the 5 SB scenarios	172
7.8	Average travel time for Alton (WB and EB) and on-ramp scenarios	173
7.9	Average travel time for multiple incident scenarios	173
7.10	Distributed system: maximum response time	176
7.11	Travel time with and without signal control for an incident on 405 NB	177
7.12	Travel time with and without signal control for an incident on 405 SB	178
7.13	Travel time with and without signal control for an incident on 5 NB	178

ACKNOWLEDGEMENT

I wish to thank my professors at the Institute of Transportation Studies in Irvine, for their tireless support, guidance and friendship during these years. Dr. Stephen Ritchie, my advisor, for always asking the right questions. Dr. Michael McNally for being there whenever I had a question. Dr. Jayakrishnan for the animated and fruitful discussions. Dr. Will Recker for always being supportive and understanding. Dr. Amelia Regan for giving me the example.

I wish to thank all my friends at ITS, students, researchers, and staff. Craig in particular, James, Ziggy, Anne Marie, Kathy, Lyn and Tom, for bearing with me.

I dedicate this dissertation to my family, whose love and guidance always supported me.

CURRICULUM VITAE

Filippo Logi

Education

- 1991** Laurea (Bachelor of Science) Degree in Computer Science. University of Pisa, Italy.
Faculty of Mathematical, Physical and Natural Sciences
Thesis: Parallel Algorithms for String Matching with Variable Length Don't Cares
Advisor: Prof. Alan A. Bertossi
- 1995** Master of Science in Civil Engineering, with concentration in Transportation. University of California, Irvine. School of Engineering Thesis: A Software Architecture for the Integration of Advanced Transportation Management and Information Systems
Advisor: Prof. Stephen G. Ritchie
- 1999** Ph.D. in Civil Engineering, with concentration in Transportation. University of California, Irvine. School of Engineering
Dissertation: An Intelligent, Cooperative Approach to Distributed, Network-Wide Traffic Control Management
Advisor: Prof. Stephen G. Ritchie

Work Experience

- July 1991 - December 1991* Student Researcher at the Department of Computer Science of the University of Pisa, Italy
Research in the field of parallel pattern matching
- November 1991 - April 1992* Teacher at the ITIC Technical High School, Livorno, Italy.
Substitute teacher of Microprocessor Organization and I/O Interface.
- May 1992 - July 1993* Visiting Researcher at the Forschungszentrum Informatik, in Karlsruhe, Germany.
Blackboard-oriented architecture for traffic management in urban networks

Graphical user interface development tool

August 1993 - to date Graduate Student Researcher at the Institute of Transportation Studies. University of California. Irvine

Real-time decision support for surface street traffic incident management

Real-time knowledge-based architecture for incident management

Real-time network-wide distributed traffic control management

Awards

J. William Fulbright Scholarship. 1993-1994

University of California Transportation Center (UCTC) Ph.D. Dissertation Fellowship, 1996-1997

European Community/University of Pisa Fellowship at the Forschungszentrum Informatik. Karlsruhe. Germany, 1992-1993

University of California Regents' Fellowship, Fall 1995

University of California Regents' Fellowship, Winter 1996

University of California Regents' Fellowship, Spring 1996

Publications

Papers in Refereed Journals

J. Hesse, R. Koenig, F. Logi, and J. Herder, "A Prototype for an Interface Builder for the Common Lisp Interface Manager, CLIB". ACM Sigplan Notices, 28 (8), 1993.

A. A. Bertossi and F. Logi, "Parallel String Matching with Variable Length Don't Cares". Journal of Parallel and Distributed Computing, 22.(2), 1994.

F. Logi and S. G. Ritchie, "Development and Validation of a Knowledge-Based System for Traffic Congestion Management and Control", submitted for publication to Transportation Research, Part C, 1999.

Papers at Refereed Conferences

F. Logi and D. Wild, "An Intelligent Predictor for Urban Traffic", Second Meeting of the EURO Working Group on Urban Traffic Transportation. Paris, France, 1993.

F. Logi, B. Wild, G. Ambrosino, M. Bielli, and M. Boero, "Artificial Intelligence in Traffic Engineering: a Survey of Current Applications", First Conference on Applications of Artificial Intelligence Techniques in Engineering. Napoli, Italy, 1994.

M. Molina, F. Logi, S. G. Ritchie, and J. Cuenca, "An Architecture Integrating Symbolic and Connectionist Models for Traffic Management Center Decision Support", Fourth International Conference on Applications of Advanced Technologies in Transportation Engineering. Capri, Italy, 1995.

S. G. Ritchie, F. Logi, S. Kang, and C. Rindt, "Evaluation of a Real-Time Expert System for Surface Street Traffic Management and Control", Fourth International Conference on Applications of Advanced Technologies in Transportation Engineering. Capri, Italy, 1995.

F. Logi and S. G. Ritchie, "A Knowledge-Based System for Integrated Freeway and Arterial Incident Management and Control. Implementation and Validation Results", Eighth IFAC Symposium on Transportation Systems. Chania, Greece, 1997.

F. Logi and S. G. Ritchie, "A Distributed Approach to Network-Wide Traffic Control Management", Fifth ASCE International Conference on Application of Advanced Technologies in Transportation Engineering. Newport Beach, CA, 1998.

Other Publications and Presentations

F. Logi and S. G. Ritchie, "A Cooperative Approach to Distributed Network-Wide Traffic Congestion Management", presentation at the INFORMS Meeting. San Diego, CA, 1997.

F. Logi and S. G. Ritchie, "Verification, Validation and Evaluation of TCM, a Knowledge-Based System for Traffic Congestion Management", ITS Research Report UCI-ITS-RR-97-1, 1997.

F. Logi and S. G. Ritchie. "A Distributed Approach to Network-Wide Incident Management and Traffic Control", presentation at the Fourth University of California Transportation Center Graduate Student Conference. Berkeley, CA. 1997.

ABSTRACT OF THE DISSERTATION

CARTESIUS: A Cooperative Approach
to Real-Time Decision Support
for Multi-Jurisdictional Traffic Congestion Management

By

Filippo Logi

Doctor of Philosophy in Civil Engineering

University of California, Irvine, 1999

Professor Stephen G. Ritchie, Chair

This research describes an innovative distributed approach for the provision of real-time decision support to Transportation Management Center (TMC) operators for coordinated, multi-jurisdictional traffic congestion management on freeway and arterial networks. Coordinated responses among the agencies that share responsibilities for urban traffic management avoids the implementation of operations that may be conflicting or counter-productive.

A distributed software architecture, called CARTESIUS (Coordinated Adaptive Real-Time Expert System for Incident management on Urban Systems) was designed, developed and evaluated. CARTESIUS is composed of two interacting, real-time decision-support systems for TMC operator that are able to perform cooperative reasoning and resolve conflicts, for the analysis of non-recurring congestion and the formulation of suitable integrated control responses. The two agents support incident management operations for, respectively, a freeway and an adjacent arterial subnetwork. Each module interacts with a human operator in one of the agencies, is able to receive real-time traffic and control data, and provides the operator with control recommendations in response to the occurrence of incidents. The multi-decision making approach adopted by CARTESIUS reflects the spatial and administrative organization of traffic management agencies, providing a coordinated solution that attempts to satisfy all parties, preserves their own levels of authority, and reflects the inherent distribution of the decision-making power.

The structure of the distributed processing and the interaction between the agents

is based on the Functionally Accurate, Cooperative (FA/C) paradigm, a distributed problem solving approach aimed at producing consistent global solutions even when complete and up-to-date information is not directly available to the agents, in order to reduce communication requirements and synchronization time delays.

The contribution of this research lies in demonstrating the validity of the assumption that satisficing control solutions can be efficiently obtained by relaxing the requirement that agents have shared access to all globally available information, and the application of theoretical principles of the FA/C paradigm to traffic control, through the development of CARTESIUS. The simulation-based validation of the system performance has demonstrated the effectiveness of such an approach in producing real-time, integrated traffic control solutions that reduce the adverse impact of incidents on traffic circulation, network-wide.

Chapter 1

Introduction

1.1 The Need for Integrated Management of Congested Facilities

The current level of congestion in urban areas, a significant index of the degree of reduction of urban mobility and of the consequent quality of life, has already shown that the growth of traffic problems cannot be overcome simply by meeting the demand through the creation of new roads. After concentrating on the development of efficient traffic control means for the solution of localized problems, the necessity of coordinating efforts among the several agencies that share the management and control of traffic networks and the creation of a seamless transportation system has been recognized. Emphasis has been put on the development of integrated, area-wide policies among the various interacting subsystems. In large urban areas, several agencies maintain operations centers responsible for the management, control and maintenance of their component of the transport infrastructure. Even if the ultimate goal of these agencies is, in general, an efficient and environmentally sensitive use of the urban network, each agency has policies that both differ and sometimes generate conflicts. Due to the heavy interaction between different modes of transportation, such as public and private vehicles, on networks composed of subsystems with different demand and performance characteristics, such as freeways and surface streets, a significant interdependence in the control decision-making process is required. Several reports (Derr 1987, Van Aerde & Yagar 1988, Grasso, Ward, Hall, Perez & Eiger 1995) address the need of inter-agency cooperation for a more efficient resolution of the conflicts that may arise.

A central capability for Advanced Transportation Management and Information Systems (ATMIS) is a timely and efficient response to congestion (TRB 1997). Congestion is basically recurring or non-recurring, in nature. Recurring congestion is normally triggered by a daily or a periodic event, and it is generally addressed by planning suitable responses and pre-programming their implementation. Non-recurring congestion, on the other hand, is normally due to a truly unusual or unexpected event, such as excessive demand due to a special event or a sudden reduction in capacity due to an accident. Such congestion must be dealt with by implementing timely response measures aimed at ameliorating traffic conditions in the congested area and at avoiding the spread of congestion across the network (McShane & Roess 1990).

Often, both the identification of a critical situation and the implementation of response measures for its resolution actively involve several different transportation agencies that share responsibilities for traffic management on urban networks. Coordinated congestion responses among such agencies may avoid the implementation of potentially conflicting and therefore counter-productive operations. Due to the complexity of urban networks, the spatial and administrative organization of such agencies often results in a localized distribution of traffic data, such as traffic detector data used for monitoring traffic conditions across a subnetwork, and control data that provides information on the current status of the control devices operated by an agency. Furthermore, different agencies may pursue different goals and adopt different criteria to achieve those goals.

In order to maximize the benefits to be expected by real-time response to operational problems, and guarantee sufficient adherence to coordinated strategies, a cooperative effort is required, that satisfies all involved parties. Cooperation must respect the various levels of authority, guarantee privileged control of data, and in general reflect the inherent distribution of the decision-making process. The issue therefore becomes one of coordination and conflict resolution.

An approach is required that takes into account the desire of the various agencies to preserve their autonomy and maintain the control of the facilities under their jurisdiction, yet tries to exploit their willingness to cooperate and unify their problem-solving capabilities towards a potentially conflict-free, integrated response to operational problems.

1.2 Real-Time Incident Management

Real-time incident management provides a systematic approach to reduce the impacts and costs of traffic congestion, by responding in a timely fashion to the occurrence of critical situations through the formulation of adequate management directives. Such responses consist of methods to detect and identify anomalous situations, and to develop efficient control plans that minimize their effect when the network performance is negatively affected. These methods include:

- network surveillance,
- data collection and monitoring,
- detection, verification and analysis of incidents,
- identification and evaluation of traffic flow control responses,
- implementation of these responses through signal control and information to motorists,
- dispatching of emergency services,
- overall coordination,
- monitoring of the recovery process.

The identification of critical conditions and the formulation of real-time responses requires the ability to adequately capture the behavior of traffic, under both normal and congested conditions, and its response to control directives.

Incident management is thus characterized by real-time constraints, by the presence of different sources of input data, by uncertainty about the interaction between problems and control, and in general, by the lack of well-defined algorithmic procedures (Ritchie 1990, Cuenca, Hernandez & Molina 1995). Domains that share such characteristics have been increasingly addressed, in recent years, by the application of advanced software engineering techniques that adopt alternative knowledge-representation and inference methodologies, aimed not only at coping with real-time data but also at capturing real-time changes in the knowledge involved (Kirby & Parker 1994, Cohn & Harris 1992, Jämsä 1990). Knowledge-based systems (KBS) are one of the dominant types of such techniques. They address

ill-structured problems where algorithmic solutions are not available, are impractical or inadequate, and attempt to emulate basic human problem-solving that involves specialized knowledge of narrow, well-defined, domains (Ritchie 1990, Bielli, Ambrosino & Boero 1994). KBS provide methodologies that incorporate different sources of information and implement different types of control strategies, based both on heuristic and algorithmic knowledge. Such characteristics are typically required for the provision of real-time decision-support to Traffic Management Center (TMC) personnel.

1.3 Distributed Problem Solving

The challenges introduced by the issues mentioned above, and the recent theoretical and practical advances in mechanisms for distributed approaches to problem solving (Bond & Gasser 1988) strongly suggest an exploration of the applicability and the effectiveness of these mechanisms to the problem of network-wide congestion management and traffic control.

Research in *Distributed Problem Solving* (DPS) considers how the work of solving a given problem can be divided among a number of modules (often called *agents*, since each module is normally expected to act as a problem-solving entity in its own right), that cooperate at the level of dividing and sharing knowledge about the problem and about the developing solution (Lesser & Corkill 1987). Since problem solvers can cooperate in many different ways, DPS provides a rich environment to study issues in cooperation among artificial systems, and many different approaches have been defined for dealing with them. In general terms, the significance of the achievements in DPS can be identified along the following directions (Rich & Knight 1991, Bond & Gasser 1988):

- the ability to model psychological and societal interaction and problem solving, after recognizing that more and more human problem solving and activity is a cooperative effort that involves groups of people;
- the efficiency improvement that can be achieved by decomposing activities among a set of problem solvers, that can operate concurrently on subsets of the input data;
- the ability to organize systems in a modular fashion, that makes them easier to build, to understand, and to update;

- the ability to model heterogeneous problem solving, by allowing agents to adopt the knowledge representation formalisms and reasoning techniques that are most suitable for working on their parts of the problem, using the data available to them;
- the ability to represent multiple perspectives, i.e., to efficiently combine multiple decision-makers and multiple criteria, synthesizing converging ideas and resolving conflicts;
- the ability to treat problems that are inherently distributed, where input data are available at several distinct physical locations, or such that are composed of tasks that are naturally decomposable, and executable by distinct entities.

Among the several models for cooperation between distributed problem solvers, the *Functionally Accurate, Cooperative* (FA/C) (Lesser & Corkill 1981) approach is particularly suitable to address some of the key issues that arise in the context of real-time traffic control in urban networks. The goal of this approach is to permit interacting agents to cooperate effectively even though they have limited and potentially inconsistent information about the problem-solving activities of other agents. The FA/C approach was developed in response to what were perceived as deficiencies in the conventional models of how agents should interact in a distributed environment: in such models, tasks were decomposed so that each agent had sufficient data and *know-how* to solve its assigned subproblems completely and accurately, with little or no interaction with other agents. For many applications, this model of task decomposition is inadequate, since it is not always possible or convenient to decompose a problem into a set of subproblems such that there is a perfect match between the location of information, expertise, processing and communication capabilities, and the computational needs for effectively solving each subproblem. In such cases, the communication requirements and the delays involved by guaranteeing that each agent has accurate, complete, and up-to-date information may be too high to be of practical use.

Since its conception in the early 1980s, a series of increasingly sophisticated cooperative control mechanisms have been developed to guide the design and implementation of distributed systems. Most of the applications are in distributed sensor interpretation and, to a lower extent in distributed resource allocation.

Until now, the applicability of the underlying principles of the FA/C approach and their effectiveness in helping design a distributed system able to perform cooperative

reasoning with reasonable communication requirements has never been tested in the realm of distributed real-time, urban traffic incident management.

1.4 Research Objectives

The main objectives of this research were to develop a new distributed problem-solving approach for the provision of real-time decision-support to TMC personnel in multi-jurisdictional, network-wide management of non-recurring congestion. The approach is based on functionally accurate cooperation among distinct decision-making modules. This research intended to demonstrate the applicability of this approach and assess its effectiveness, through the development and evaluation of a distributed software system based on its underlying principles.

The main characteristics of this innovative approach allow to:

- reflect the jurisdictional distribution of traffic problem-solving expertise:
- provide separate access to data sources:
- reduce synchronization delay and communication overhead.

The work described in this dissertation is an application of research in Distributed Problem Solving to the development of cooperation mechanisms to effectively implement existing traffic control methodologies in a multi-decision-maker environment.

1.5 Research Contribution

The major contribution of this research is the design, development, application, and partial evaluation of a new distributed approach to multi-jurisdictional, network-wide traffic congestion management and control.

A distributed software system, called Cooperative Adaptive Real-Time Expert System for Incident management on Urban Systems (CARTESIUS) has been designed and developed. CARTESIUS' architecture is composed of two interacting real-time decision support systems that are able to perform cooperative reasoning and resolve conflicts, for the analysis of congestion phenomena and the generation of suitable network-wide, integrated responses. The evaluation of the system's performance has shown, in a simulated environment, substantial reduction in travel time and delay.

The two interacting modules build upon an existing prototype (Logi 1995, Molina, Logi, Ritchie & Cuena 1995), suitably enhanced to provide cooperation mechanisms based on the principles underlying the FA/C concept. The focus of this research is on the efficient *integration* of existing techniques for real-time generation and assessment of appropriate settings for traffic control devices, with emphasis on the delicate problem of coordinating the efforts of the agencies responsible for operations and control in urban freeway and arterial networks. The analysis of the network state and the search for satisficing control plans are based on a structured combination of heuristic approaches and well-established algorithms based on traffic flow and control theories. Previously defined and commonly used algorithms for signal timing and ramp metering are used when suitable conditions for their applicability are deemed satisfied. Given the vastness of available solutions, both in the state of the art and the state of the practice and the lack of a uniformly accepted methodology, this research does not focus on implementing the most up-to-date methodology, but rather, to apply commonly used and easy to implement algorithms, and structure their use in a more general distributed framework that provides the means for cooperation and conflict resolution among multiple decision makers. Once the efficacy of this approach is well established, it will be possible to use the architecture as a platform for testing the effectiveness of more sophisticated algorithms, by defining an opportune interface within the current modular structure.

The need for the integration of existing management and control methodologies into a structured framework, has been widely acknowledged. The issues that arise when integrating existing management and control methodologies into a structured framework and defining an interaction protocol that allows separate agencies to cooperate for the development of network-level control solutions are mostly at practical level. They include how to efficiently interface the cooperating modules, how to distribute the information among them, how to assess and limit the overhead imposed by data communication, and how the exchange of data and control directives affects the overall performance.

The design and development of this architecture was aimed at dealing with such issues, through the development of a structured methodology to build applications for integrated, multi-decision-maker congestion management.

The CARTESIUS system is intended to enhance the decision support technology available to transportation management center operators for area-wide distributed incident management and specify communication requirements between traffic management agen-

cies.

The model also provides an advancement in the synthesis between heuristics and optimization-based problem-solving methodologies.

1.6 Outline of the Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 contains an analysis of the relevant state of the art and of the practice in integrated urban traffic management and control, with a summary of the results of recent research in this area. Chapter 3 describes the functions of the centralized knowledge-based module, called TCM (Traffic Congestion Manager), upon which the two interacting agents are based, and the results of its evaluation. Chapter 4 reviews issues about distributed problem solving and introduces the cooperation paradigm on which the interaction among the agents is based. Chapter 5 presents a detailed description of the interaction among the two agents that compose the architecture, for the analysis of traffic problems and the formulation of suitable solutions, and provides a justification of the suitability of this approach. Chapter 6 provides details on the algorithms that the cooperating agents use for the selection of signal control plans, freeway ramp metering rates and the identification of suitable messages for Changeable Message Signs (CMS), i.e., panels for information to drivers that are able to display different types of messages. Chapter 7 describes the process and the results of the simulation-based evaluation of the system performance in providing network-wide responses to the occurrence of incidents artificially injected in the simulation. Chapter 8 concludes this dissertation, by highlighting the main contributions of this work and pointing directions for expansion of some of the ideas presented herein.

Chapter 2

Integrated Traffic Management

Euler (1990) identified several characteristics that describe ATMIS and that differentiate them from traditional transportation management systems. These are:

- ATMIS work in real-time.
- ATMIS respond to changes in traffic flows. In fact, they may often be a step ahead in predicting when and where congestion is likely to occur based on combined real-time and historical information.
- ATMIS include area-wide surveillance and detection systems.
- ATMIS integrate management of several functions, including demand management, freeway ramp metering, arterial signal control, and navigation information to motorists about roadway conditions, through audio and visual means.
- ATMIS imply collaborative action on the part of the transportation management agencies and jurisdictions involved.
- ATMIS include rapid response incident management strategies.

The existence of integrated transportation management and information systems aims to facilitate both the provision of better information to motorists and the application of better control strategies.

Several definitions have been proposed for integrated traffic control systems, and several classifications have been identified, based on their characteristics. Diakaki, Papa-georgiou & McLean (1997), for example propose a taxonomy that distinguishes between

aspects of integrated modeling of traffic processes, integrated design of control strategies, integration of software components, integration of communication and hardware facilities, integration of control centers, and integration at an administrative level. Euler & Lindley (1989) highlight the distinction between hardware and functional integration, although they recognize that the first in fact provides the tools that lead to the second. Hardware integration may range from the development and installation of data-sharing communication systems to the establishment of a single control or operations center. Functional integration includes sharing information among several control systems, usually computerized arterial signal control and freeway surveillance and control systems, but it may extend to other systems such as transit operations and commercial vehicle monitoring systems.

In response to the recognized needs of integrated, area-wide transportation management infrastructures, several research and development projects are currently underway, both in the US and in other countries. The remainder of this chapter summarizes the results of recent research in this area and describes several ongoing development projects. In particular, in the next sections, after an introduction of the leading research directions in the field of integrated transportation management, four research projects are presented in more detail, with a description of their contribution to the state of the art and a discussion of their limitations. The first and probably most important project is the Santa Monica Smart Corridor (Roseman 1994), a very advanced research project that reached full operational status in 1996, and that deals with institutional issues related to the partition and assignment of control roles as well as with more strictly operational issues, such as data communication links and the quantity and quality of data shared among participating agencies. Another research project that is closely related to the research described in this dissertation is the City of Irvine Field Operational Test (FOT) project, a partnership of public and private agencies designed to deploy and evaluate Intelligent Transportation Systems (ITS) concepts and technologies for corridor incident management and traffic control. From the point of view of real-time development and implementation of control plans, corridor control techniques in Glasgow, based on automatic control theory and described in Diakaki, Papageorgiou & McLean (1998), constitute an interesting approach to the problem of development and implementation of integrated control schemes for traffic diversion via CMS and signal control. Tsavachidis, Hoops & Keller (1998) analyze a similar problem, the freeway access to the city of Munich, and report on the modification and integration of previously developed traffic management systems for a coordinated approach that attempts

to improve critical freeway operations by taking advantage of spare capacities on adjacent arterials.

2.1 Integrated Advanced Transportation Management and Information Systems

Some research projects aim at the development of area-wide control strategies and the evaluation of the potential benefits that can be expected from the integrated control of freeway and adjacent arterial subnetworks. In these cases, the focus is often on determining conditions for which such strategies offer improvements and the selection of globally compatible control solutions.

Pooran & Lieu (1994b) have developed system operating strategies for the coordinated operation of ramp metering and adjacent traffic signal systems, in order to improve the operational performance of corridors composed of freeways and parallel arterials. They propose a conceptual design that provides guidelines for the selection and implementation of appropriate high-level strategies and lower-level control actions (*tactics*). They designed a central Integration System (IS), capable of interfacing with both freeway ramp control and arterial traffic signal control systems. The system receives data describing network conditions from both control systems and, based on a set of pre-specified thresholds, assesses the need to select a control strategy from a pre-specified database. The selected strategy is then translated into a set of tactics, through an automatic look-up table that identifies the possible tactics associated with each strategy. Pooran & Lieu (1994a) report on the simulation-based evaluation of the implementation of different strategies under different levels of coordination (base-case, minimal interaction, and maximum interaction among the control centers), generally showing a modest increase in the network performance, with respect to the freeway, arterial, and ramp subnetworks. Chang, Ho, Liu & Hsu (1995) describe the process of development of an integrated freeway-management expert system (IFMES) to support freeway and expressway incident management in Taiwan. The system is based on a traffic diversion control module, that includes subsystems for flow prediction, origin-destination (OD) estimation, and dynamic traffic assignment, and is expected to provide real-time traffic control in response to critical freeway incidents. The Commission of the European Union in the last decade has also sponsored several research

and development projects (DRIVE, PROMETHEUS, JUPITER, SLAM (Commission of the European Communities 1991)) aimed at developing integrated traffic control strategies, such as policies between traffic control and driver information (Durand-Raucher 1995, Cuenca et al. 1995), and improvement of public transport traveler information (Ambrosino, Boero & Turrini 1995).

These research projects, aimed at the identification of suitable conditions for the implementation of network-wide control and the development of strategies that make use of various control techniques in an integrated fashion, represent a fundamental step towards the realization of fully-integrated transportation management systems. It is imperative, however, that such a step is corroborated by the analysis of the overhead imposed by the complexity of inter-agency communication and the recognition of important institutional issues that come into play when proposing modifications to existing control systems, such as the presence of several management agencies that share the retrieval of information and the responsibilities over the control of the subsystems composing urban networks.

Another important research branch focuses on the institutional issues related to the deployment of integrated ATMS. Some research projects try to address the complex problem of data exchange and responsibility allocation between various agencies. The Portland ITS master plan (Lee & McCourt 1995), for instance, calls for the development of an interagency jurisdictional framework that identifies agency roles and key coordination issues, aimed at providing benefits such as expeditious incident detection, confirmation and response, and implementation of effective traffic control strategies, through the efficient information sharing and the clear identification of management responsibilities. The Portland ITS master plan engages as many as 60 cooperating agencies, including traffic operation and management centers, transit agencies, and emergency services.

In the state of Washington, the North Seattle ATMS (Sumner, Jacobson & Berg 1995, *North Seattle ATMS* 1997), is an on-going multi-jurisdictional project for control systems operations and information sharing. The project, which brings together several city, regional traffic, and transit operations agencies, as well as the Washington State Department of Transportation, intends to define a geographically distributed traffic management and information system, based on the development of a common communication link, using the National Transportation Communications/ITS Protocol (NTC/IP) (NTCIP Steering Group 1995) between the traffic and transit operations and control systems involved. The primary objectives of the project, scheduled to terminate by the end of 1998, include the

implementation of a regional monitoring and data sharing system for the development of coordinated operations between jurisdictions. The sharing of these data, which include traffic data (detector volumes and occupancy rates), control data (signal and ramp metering parameters), incident data, and bus location and identification data, will allow the agencies to implement preplanned diversion plans whenever specific traffic thresholds, previously mutually agreed upon, have been exceeded.

2.2 The Santa Monica Smart Corridor

In Los Angeles, the Santa Monica Smart Corridor (Roseman 1994, Clelland 1993, Karimi, Gupta, Wheeler & Massarano 1992, Rowe 1989) is one of the most advanced ITS operational projects in the world. The Smart Corridor (SC) provides electronic connection and coordination in freeway and arterial data gathering, develops and implements computer generated action plans, and supplies motorists with real-time congestion information within a corridor, thus representing a very important operational test of the integrated applications of a variety of ITS technologies and transportation management strategies. The SC system is designed to use the resources of all participating agencies to detect incidents and provide decision support for the management of congestion on a network of freeway and adjacent surface streets. The major objectives of the SC include the operation of the individual facilities within the corridor at maximum efficiency, the balancing of traffic flow between the freeway and parallel arterials in order to optimize the use of existing capacity and minimize delay and congestion, and the management and dissemination of motorist information. The SC project area consists of a 14.5 mile stretch of the Santa Monica Freeway (Interstate 10), one of the most heavily traveled freeways in the United States, between the Santa Ana Freeway (Interstate 5) and the San Diego Freeway (Interstate 405), and 378 miles of surface streets, including five major parallel arterials, as shown in Figure 2.1.

Several public agencies are involved in the SC, including the State of California Department of Transportation (Caltrans), California Highway Patrol (CHP), the Los Angeles Department of Transportation (LADOT), the Los Angeles County Metropolitan Transportation Authority (LACMTA), the City of Beverly Hills and the City of Culver City. Caltrans and the CHP are responsible for freeway operations, LACMTA for transit operation, and each of the cities is responsible for surface street and signal operation. The operational activities of LADOT's Automated Traffic Surveillance and Control (ATSAC)

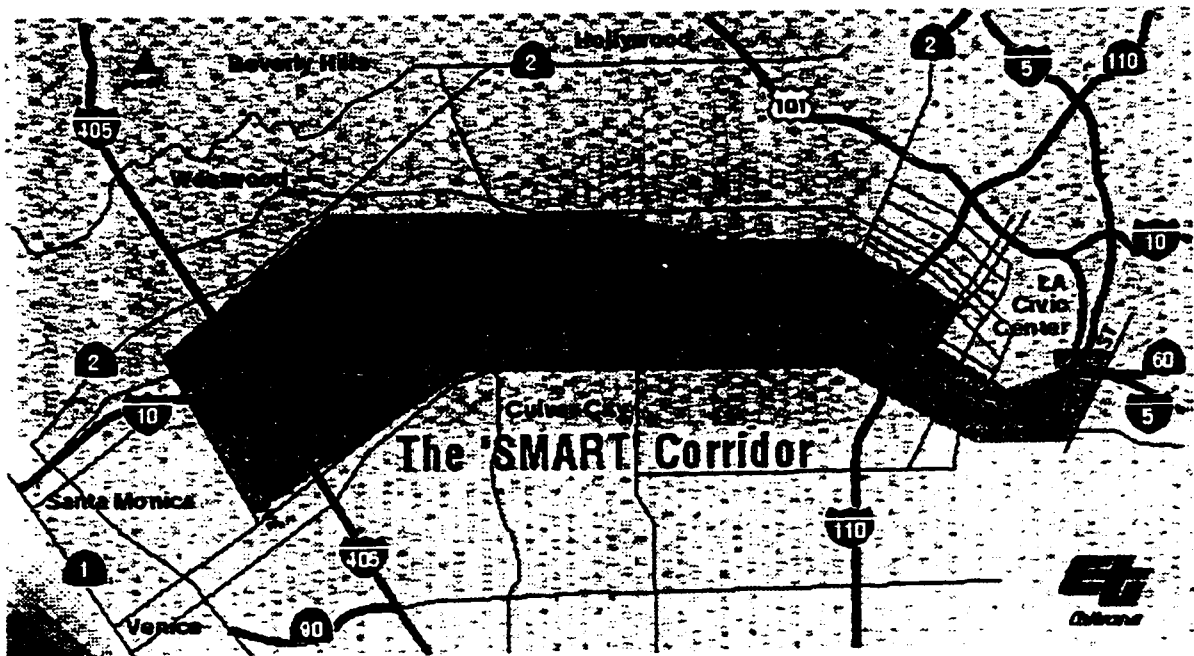


Figure 2.1: The Santa Monica Freeway Smart Corridor

Operations Center, Caltrans' Traffic Operations Center, and the CHP Los Angeles Communications Center are electronically linked through a network of distributed databases. Operators in the various agencies interact with each other and with the SC system through a network composed of nine PC-based graphical user interface (GUI) workstations and eleven "backbone" Sun SPARCstations.

The implementation of the SC system was targeted at supporting the congestion management activities of the various agencies by building upon the existing infrastructure and providing the means by which these agencies can coordinate their operations. The SC thus is an "umbrella system" which incorporates new and existing systems into a single coordinated management architecture which can be accessed by any agency operator. It supports an environment in which agency operators have access to *all* relevant data and resources within the corridor, regardless of agency ownership, through a common GUI.

2.2.1 The System Architecture

The SC architecture is based on an object-oriented design approach to information retrieval, processing, analysis and control. Clelland (1993) describes the system architecture

as composed of the following subsystems:

Agency Interface. Provides the interface between the SC system and the stand-alone computer systems in the operations centers of the various agencies.

Data/Data Management. A distributed database composed of three separate agency databases organized into a common data manager that stores all static and dynamic information for access to all systems.

Incident Management. A network of three expert systems that provide decision support for automated incident detection, correlation, confirmation, and response, and dynamic incident management.

Operator Interface. The GUI for inter-operator communication

Public Advisory. The interface between the SC system and the Motorist Information Systems, that include CMS, Highway Advisory Radio (HAR), Highway Advisory Telephone (HAT), Dial-up Services, and the Media Interface.

Communications. The communication media for transfer of information among subsystems and processes within the system.

2.2.2 Incident Management

Multi-jurisdictional dynamic incident detection, analysis and response require the design of a system that is operationally robust while respecting individual agency authority and minimizing liability exposure. During the design phase, institutional and legal issues such as determining how to assign the control of the system's multi-agency operation required at least as much consideration as software design issues. The original design proposal included the development of an "autocratic" corridor management system organized as a single *Corridor Manager* (CM) system. The CM was intended to have responsibility for operating the corridor, confirming incidents, modifying and approving system generated response plans. Agency authority would be respected by allowing each agency veto power over control plans relating to facilities under their jurisdiction, thus, at the same time, addressing potential liability issues. Each affected agency, when presented with a control proposal approved by the CM, would have the choice between accepting or refusing the actions on its control facilities. In case of refusal, the CM would have to reconsider the

problem with the new constraints introduced by the agency's response. Several problems were identified with such a design: operator conflict resolution would have the potential of substantially slowing down the system response, by requiring back-and-forth colloquia with each operator. Furthermore, the Corridor Manager concept presented a number of operational and institutional problems. In particular the following issues arose:

- What agency should be in charge of the CM? Caltrans, LADOT, or a third party? What level of authority would be given to a third party? How would third party operation be funded?
- Should CM responsibilities be rotated among the participating agencies? What would be the assignment based on:
- Would LADOT be willing to let Caltrans make control decisions on signal plans selection? Would Caltrans allow LADOT the selection of ramp metering rates?
- Would the agencies be willing to accept the additional responsibility and potential liability for operations outside their traditional jurisdictions?
- Would a single CM be able to effectively manage multiple concurrent incidents? Would there be the potential for operator overload?

These and other issues led to a modification of the CM concept and the development of the *Incident Manager* (IM) control concept, by which expert system software would dynamically assign incident management responsibilities to agency operators based on the location of a detected incident. In this way, rather than having a single corridor manager, the management of each single incident is assigned to an agency which then becomes the IM for that incident. If an incident is detected on the freeway, Caltrans assumes the responsibility, and becomes the IM for that incident. Analogously, if an incident is detected on a surface street, the local city responsible for the affected location becomes the IM for that incident. The IM thus becomes responsible for directing operations that may or may not involve other jurisdictions. The key assumption is that pooling the collective skills of multiple agency operators and placing the agency with political jurisdiction and typically the most operational expertise in the area at the lead of the response provides a more competent corridor management than that of a single CM responsible for operations corridor-wide.

By distributing incident management responsibilities across multiple operators, the system evolves into a decision support tool for the individual operator and the facilitator of multi-agency coordination and response.

The IM control structure performs the following functions:

Incident Detection. Incident notification is generated by Caltrans' Semi-Automated Traffic Management System (SATMS), LADOT's arterial incident detection algorithm, CHP's Computer Aided Dispatching system, and manual reports by operators.

Incident Correlation. Incident reports from the various sources are analyzed and correlated into an incident summary to minimize multiple notifications of a single incident.

Incident Manager Assignment. Based on the incident location, an operator of the agency under whose jurisdiction the incident is located becomes the Incident Manager, responsible for directing the response.

Incident Confirmation. The system requests the IM to confirm the incident. Confirmation is normally performed through the use of Closed Circuit Television (CCTV) cameras.

Response Plan Generation. This step includes the estimation of the expected incident duration, the analysis of current traffic flows relative to the reduced capacity over the estimated duration of the incident, the selection of goals to mitigate the impact of the incident, the development of goal-supporting actions, and the presentation of goals and actions to the IM. These preplanned traffic management strategies address both recurrent and non-recurrent congestion.

Response Approval. The IM can modify the response goals and only the actions related to his or her agency. The other agencies are then presented with the plan and a request for the execution of the appropriate actions to support the suggested goals. The agencies can approve, modify or reject the control actions on their facilities.

Plan Implementation. The approved management actions are implemented by the system through the various agencies.

Dynamic Incident Management. Traffic conditions around the incident location are monitored, and new response plans are recommended to the IM based on the dynamics of the incident.

Response Termination. When conditions return to normal, the system recommends the termination of response control actions.

When the response plan includes multi-agency actions, as it is often the case for major incidents, each control action must be approved by the agency that is responsible for the operation of the corresponding control device. The system performs an operational request to the agency for action approval, providing information about the global plan approved by the IM. Each agency may slightly modify the actions, as long as they are supporting the requested goals.

During the Dynamic Incident Management phase, if actions need to be modified in response to the dynamics of the incident, the system requests those changes for approval directly to the affected agency, but actions must always be approved also by the IM.

Coordination among autonomous operators is achieved by acknowledging that each agency has specific roles and expertise, and successful incident management evolves through the agreement on coordinated response goals and the translation of such goals into control actions by the agencies that have the highest expertise needed to determine how to reach those goals.

The multi-agency Incident Manager architecture provides a control environment in which agencies can coordinate response actions while maintaining a certain degree of autonomy. The complex data communication links that support sharing of information among the involved agencies provide each agency with access to corridor-wide resources and decision-support, at the same time preserving their authority and limiting their liability exposure.

The project officially became operational during a kick-off ceremony held on October 11, 1996 at the Los Angeles Public Library. The total cost of the eight-year project was approximately \$48 million which was provided through federal, state and local sources. The estimated maintenance and operation cost is \$2.7 million a year.

2.2.3 Critique

The major limitation of the SC system lies in the rather rigid structure of the incident response process that includes the selection of pre-planned actions by the IM, based on previously agreed upon guidelines. Even if each agency has ultimate responsibility for control actions related to devices under its jurisdiction, the decision process performed

by the IM expert system, that leads to the proposal of a network-wide response, is not a truly multi-decision-maker process. The control structure of the SC places the agency that currently has the role of IM in a privileged position within the decision-making process. With such a control structure, the only way to prevent the system from proposing solutions that are likely to be vetoed by other agencies, causing time-consuming interaction between operators of different agencies to reach a mutually satisfying agreement, is to base the decision-making process on the verification of previously agreed upon conditions and the selection of prespecified control responses. Furthermore, in order for the expert system to propose truly integrated response, it is necessary for the IM to have access to all relevant data, which is the ground reason for the existence of the complex database that provides each agency with shared access to all information sources across the corridor. A more dynamic decision-making process can be achieved by designing a distributed, multiple decision-maker architecture in which separate reasoning units exchange goals and partial results during their decision-making process, and are able to propose an integrated combination of control plans that results from a truly dynamic decision-making process. Within such a structure, the selection of appropriate settings for local control devices, that verify local and global compatibility criteria, is performed by the agency that is responsible for the operation of such devices, and the validity of the resulting global plans is assessed by a mutual agreement of the interacting agencies. This type of control structure is the one described in this dissertation.

2.3 The City of Irvine Field Operational Test

The City of Irvine FOT brought together freeway and arterial transportation management agencies (California Department of Transportation (Caltrans) District 12 and the City of Irvine) and private agencies (PB Farradyne Inc. and National Engineering Technology Corporation) for the deployment and evaluation of coordinated traffic control strategies in a highly congested, rapidly growing corridor network in Southern California, composed of a 4-mile section of the Interstate-405 Freeway and a subnetwork of adjacent arterials, with a major arterial, Alton Parkway, acting as possible alternative in case of freeway congestion. The FOT (Mattingly & McNally 1998) aims to integrate and coordinate the System Wide Adaptive Ramp Metering (SWARM) system, a centrally-controlled freeway ramp metering system (National Engineering Technology Corporation 1996), with a cen-

trally monitored, adaptive arterial traffic management system that uses Optimized Policies for Adaptive Control (OPAC) (Gartner 1983).

The FOT's goals include the identification of:

- the potential of adaptive traffic signal control techniques to accommodate major traffic transients arising from freeway diversion;
- the potential of centrally-controlled ramp metering strategies to prevent and respond to the occurrence of congestion by optimizing flow on freeway sections;
- the potential of advanced traffic controllers to incorporate adaptive traffic control algorithms in a real-time dynamic environment;
- the potential and effectiveness of inter-agency cooperation for joint traffic system management

As part of the project, an Advanced Traffic Management System is being developed for Caltrans District 12 (Orange County), which is capable of freeway surveillance, incident detection, and response, through selection from a database of pre-planned response plans that are subject to the TMC operator for approval. Responses take into account freeway real-time traffic conditions and consist of actions from both Caltrans and the City of Irvine. The city of Irvine TMC is being equipped with the Management Information System for Transportation (MIST) (Farradyne Systems Inc. 1996), a data management and communication system between operators and signal controllers. As part of the FOT, new 2070 Advanced Traffic Controllers (ATC) are being installed for control of signalized intersections, freeway ramp meters, and arterial CMS.

The main components of the Irvine FOT are briefly described in the remainder of this section.

2.3.1 Ramp Metering

SWARM is composed of two algorithms: an algorithm for system-wide ramp metering that uses traffic density forecasting to predict and possibly avoid the occurrence of congestion on freeway sections (SWARM 1) and a local adaptive traffic responsive algorithm (SWARM 2).

SWARM 1 uses measured freeway volume and occupancy as input to a forecasting procedure based on linear regression and Kalman Filtering techniques for the prediction

of short-term density trends. The procedure is repeated every time new data are available (e.g., every 30 seconds). The controlled freeway network is partitioned into sections and parameters are computed for each section. A *saturation density*, corresponding to the maximum desired density level is defined. When the forecast value of density exceeds the saturation density, metering rates are computed that attempt to reduce the downstream density, through a reduction of the upstream input volumes (from metered ramps). A target density is computed based on the current density, the excess density (i.e., the difference between predicted and saturation density), and the length of the look-ahead interval (i.e., the time difference between the current time and the time at which excess density is expected). The target density is transformed into a target volume, that identifies the volume reduction necessary to reach the target density. Then, starting at the section that is expected to reach excess density and working upstream, metering rates are computed based on the computed ramp volume reductions (subject to prespecified minimum and maximum rates). Excess volumes are propagated to ramps corresponding to upstream sections.

SWARM 2 is an adaptive algorithm that attempts to control headway between vehicles (the time between successive vehicles) on each section to reach a density that corresponds to maximum flows.

Both SWARM 1 and SWARM 2 algorithms are supposed to run continuously and the SWARM system selects the most restrictive rate among the two. SWARM contains procedures for data completion, to be used, for instance, when detector failures cause data loss. Available data are used if they are above minimum required data thresholds. If the available data are judged insufficient for analysis (i.e., below the thresholds), the corresponding detection station is removed from the analysis, and data from upstream detectors are used instead.

2.3.2 Adaptive Signal Control

OPAC is an on-line optimization algorithm for signal control that computes signal timing parameters to optimize a performance function representative of the total intersection delay and stop times, for isolated intersections. In order to compute phase duration, OPAC uses measured intersection volumes (from upstream detectors) and a modeled backbone demand. The optimization process uses a rolling horizon approach, whereby at each time step (typically a few seconds) current upstream volumes are used for a short-term prediction

of arrival patterns that determine the optimal phasing (based on the optimization of the performance function).

From its initial design (Gartner 1983), the algorithm has undergone several enhancements (Gartner 1991) and evaluation, both in simulated environments and in real-world implementation (Andrews, Elahi & Clark 1997). Improvements include the modeling of left turn movements and the specification of left turn phasing, the inclusion of pedestrian requirements, the ability to optimize phases according to standard NEMA (National Electric Manufacturers Association) configurations, and the provision of coordinated operation (Farradyne Systems Inc. 1996). As part of the Irvine FOT, OPAC's functions were to be tested using the new 2070 ATC.

2.3.3 Data Management and Communication

MIST is a computer-based system designed for real-time management, monitoring, and transmission of transportation-related information between operations centers and field devices, such as signal controllers, CMS, and detector stations. MIST is intended to reduce operator overload, by automating and facilitating many of the traffic control tasks that require monitoring of input data and transmission of control directives.

MIST will use a database management system that facilitates retrieval and storage of information, and the specification of several types of user access rights to different items in the database. MIST will monitor communication equipment and data sources and is capable of notifying operators in case of malfunction or data corruption.

MIST provides static and dynamic (real-time) reports on the status of field devices, and has a graphical user interface (GUI) that allows the operator to issue commands to automatically control those devices, through the selection of pre-specified plans (signal timing plans and CMS messages) from the database and through the specification of parameters for the construction of new plans. In its current state, though, MIST's ability to accept control parameters is limited, and the issue of control commands is possible only through its GUI. Current modifications call for the system to be able to receive and transmit control data in a programmable way (i.e., without using the GUI).

2.3.4 Critique

At the time of writing, the Irvine Field Operational Test is yet to be evaluated. Evaluation results will provide insights on the applicability of the techniques used and of the effectiveness of their integration, as well as a documentation and standardization of the development process to assist implementation elsewhere.

Emphasis is placed on the analysis of the benefits expected by the mutual exchange of data and control information between operations centers for the identification and analysis of non-recurring congestion, and the selection of previously agreed upon, coordinated control strategies that involve traffic diversion from the freeway to the arterial network, system-wide ramp metering and adaptive signal control.

Analogously to the Santa Monica Smart Corridor project, the selection of control responses is currently limited to pre-constructed control plans that clearly define the role of both agencies in the response process. The signal control algorithm OPAC was field-tested in New Jersey, by comparing its performance against a time-of-day signal plan under several different demand conditions. The evaluation was performed for both isolated intersections and arterial control and showed a significant reduction in travel time and stop time (Andrews et al. 1997). It will be interesting to assess the ability of the system to respond to major traffic transients arising from freeway diversion and be actively incorporated in coordinated freeway/arterial traffic management strategies.

2.4 Integrated Corridor Control in Glasgow

Glasgow (Scotland) is one of the test sites of the European project TABASCO (Telematics Applications in Bavaria, Scotland and Others) (TAB 1997). The realization of the potential benefits achievable by the appropriate integration of advanced ITS traffic management strategies, resulted from simulated investigations of the impact of coordinated operation of ramp metering and traffic diversion, within the DRIVE-II project EUROCOR (European Urban Corridor Control) (Diakaki et al. 1997). An INtegrated Traffic-responsive Urban Corridor control strategy (IN-TUC) was developed and applied to an urban network in the city of Glasgow composed of a 4-km section of a heavily traveled motorway and a subnetwork of adjacent urban arterials with seven signalized intersections, two motorway entry ramps and three CMS (Diakaki et al. 1998).

The IN-TUC strategy is based upon methods of automatic control theory and is composed of two main elements: signal control and CMS control. In its initial phase, IN-TUC does not explicitly perform ramp metering, even though the simulated studies described in Diakaki et al. (1997) included the integration of the ALINEA adaptive ramp metering algorithm (Papageorgiou, Haj-Salem & Blosseville 1991) with traffic diversion strategies. The integration of control strategies lies in the mutual exchange of information between the two algorithms, so that the computation of path travel time, which is used to determine the appropriate CMS messages, depends on the current control at signalized intersections. The current formulation does not allow for signal control to take advantage of information about path volumes resulting from traffic diversion, but the authors are currently investigating this issue.

2.4.1 Signal Control

Signal plan optimization is based on the store-and-forward network modeling approach proposed by D'Ans & Gazis (1976). The algorithm uses rather coarse simplifying assumptions to determine suitable green times at intersections based on volumes on approaching links.

The network is represented as a graph (Z, J) , with links $z \in Z$ and nodes $j \in J$. Vehicles are assumed to experience a constant travel time on a link and are stored at the end of the link if the outflow capacity is smaller than the inflow demand. Both the inflow and the outflow depend on the movement capacity at the upstream and downstream controlled junctions. A signalized intersection j has a set of incoming links I_j and a set of outgoing links O_j , as shown in Figure 2.2 a). It is assumed that all permissible movements from link z have simultaneous right-of-way (r.o.w), and that the cycle time C_j and the intersection lost time L_j are fixed. It is assumed $C_j = C$ for all intersections $j \in J$. The intersection cycle time is partitioned into a fixed number of stages belonging to the set F_j . The set $\nu_z \subset F_j$ denotes the set of stages during which link z has r.o.w. The saturation flows S_z are known and the turning movement rates t_{zw} , $z \in I_j, w \in O_j$ are assumed known and fixed. The effective green time of stage i at intersection j is denoted by g_{ji} . The following constraints must hold:

$$\sum_{i \in F_j} g_{ji} + L_j = C \quad \forall j \in J$$

$$g_{ji} \geq g_{ji_{\min}} \quad \forall i \in F_j \quad \forall j \in J$$

where $g_{ji_{min}}$ is the minimum permissible effective green time.

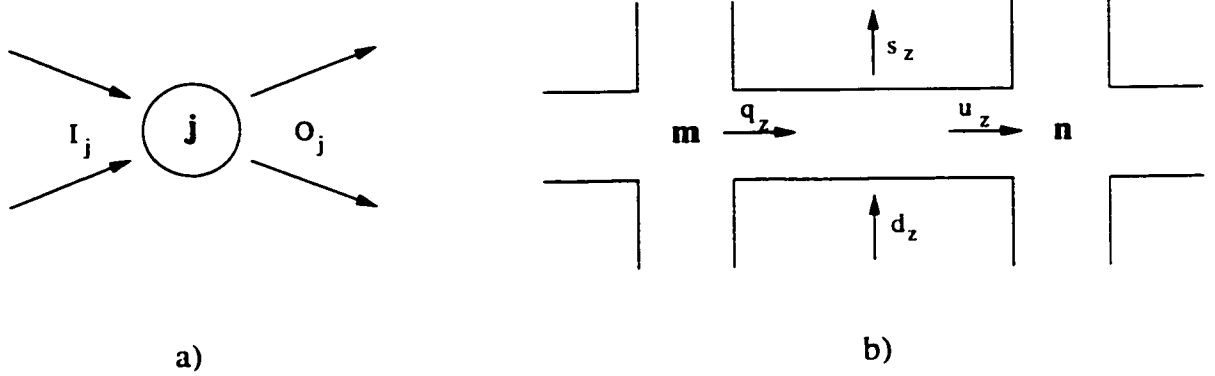


Figure 2.2: An intersection and a link in the graph (Z, J)

For each link z connecting two signalized intersections m and n , $z \in O_m$ and $z \in I_n$, x_z , q_z , and u_z are respectively the number of vehicles on link z , the inflow, and the outflow for link z over the period $[kT, (k+1)T]$, where T is the control time interval, as shown in Figure 2.2 b). The travel time on link z is the summation of a constant travel time (free flow) $\kappa_z T$, and a queueing delay due to the number of vehicles on link z . The number of vehicles on link z is regulated by the following equation

$$x_z(k+1) = x_z(k) + T[q_z(k - \kappa_z) - s_z(k - \kappa_z) + d_z(k) - u_z(k)] \quad (2.1)$$

where d_z and s_z are respectively the demand and the exit flows. The exit flow is given by

$$s_z(k) = t_{z0} q_z(k) \quad (2.2)$$

assuming the exit rates t_{z0} known and fixed. The demand flow is assumed to be null. Choosing the control interval so that $\kappa_z = 0$, equations 2.1 and 2.2 lead to

$$x_z(k+1) = x_z(k) + T[(1 - t_{z0})q_z(k) - u_z(k)] \quad (2.3)$$

The inflow of link z is given by

$$q_z(k) = \sum_{w \in I_m} t_{wz} v_w(k) \quad (2.4)$$

The exit flow of link z is assumed to be equal to the saturation flow S_z when the link has r.o.w. and null otherwise. This requires the assumption that the link demand is sufficiently high (i.e., none of the movement capacity is wasted) and that vehicles can

move to the downstream links, implying that there is no *de facto red* condition (a *de facto red* condition occurs when, due to downstream congestion, vehicles can not move past the intersection, even when they have r.o.w.). This assumption may not always be justified, especially in incident management, when *de facto red* conditions may occur. This is not a major shortcoming, though. It only requires an analysis of the available capacity on the downstream link. When the control interval T is higher than the cycle time C , an average value is computed for the exit flow as follows:

$$u_z(k) = \frac{S_z G_z(k)}{C} \quad (2.5)$$

where the effective green time assigned to the movements of link z is

$$G_z(k) = \sum_{i \in \nu_z} g_{ni}(k) \quad (2.6)$$

Introducing equations 2.4, 2.5, 2.6 into equation 2.3, and subtracting from the resulting equation its steady-state version, the following state equation is obtained

$$x_z(k+1) = x_z(k) + T \left[(1 - t_{z0}) \sum_{w \in I_m} t_{wz} \frac{S_w \sum_{i \in \nu_w} \Delta g_{mi}(k)}{C} - \frac{S_z \sum_{i \in \nu_z} \Delta g_{ni}(k)}{C} \right] \quad (2.7)$$

where $\Delta g = g - g^N$ with g^N the nominal value of the green time, i.e., the default value that characterizes steady-state conditions.

For a system of signalized intersection, equation 2.7 becomes

$$\mathbf{x}(k+1) = \mathbf{A} \mathbf{x}(k) + \mathbf{B} \Delta \mathbf{g}(k)$$

where $\mathbf{x}$ indicates the number of vehicles on links $z \in Z$, $\Delta \mathbf{g}$ is the vector of the deviations from the nominal green times $\Delta g_{ji} = g_{ji} - g_{ji}^N, \forall j \in J, \forall i \in Fj$, and $\mathbf{A} = \mathbf{I}$ and $\mathbf{B}$ are, respectively, the state and input matrices.

The problem objective, expressed in a linear quadratic (LQ) formulation, is to balance an approximation of the ratio of volume to capacity for all links by attempting to equalize the ratio $x_z/x_{z_{max}}$ for all z , where $x_{z_{max}}$ is the desired number of vehicles on link z . The objective function, which has to be minimized, is

$$\mathfrak{J} = \frac{1}{2} \sum_{k=0}^{\infty} (\|\mathbf{x}(k)\|_{\mathbf{Q}}^2 + \|\Delta \mathbf{g}(k)\|_{\mathbf{R}}^2)$$

where $\mathbf{Q}$ and $\mathbf{R}$ are nonnegative definite, diagonal weighting matrices. The diagonal elements of $\mathbf{Q}$ are set to the inverse of the maximum number of vehicles on the corresponding

link. The green times, that determine the downstream movement (link) capacity, are varied around the nominal values g^N . The inclusion of the $\Delta \mathbf{g}$ component avoids the selection of too high deviations from the nominal greens. The weighting matrix $\mathbf{R}$ determines the magnitude of those deviations, and it is determined by a trial-and-error process.

The minimization of the objective function (Anderson & Moore 1990) leads to the following control law

$$\mathbf{g}(k) = \mathbf{g}^N - \mathbf{L}\mathbf{x}(k)$$

where $\mathbf{L}$ is the constant feedback gain matrix (the control matrix), which depends on $\mathbf{A}, \mathbf{B}, \mathbf{Q}$, and $\mathbf{R}$.

Since the link density can not be directly measured, occupancy measurements, obtainable from inductance loop detectors, are used instead.

2.4.2 Traffic Diversion

The control of CMS for traffic diversion (Diakaki et al. 1997) aims to determine appropriate CMS settings in order to equalize the travel time on alternative paths, based on feedback control. This approach uses a measure of the difference between simultaneous travel times on alternative paths to a given destination, to determine a “desired” degree of diversion, which varies from the “nominal” degree of diversion, that corresponds to the default split among alternative paths. A basic assumption of this approach is the availability of a function that allows determination of suitable CMS settings from a desired degree of diversion.

Assuming that for each CMS r in the set R of CMS there are two alternative paths 1 and 2, from the CMS location to a given destination, the *desired* portion of vehicles that should take path 1 is computed as follows:

$$\beta_{r1}(k) = \beta_{r1}^N - K_r(\tau_{r1}(k) - \tau_{r2}(k)) \quad (2.8)$$

where β_{r1}^N is the nominal value of β_{r1} , K_r is a regulation parameter which determines the magnitude of the feedback control (selected through a trial-and-error process), and τ_{r1} and τ_{r2} are the measured or estimated instantaneous travel times on paths 1 and 2.

Letting L_{ri} be the set of links on path i downstream of the location of CMS r , the path travel times are computed by

$$\tau_{ri}(k) = \sum_{z \in L_{ri}} \tau_z(k) \quad \text{for } i = 1, 2$$

where τ_z is the current travel time on link z .

The computation of the link travel time depends in part on the currently applied control (hence the integrated approach) and is computed based on an average value of the link speed for freeway sections, and the approach capacity and saturation flow at the downstream approach, for links upstream of signalized intersections and metered ramps.

The target path load ratios must also satisfy the constraints

$$0 \leq \beta_{ri} \leq 1 \quad \text{for } i = 1, 2$$

According to equation 2.8, the provision of information on instantaneous path travel times has the effect of modifying the route split at a decision point (the node or the link at which the CMS message is read by drivers, causing them to reevaluate their route decision) from the nominal value, of a quantity that is directly proportional to the oscillation parameter κ_z , and the difference between the path travel times.

The selection of appropriate CMS messages is driven by a transformation function that binds available messages to β -intervals, as follows:

$$D_r(k) = \begin{cases} display_1 & \text{if } 0 \leq \beta_{r1}(k) < \beta_{r1T_1} \\ display_2 & \text{if } \beta_{r1T_1} \leq \beta_{r1}(k) < \beta_{r1T_2} \\ \dots & \dots \end{cases}$$

where D_r is the setting selected for CMS r and $\beta_{r1T_1}, \beta_{r1T_2} \dots$ are threshold values of β_{r1} .

2.4.3 Simulation Results

The IN-TUC strategy was applied to a simulated version of the motorway corridor in the Glasgow subnetwork. In the simulated environment, the computation of control parameters drives the selection of predefined urban signal plans, from a previously specified database. At the time of writing, the complete results of the investigation have not been published, but the authors report preliminary results that show a reduction in the range of 1-2 percent of the total travel time in a network composed of 27 signalized intersections, seven of which were controlled by the IN-TUC strategy, which leads them to estimate a somewhat greater travel time reduction (5-10 percent) for the sub-area controlled by the strategy.

2.4.4 Critique

The IN-TUC strategy has two major limitations. The first one, with respect to the concept of integration, is the independence of the signal control strategy from the CMS control currently performed. The strategy assumes that exit rates at signalized intersections are fixed, when in reality, providing travelers with information should produce a change in the traffic flow distribution across the network, that is not captured if fixed turning ratios are assumed. A truly integrated control strategy requires a signal control module that takes into consideration the effect of real-time information on the flow distribution.

Another major shortcoming of the IN-TUC strategy lies in the assumption of the existence of a transformation function that relates desired levels of diversion to different CMS displays. The exact computation of such a function is by no means a trivial issue, since even though a number of researchers have explored the problem of assessing driver's compliance to CMS messages, to date no algorithm or model has yet demonstrated its superiority over the others and, in the literature, estimation of driver response to CMS varies extensively. Even if such a function were available, practical considerations about the type and quantity of information that can be conveyed through CMS would necessarily make it a rather coarse one. With such a function as the base for the selection of CMS messages, it is very likely that the computation that leads to the identification of the desired level of splitting among alternative paths loses significant accuracy. Indeed, once the desired splits have been determined, a wide gap between the desired degree of diversion and the one realistically achieved by the selection of a CMS display based on such a function, is to be expected. This shortcoming is determined by the assumption that the degree of compliance to CMS messages is a variable that can be controlled according to well-defined laws of automatic control theory, when in reality it is the result of a more uncertain phenomenon. The proper modeling of drivers' response to CMS and the exact determination of the information that will lead to an equilibrium solution is a complex problem in itself and probably beyond the scope of the IN-TUC strategy, and of the research described in this dissertation. However, even if in the scientific literature, quantitative results about CMS message compliance vary considerably, several variables have been identified as affecting driver response and route change. Thus, a more flexible and realistic approach to the problem of selecting the appropriate set of CMS messages may be one that includes those variables, such as expected delay, travel times, and type of congestion, directly in the

selection process (i.e., one that considers such variables as arguments of the transformation function).

2.5 Coordinated Traffic Management in Munich

The Greater Area of Munich (Germany) is in the process of adopting a coordinated approach towards urban and interurban traffic management. Some of the basic components were developed within the Munich-COMFORT European project (COoperative Management FOr urban and Regional Transport (Csallner & Schlichter 1995)). COMFORT was aimed at developing and implementing transportation management strategies for an improvement of transport and environmental quality, as well as traffic safety and efficiency in the Munich Greater Area. Under the umbrella of the TABASCO project, integrated strategies for the coordination of various traffic management subsystems are being developed and evaluated (Tsavachidis et al. 1998, Hoops & Tsavachidis 1997) and prepared for the operational phase of system integration and implementation. Current research interests lie in the analysis and quantification of the potential gains of integrated traffic management operations. Research in the Munich test site in particular intends to assess the potential improvement of traffic conditions on motorway access to the city by implementing real-time, adaptive traffic management strategies through the use of CMS and signal control, taking advantage, when possible, of spare capacities on the arterial urban network surrounding the motorway. To this end, several traffic modeling and control tools are being blended into an integrated architecture that embodies different spatial and functional levels of control, but respects the administrative authority of each component subsystem.

2.5.1 System Architecture

The system architecture can be represented as a hierarchy of two control layers that are differentiated according to their scope in the decision-making process. The *control framework* comprises political and institutional decisions concerning economic, environmental, and safety issues that are translated into target functions such as travel time and fuel consumption. The relative importance of the resulting strategies is determined by associating varying weights to the objective functions by the local authorities, that take into account the occurrence of particular conditions, such as environmental hazard, special events, or major incidents. The occurrence of such event determines the selection of a prespecified set of

measures, called plans of actions. The implementation of a plan of action is carried out by the subsystems that compose the *control kernel*. Such subsystems operate within different but overlapping spatial layers after the recognition of the potential improvement achievable by instituting an integrated mode of operation.

One of the main elements of this cooperation effort lies in the integration of previously developed management algorithms into a combined framework for collective route guidance based on network condition assessment and dynamic traffic routing that comprises both the motorway and the urban arterial networks and complies with both urban and interurban policies. The two main components of this system are an algorithm to dynamically determine alternative routes based on a dynamic OD estimator (VARIA) (Sachse 1995), and an algorithm for assessing congestion levels and spare capacity on arterials (AIDA) (Hangleiter 1995).

The process of assessing the need and the potential short-term benefits of motorway traffic diversion in VARIA differs from other congestion management strategies in that it attempts to anticipate the onset of congestion by looking at current traffic flows and historical OD-matrices, and performs a short-term prediction of near-future network load. The analysis of current network flows and the estimation of near-future conditions is based on a dynamic OD-matrix estimation module that determines traffic flows based on time-series of traffic counts. The METANET macroscopic traffic simulator (Messner & Papageorgiou 1990) is used to assess the impact of control decisions.

The second component of the integrated system is an algorithm that analyzes traffic volumes on the urban arterials that constitute potential diversion routes. It detects the presence or absence of congestion and, based on the current spare capacity, determines whether additional diverted traffic can be efficiently managed.

The functional flow-chart of the integrated system is described as follows:

1. Whenever the onset of congestion is detected on the arterial network, the arterial component AIDA notifies the freeway module VARIA. This prevents the freeway module from considering the possibility of diversion to the arterial network. Periodically (every five minutes), data describing current travel times on the arterial subnetwork are transmitted to the diversion module.
2. VARIA analyzes current traffic conditions on the motorway network and performs a qualitative prediction of near-future traffic conditions. The traffic state is categorized

as “free-flow”, “overloaded”, or “congested”. The prediction of congestion sets off the analysis of a diversion process. Given the current capacity and expected demand levels, a “desired” diversion volume Δq is computed.

3. AIDA receives notification of the potential additional volume being diverted from the motorway to the arterial alternative route and performs a short-term prediction of the total traffic volume.
4. Given the expected volume/capacity ratios, travel time on the original and alternative path is estimated. If the assessment of travel times to be expected from the diversion indicates a global improvement, appropriate CMS messages are selected to inform motorway drivers on the availability of the alternative route to their destination.

The evaluation of the adopted strategies is not yet complete. Current research includes improved methodologies for the assessment of CMS compliance rates. A system for the analysis of network conditions (NEMO) was used in a simulated environment for the determination and assessment of diversion routes (Hoops & Tsavachidis 1997). Heavy demand levels were simulated, resulting in high travel times on the original route. The rerouting strategies (using the combined arterial and freeway subnetworks) determined by NEMO were compared with the one traditionally implemented (that use only alternative freeway sections) and showed a reduction in travel time and travel distance. While the freeway alternative route determined a 10 percent reduction in travel time and 16 percent reduction in traveled distance, the alternative route that used major arterial links showed a saving of 29 percent in travel time and 28 percent in traveled distance.

2.5.2 Critique

For truly integrated transportation management, this architecture requires a signal operation module that dynamically adjusts arterial capacity in response to diversion. In other words, for truly integrated operations the system should take advantage of the knowledge about the expected diversion volume Δq , which could change in a rather sudden and significant way the flows across the arterial subnetwork. Within the current mode of operation, the signal control module does not use such information.

The scope of the system is limited to the diversion of freeway traffic to adjacent arterials. As in some of the other systems described above, cooperation does not result from

a truly “egalitarian” distributed process, composed of decision-making entities at the same level. The relationship among them is rather one of subordination of the agency responsible for the arterial network management to the one responsible for freeway operations.

Chapter 3

The Centralized Basic Component

The two agents that constitute the distributed architecture are based on Traffic Congestion Manager (TCM) (Logi 1995, Molina et al. 1995), a centralized knowledge-based expert system designed to provide real-time operator decision support for traffic congestion management over combined freeway and arterial networks. In the distributed architecture, each agent is a modified instance of TCM, suitably enhanced to provide inter-agent communication, cooperative reasoning and conflict resolution capabilities. Thus, a comprehension of the main characteristics of the single module TCM is necessary for understanding the functions of the distributed architecture.

This chapter describes the main functions of TCM and the process and results of its evaluation, a fundamental and sometimes neglected component of the development process of any new technology. The evaluation process includes the verification of the correctness of the underlying knowledge base and the assessment of the system's performance, through the definition of several simulation-based scenarios. The results demonstrate the ability of TCM to effectively reduce network-wide driver travel time and stop time under simulated congested conditions, through the implementation of real-time traveler information and signal control. It also provides insights into some of the limitations of the TCM approach, that have been addressed in the enhanced model to improve its reasoning capabilities and performance.

In the past decade, several knowledge-based systems have been developed for a variety of transportation applications. Yet, as reported by Spring (1997), only a small percentage of them have been subject to a formal validation process. KBS, as any other piece of software, must undergo some form of evaluation process, but the process of evaluation of KBS differs substantially from that of traditional software because of the nature of the

problems they address, in particular when dealing with real-time applications (O'Keefe, Balci & Smith 1987). Despite several attempts at the specification of guidelines for the evaluation of KBS, standard criteria have not yet been defined. Several reports on the evaluation of KBS in different fields are described in the literature (a comprehensive review is given by O'Keefe & O'Leary (1993)), but very few of them address real-time transportation related domains (Spring 1993).

The assessment of the strengths and weaknesses of the TCM approach was a basic step for the development of the distributed architecture.

3.1 Traffic Congestion Manager

TCM is a knowledge-based decision-support system for Traffic Management Center (TMC) operator support. TCM addresses non-recurring congestion phenomena, by determining timely control measures aimed at ameliorating traffic conditions in congested areas and at avoiding the spreading of congestion across the network. As noted in Chapter 1, congestion in general manifests itself when the current demand is higher than the available capacity. Non-recurring congestion in particular, is normally due to unusual or unexpected events, such as an extraordinarily excessive demand or a sudden capacity reduction due, for instance, to an incident.

TCM is organized as a structured collection of task-solving modules for the analysis of non-recurring congestion, the selection and implementation of control plans that integrate various management and control methodologies available to control center operators, and the monitoring of the implemented control with respect to the congestion dynamics. Control is performed by selecting plans for intersection signals and freeway ramp meters. Traffic advisories are provided by posting messages on a sparse system of CMS. The onset of non-recurring congestion, in TCM terminology, is called a *problem*.

As shown in Figure 3.1, TCM is able to receive real-time traffic data, (volume, occupancy, and speed, when available), from the TMC data management system, and control data, describing the current status of the control devices. It has an interface to receive and process input data from an automatic incident detection (AID) algorithm, and to accept incident data and control acknowledgments from the operator via an interactive, user-friendly GUI. The system is *event-driven* and continuously monitors the network state. At the onset of non-recurring congestion, it is alarmed, either by an external AID algorithm or by input

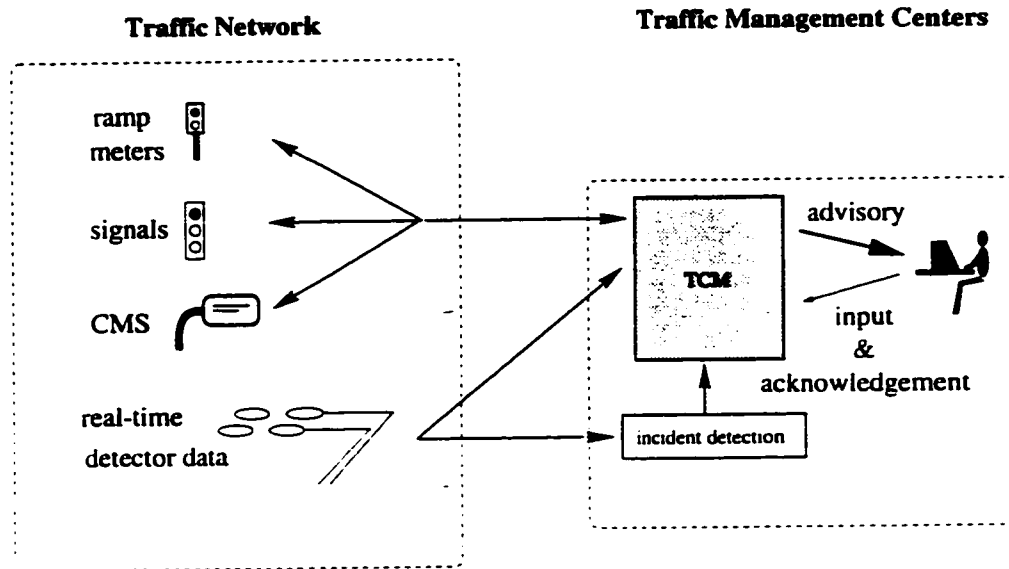


Figure 3.1: A schematic representation of input and output configuration

provided by the operator, and it starts a reasoning process that analyzes the current input data and the current and past network states. TCM then proposes a set of alternative control plans, waiting for acknowledgment from the TMC operator. The operator is able to select or edit the plan considered most effective, based on explanation of the reasoning process and an assessment by TCM of the solution's expected benefits. Using TCM's GUI, the operator is then able to automatically transmit the selected directives to the traffic control system. For each control plan TCM uses heuristic knowledge and procedural methodologies for the delay and congestion duration assessment (D'Ans & Gazis 1976, HCM 1994), and estimates the time required by the control plan to dissipate congestion. This time, called the *recovery time interval*, corresponds to the time during which TCM monitors the evolution of a current problem. During the recovery time, unexpected events may arise that are caused by or may affect the current control (such as the occurrence of further incidents, the dissipation of current ones, or unforeseen effects of the implemented control). In this case, the problem is analyzed once again and a revised control plan may be submitted to the attention of the operator.

3.1.1 The Knowledge Organization

The modular structure of the knowledge organization provides different levels of abstraction, making the knowledge embedded in the system easily accessible, understandable and adaptable. Each knowledge module, or *knowledge unit*, in accordance with a knowledge representation methodology proposed by Chandrasekaran (1983b), is a complex task-solving entity which, using its own *knowledge representation* and *inference* techniques (algorithmic procedures, knowledge-based reasoning processes, or a combination of both) proposes a set of answers to the tasks it is designed to solve. The organization of knowledge units is a hierarchical one, as shown in Figure 3.2, where each level in the graph corresponds to a different level of abstraction and decomposition of the problem solving process.

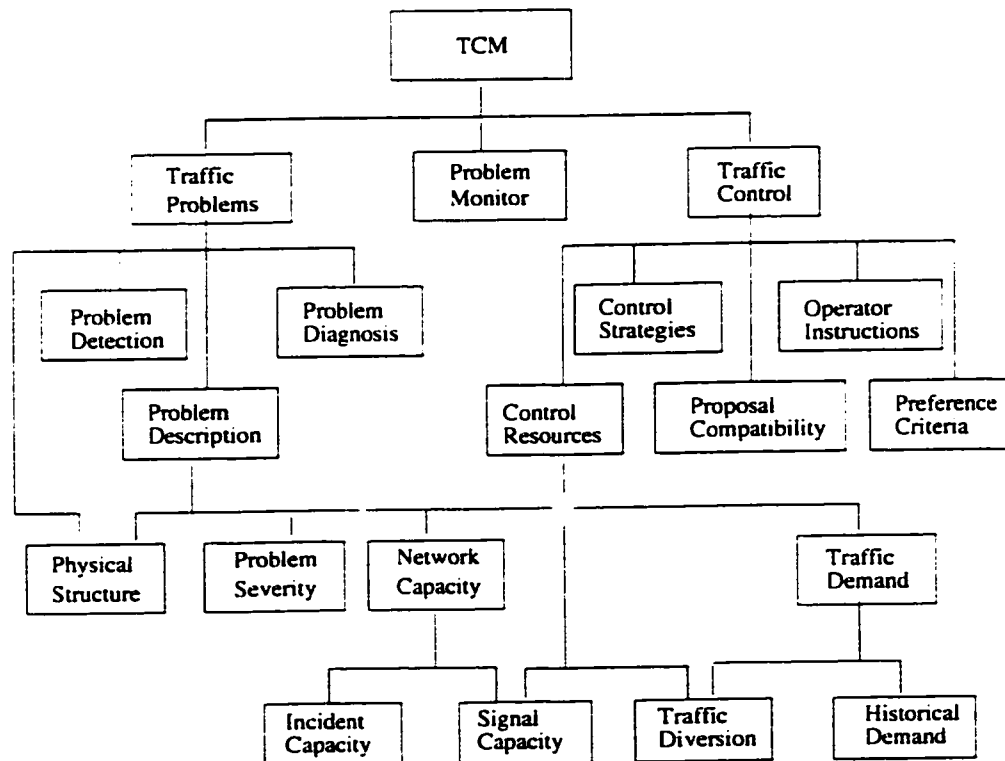


Figure 3.2: The structured collection of knowledge units

According to this structure, TCM is composed of three high-level knowledge modules, two of which are, in turn, composed of lower level units. One module, *Traffic Problems*, is responsible for the analysis of real-time traffic data, the classification of congestion phenomena and the synthesis of related information such as the congestion type, its source, and

the impact it is expected to have on traffic. Another module, *Traffic Control*, determines appropriate control strategies aimed at minimizing the impacts of congestion. A third module, *Problem Monitor*, is devoted to the task of monitoring the dynamics of congestion phenomena for which a control plan has already been selected, but not yet terminated. Problem Monitor observes the effect of the implemented control, and analyzes the causal relationships between multiple congestion events, notifying the operator if the current control becomes counterproductive or obsolete.

3.1.2 Problem Characterization

TCM continuously collects and analyzes input traffic data and, upon receiving notification of the onset of congestion, starts an inference process for the analysis of the problem and the search of appropriate solutions. Input data are received at intervals of 30 seconds (the *polling interval*). Notification can be provided both by an external AID algorithm and by the operator via the system's GUI. TCM is configured to receive incident-related information at every polling interval expressed as a binary representation of the estimated state (congested, non congested) of links that are equipped with system detectors. Applicable algorithms include, for instance, Artificial Neural Networks applications, such as those described by Khan (1995) for arterials and Abdulhai (1997) for freeway networks. The AID algorithm actually is not part of TCM. The reception of manual or automatic incident-related inputs starts the system's data elaboration process.

The Problem Characterization subprocess aims to develop an abstract, high-level representation of the traffic problems and a path-based estimation of the current demand distribution. In order to determine opportune control strategies, the source of the problem, expressed in terms of its location (*critical section*), and its type, must be identified. The critical section is the link or the intersection lane movement at which the current demand exceeds the current capacity. Depending on the problem type, it does not always correspond to the link where congestion is observed: for example, depending on the location of system detectors and on the network topology, congestion may be observed upstream of a freeway on-ramp, while in reality it is due to the merging of the two streams of traffic at the ramp itself. The type of the problem (e.g., an accident, insufficient capacity at a congested signalized intersection, or a bottleneck at a freeway on-ramp) determines the response strategies that will be selected. For instance, when congestion is observed upstream

of a signalized intersection, if it is due to excess demand (e.g. during a special event), a possible strategy involves augmenting the capacity of the corresponding approach through the selection of a different signal plan. If, instead, congestion is caused by the occurrence of a lane-blocking accident upstream of the intersection, increasing the approach capacity is not a viable solution, since the bottleneck is not due to inefficient signalization.

Problem Diagnosis

TCM uses input real-time data and knowledge-based information to infer possible explanations for the observed data, when such an explanation is not provided by the operator. If the reasoning process is not able to identify a unique explanation (i.e., a single critical section and problem type), additional information is requested from the operator, under the assumption that the operator can gather this information, using additional data sources, such as closed circuit television (CCTV) or incident reports from field personnel.

The identification of a problem's critical section and type is achieved through a partial symbolic pattern-matching process (Shapiro 1987) that maps the current network condition with previously defined templates describing typical traffic problems (patterns). The goal of this procedure is to recognize when the traffic condition at a certain location matches a more general situation which corresponds to a certain type of problem, and to infer additional information, such as the cause of the problem and its location. A hierarchical database of knowledge packets, or *frames* (Minsky 1974, Winston 1975), called *problem frames* is constructed. A frame is an information storage unit that allows one to organize static and dynamic knowledge (concepts and data) in such a way to provide easy information retrieval and manipulation. Each problem frame is defined by attributes (or *slots* in the frame terminology) that contain a qualitative description of the expected traffic parameters (e.g., *high upstream occupancy, low downstream volume, operator notification = congested*) associated with the occurrence of a specific type of problem on a generic network component, such as an intersection approach, a freeway junction, or a link.

The network is partitioned into objects whose structure corresponds to the problem templates. Thus, *network objects* are defined to represent intersection approaches, freeway junctions, and links. The difference between a network object *o* and a problem frame *p* is that the information stored in *o* is a dynamic qualitative representation (after opportune transformation) of the *current* traffic data associated with the part of the net-

work corresponding to o , while p contains *static* data against which the dynamic data has to be matched.

Each problem frame p points to a *diagnostic frame* that contains a generic description of the critical section and type of the problem template represented by p (e.g., critical section: DOWNSTREAM LINK; problem type: ACCIDENT).

Within this *frame-driven recognition* process (Charniak & McDermott 1985), if a network object o is successfully mapped to a problem frame p , then the critical section and the type of the problem associated with o are inferred from the corresponding slots of the diagnostic frame. For example, if the diagnostic frame d contains the following information

$$crit_sect(d) = \text{DOWNSTREAM LINK}$$

and d is mapped to network object o , then

$$crit_sect(o) = downstream_link(o) = \text{link name}$$

where $downstream_link(o)$ is a function that returns the downstream link of the network object o .

The recognition of the problem frame that matches a given network object involves accessing the correct high level structure from the available information and is normally the most complex aspect of the reasoning process. The extrapolation of the missing information (often called *script application*), once the recognition is achieved, is a relatively simple task. The frame recognition process (or matching) is described below.

Frame Recognition

The slots of problem frames and network objects contain information related to detectors and links. For instance, as shown in Figure 3.3, a frame describing a freeway on-ramp has slots for detectors and links upstream, downstream, and on the ramp. Each detector has slots for volume, occupancy, speed, and the result of the incident detection algorithm at the detector station. Each link contains a slot for the congestion notification coming from the operator.

Detector and link data for network objects may not always be defined (i.e., the corresponding slots may not have a value). For instance a detector may be missing, it may not provide speed data, or information from the operator may not be available. The

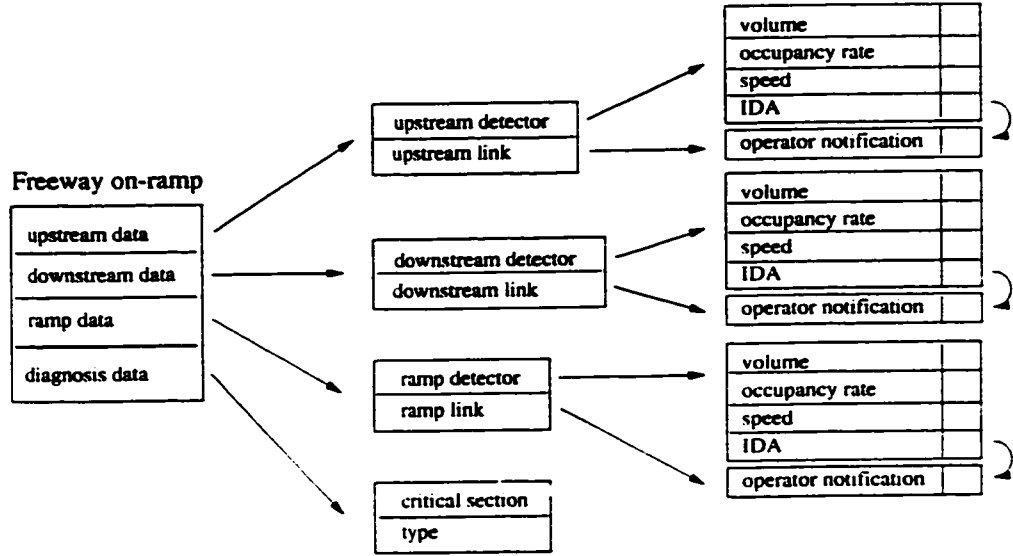


Figure 3.3: A subset of the frame system for problem diagnosis

attributes also have different levels of importance within the matching process: for instance, information from the operator about the occurrence of an incident may override input data from the AID algorithm. In order to deal with the problem of partial or inconsistent information, and represent priority relationships among attributes, a priority scheme is defined, whereby each slot of a problem frame contains a pointer to another slot (the *next* slot) of higher priority. If the operator is aware of congestion on some link, the operator's input overrides the AID algorithm. On the contrary, if no input has been received by the operator, the matching process is based on the result of the AID algorithm.

Given a network object o ($o \in NO$) and a problem frame p ($p \in PF$), let $o.s$ and $p.s$ be, respectively, the slots of o and p , named by s . The value of slot $p.s$ is defined by $val(f.s)$, and its pointer to the *next* slot is given by $next(s, p)$, so that $o.(next(s, p))$ and $p.(next(s, p))$ represent, respectively, the slots of o and p pointed to by $next(s, p)$. The result M of the match between o and p is a boolean value defined through a recursive formulation, as follows:

$$M = (IM(o.s, p.s) \neq \text{FAIL}, \forall s \text{ slot of } o)$$

where

$$IM(o.s, p.s) = \begin{cases} IM(o.(next(s, p)), p.(next(s, p))) & \text{if } next(s, p) \neq \emptyset \wedge \\ & IM(o.(next(s, p)), p.(next(s, p))) \neq \\ & \text{UNKNOWN} \\ DM(o.s, p.s) & \text{otherwise} \end{cases}$$

and

$$DM(o.s, p.s) = \begin{cases} \text{SUCCESS} & \text{if } val(o.s) = val(p.s) \\ \text{FAIL} & \text{if } val(o.s) \neq val(p.s) \\ \text{UNKNOWN} & \text{if } val(o.s) = \emptyset \end{cases}$$

In other words, the match between o and p is true if no slot match (internal match IM) returns *FAIL*. The internal match between slots $o.s$ and $p.s$ returns the result of a recursive call on the higher priority slot, if $next(s, p)$ is defined and the result of the recursive call is known. If the recursive call returns an unknown result or $p.s$ has a null $next$ pointer ($next(s, p) = \emptyset$), the match returns the result of a direct match (DM) between $o.s$ and $p.s$. A direct match is *UNKNOWN* if a slot has no value.

A frame system is used, instead of the more elegant propositions of predicate logic, because although frames are large units of knowledge, they are modular enough to be used as elements in a flexible database (Bender 1996). Furthermore, the deductive process is not a purely logic-based one because of its non-monotonic characteristics (Minsky 1974) whereby the availability of additional input information may change the result of the recognition process. In the classical logic approach, if input data D_1 and D_2 determine conclusions C_1 and C_2 , then the presence of additional input D_3 does not contradict the reached conclusions. In a non-monotonic system (such as virtually any real-time continuous process), the provision of additional information may result in a different conclusion. For instance, information coming from the operator, inconsistent with automatic data (e.g., the result of the AID algorithm) may change the result of the matching process.

Problem Description

Once a problem has been identified, its severity, in terms of its impact on traffic flow, is assessed. This is achieved by estimating the current capacity at the congested location and the expected demand on the network, and computing a measure of the local capacity-demand imbalance and of the expected queue clearance time, which is used to

determine the extent of the required control operations. The network demand is expressed through a time-varying, path-based description of the expected traffic assignment, based on historical information. A path-based formulation is used because, in order to determine suitable traveler information and diversion advice, knowledge of the origins and destinations of the stream of traffic crossing the congested location is required. This permits the selection of opportune CMS messages both upstream of the congested areas and along the alternative routes chosen for diversion.

The result of the problem characterization phase is a *problem_state* object, defined by a *problem_description* (i.e., a list of the current problems that need to be addressed) and a *path_demand* (i.e., a list of pairs composed by a path and its current expected demand).

$$\begin{aligned} \text{problem_state} &= \langle \text{problem_description}, \text{path_demand} \rangle \\ \text{problem_description} &= \{ \langle \text{problem} \rangle \} \\ \text{path_demand} &= \{ \langle \text{path}, \text{demand} \rangle \} \end{aligned}$$

Problems are instances of subclasses of the *problem* class, as shown in Figure 3.4. They all contain information about the location of the critical section, its capacity and its estimated demand. Furthermore, depending on the problem type, which defines the class to which a problem belongs, they have slots for additional information, such as the current queue length at intersection approaches and the estimated duration of an incident.

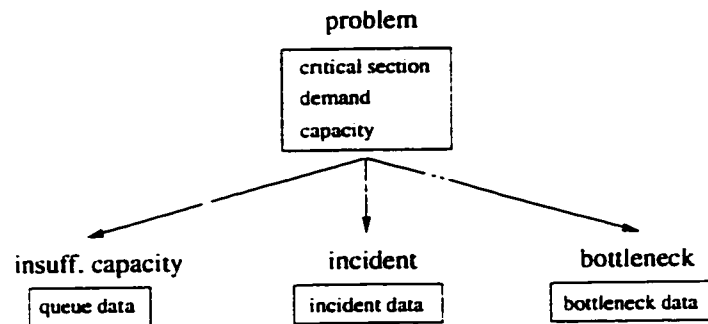


Figure 3.4: The class hierarchy of problems

3.1.3 Problem Response

Once the current problems and their estimated impact have been analyzed, TCM starts a search process for appropriate control settings. A heuristic algorithm is used, called

TPS (*Traffic Problem Solver*; the name derives from the *General Problem Solver*, the well-known algorithm developed by Newell & Simon (1963) for the decomposition of problems into subgoals and treatable subproblems).

TPS fall under the category of *planning* (Shapiro 1987). In computer-based problem solving, planning is defined as the generation of a sequence of actions that can change the environment, focusing on methods for decomposing the original problem into appropriate subparts, and recording and handling interactions among the subparts as they are detected during the problem-solving process (Rich & Knight 1991).

Key issues that need to be addressed when developing a planning system are the search control, the structure of goals and subgoals, and the representation of actions and the prediction of their effects.

Search Control

TPS uses the *Best-First Search* (BFS) algorithm (Doran & Michie 1966). In this case a heterogeneous search tree is created starting from an initial node (that corresponds to a given initial state), and nodes are visited sequentially (i.e., expanded into other sets of nodes through the application of expansion rules, until a termination condition is verified). In a BFS, the node to be visited next is the “best” node, according to certain criteria of “goodness”. Each node of the tree represents a problem state, as defined in Section 3.1.2, and each arc describes a relation between states and corresponds either to a possible subgoal (and the corresponding strategy) or to an action selected to process a subgoal that determines the creation of a new problem state. The root corresponds to the initial state, and it describes the original problems for which solutions must be determined. A path from the root to a leaf node represents a possible solution whose corresponding actions are determined by the arcs composing the path. The evaluation of a node, on which the selection criteria are based, is performed through the evaluation of a heuristic function, which is a measure of the estimated “desirability” of the corresponding solution. The heuristic function is an approximation of an objective function, in conventional mathematical optimization.

The search ends when a goal state has been reached. Given the lack of an analytical formulation of the objective function, in TPS there is no *a priori* goal state, that is, there is no computable way to determine whether a solution is the optimal one. In such cases, especially in real-time applications, it is customary to end the search either when no

reasonable alternative state can be reached (for example, all possible "reasonable" solutions have been analyzed), or when predefined thresholds (on the computation time, the number of solutions, the depth of the tree, etc.) have been reached (Rich & Knight 1991).

Goal Structure

The technique used for selecting appropriate actions during the problem-solving process, is based on the concepts of *means-ends analysis* (Ernst & Newell 1969). The search for a solution involves iteratively detecting the difference between the current and the desired state, specifying subgoals that aim to reduce that difference, and searching for actions that can achieve those subgoals. The estimated (or predicted) effect of an action's application will lead to a state that is likely to be closer to the objective, thus reducing the original problem. For the currently visited node, a set of subgoals and corresponding solution strategies is constructed. In TPS, strategies are parametric objects, instantiated from a set of predefined classes, and are dynamically created through the assignment of values to particular attributes. Strategies prescribe different, complementary or alternative ways to achieve corresponding subgoals. Typical strategies (and the corresponding subgoals) include *increase capacity* at a congested location by *signalization*, *decrease flow* at a congested location by *traffic diversion*, and *decrease flow* at a congested location by *upstream metering*.

Actions and Their Effect

For each strategy, the algorithm looks for suitable control device settings that are expected to achieve, partially or completely, the corresponding goal. In general, a strategy may require the modification of the status of more than one control device (e.g., a group of CMS). A group of control settings determined in one iteration step as part of the same algorithm is called a *control action*. Depending on the type of strategy, alternative control actions may be constructed that achieve the subgoal corresponding to the strategy. In the current version, timing plans and CMS messages are selected from a predefined database, based on knowledge elicited from engineers at the local traffic management center.

In order to expedite the analysis of applicable actions, for each action a *precondition* field is defined, that contains a list of predictions, or conditions, that the system is able to check and that must be verified for the action to be applicable (Fikes & Nilsson 1971).

They define the state of the network (the status of a link, of a path or a set of paths, etc.), the level of demand, the time of day, the presence of special events, etc.

In means-ends analysis, solutions result from the aggregation of actions selected to achieve a certain subgoal, for the solution of a subpart of the original problem. One way to keep track of the expected effect of an action is to describe the changes it is expected to have on the current state. When searching for actions while attempting to achieve a subgoal, the action's effect is matched against the subgoal. If the action is found to be relevant for that subgoal, then its effect on other subgoals can be analyzed (Rich & Knight 1991).

TCM stores information about actions (their preconditions and their effect) in frames. The selection of a control action involves the verification of its preconditions, by matching them with the current state within the problem-solving process, and the prediction of the effect of its application to the current state. Frames are used, instead of more common inference methodologies, such as production-rules (see for instance the incident management module within the Santa Monica Smart Corridor Expert System (Karimi et al. 1992)), because they are powerful tools for efficiently dealing with semantic knowledge (Rich & Knight 1991). The association of the preconditions and effect to a control action, for example, allows the system to determine not only when that action is applicable, but also *why* it should or should not be applied (depending, for instance, on the synergism of its effect with the global problem-solving process). Frame systems can be converted into rule systems (Bender 1996), but often at the price of loss in efficiency and knowledge representation power.

Identification of Dead-Ends

The search process must be able to detect incompatible combinations of control actions. Incompatibility may be due to several reasons. The simplest and easiest to detect is physical incompatibility between two different settings for the same control device (e.g., two actions are incompatible if they recommend different plans for the same signal). Another type of incompatibility may arise from the combination of settings for different control devices. For instance, when proposing a recommendation for an alternative route, all CMS messages along the alternative route must be semantically compatible (i.e., they can not provide conflicting information). A third potential source of incompatibility relates to the compatibility between control actions and problems. If a control action is selected for one

problem or a subgoal, it should not, in general, worsen other problems or work against other subgoals.

The verification of mutual consistency between solutions involves mostly prespecified knowledge-based compatibility criteria and simple constraint-satisfaction techniques. Compatibility criteria are determined, for instance, by the requirement that no action can decrease the capacity or increase the flow through a critical section (i.e., no signal plan should be selected that reduces the capacity of a congested approach, and no CMS message should be selected that diverts traffic towards a congested segment). Semantic compatibility within control actions (e.g., the compatibility of a group of CMS messages) is guaranteed by the specification of control action frames themselves: control actions composed of settings for more than one device are complete and indivisible. For instance, since CMS messages are selected from a pre-stored database, control actions for CMS are organized in predefined, indivisible groups of mutually compatible CMS messages.

The Search Algorithm

The input to the algorithm is the initial problem state PS_0 , resulting from the Problem Characterization phase. The algorithm returns an ordered list of solutions. Each solution and each internal tree node are instances of the class *node* defined as follows:

$$node = \langle prob_state, contr_acts, heur_funct, del_red \rangle$$

where

prob_state is a problem state;

contr_acts is the list of control actions corresponding to the node;

heur is the value of the heuristic function associated with the node. H_0 is the function's value at the start node;

del_red is the expected (estimated) delay reduction corresponding to the implementation of a list of control actions, with respect to the initial problem state.

The current control state, i.e., the current setting for CMS, signals, and ramp meters, as well as the current network state (the set of the currently measured traffic variables on the network) are also available to the algorithm as global variables. The *visit* function visits a node, setting the *visited* attribute of a node to *TRUE*.

The output is a list of solution nodes, ordered according to the value of the expected delay, defined as:

$$SOL = \{N \text{ is a node} : N_i.heur \geq N_j.heur \iff i < j\}$$

The TPS algorithm is defined as follows:

Algorithm 3.1 (TPS)

```

1 begin
2    $SOL = \emptyset$ ;
3    $N_0 = \langle PS_0, \{\}, H_0, 0 \rangle$ ;
4    $TREE = \{N_0\}$ ;
5   while  $\langle \exists N \in TREE, \neg visited(N) \rangle$  do
6      $N = select\_best\_node(TREE)$ ;
7      $visited(N) = TRUE$ ;
8      $PARTIAL\_SOL = expand\_node(N)$ ;
9     if  $PARTIAL\_SOL = \emptyset$ 
10      then  $push(N, SOL)$ 
11      else for  $M \in PARTIAL\_SOL$  do
12         $push(M, TREE)$ ;
13      od;
14    fi
15  od
16  for  $N \in SOL$  do
17     $compute\_delay\_reduction(N)$ ;
18  od;
19   $sort(SOL, del\_red)$ ;
20  return  $(SOL)$ ;
21 end

```

The *expand_node* procedure is as follows:

Algorithm 3.2 (Expand Node)

```

1 begin
2    $SUBTREE = \emptyset$ ;
3    $STRATEGIES = select\_strategies(N)$ ;
4   for  $S \in STRATEGIES$  do
5      $CA = determine\_control\_action(N, S)$ ;
6     while  $\langle CA \neq \emptyset \rangle$  do
7        $COMPATIBLE = verify\_compatibility(CA, N)$ ;
8       if  $COMPATIBLE$ 
9         then

```

```

10             M = create_new_node(CA,N)
12             push(M, SUBTREE);
14         else M =  $\emptyset$ ;
16     fi;
17     CA = determine_control_action(N,S);
18 od;
19 od;
20 return (SUBTREE);
22 end

```

TPS has the following characteristics:

- It generates heterogeneous but integrated solution plans composed of settings for different types of control devices (signals, CMS, ramp meters).
- It explicitly represents the subgoals and the control settings to be applied, justifying the steps of the reasoning process that led to their choices.

An example of the search tree developed by TPS is shown in Figure 3.5.

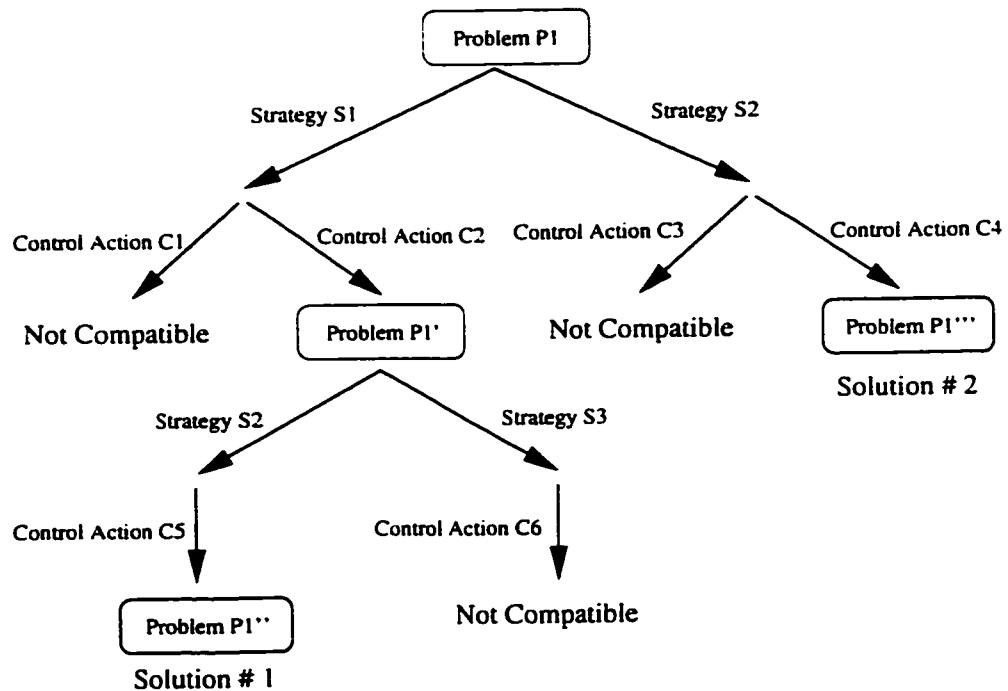


Figure 3.5: The search tree in the problem-solving process

It is not a simple task to estimate the complexity of the algorithm. In general, if at any time, any one of n control actions can be selected for a problem state, the total

number of possible solutions is given by $\sum_{i=0}^n \binom{n}{i}$, which would make the complexity of the algorithm highly exponential if heuristics were not used to reduce the number of feasible combinations of control actions.

3.1.4 Problem Monitor

The *Problem Monitor* KU has the task of monitoring the evolution of those congestion phenomena that have already been addressed, i.e., for which a control action has been selected and implemented, but that have not yet dissipated. Problems for which a control solution has been implemented take some time to dissipate. TCM estimates this time (defined in Section 3.1 as the recovery time interval), using several sources of information, such as input from the operator, (the estimated time to restore full capacity at an incident site or the queue length at an intersection), and algorithmic knowledge from the queueing theory.

TCM builds a dynamic database of problem objects that stores the response plan associated with each analyzed problem, the conditions that characterize its existence (*presence conditions*), and conditions that, if satisfied, require a new analysis of the problem, called *context conditions* (e.g., the onset of congestion on diversion paths). Before a problem is stored in the database, the operator is asked to acknowledge or edit the recovery time. During the recovery time interval, TCM monitors traffic dynamics, verifying the presence and context conditions, and submitting the problem to the operator for further analysis if these are not satisfied. At the end of the recovery time, the operator is notified, and if congestion has indeed cleared a new assessment is performed to adjust control plans accordingly.

This function performs a sort of feedback on the implemented control. Also, it renders the system able to analyze causal relationships among congestion events, and devise control solutions that take those relationships into account. Furthermore, by helping the system recognize input data describing problems that have already been analyzed, it avoids continuous repetitions of the same analysis, alleviating the system's burden and relieving the operator from repetitive interaction with the machine.

3.1.5 The Test Site

TCM was implemented for a subnetwork of the city of Anaheim, in southern California, as shown in Figure 3.6. The network includes a 4-km section of a major freeway and

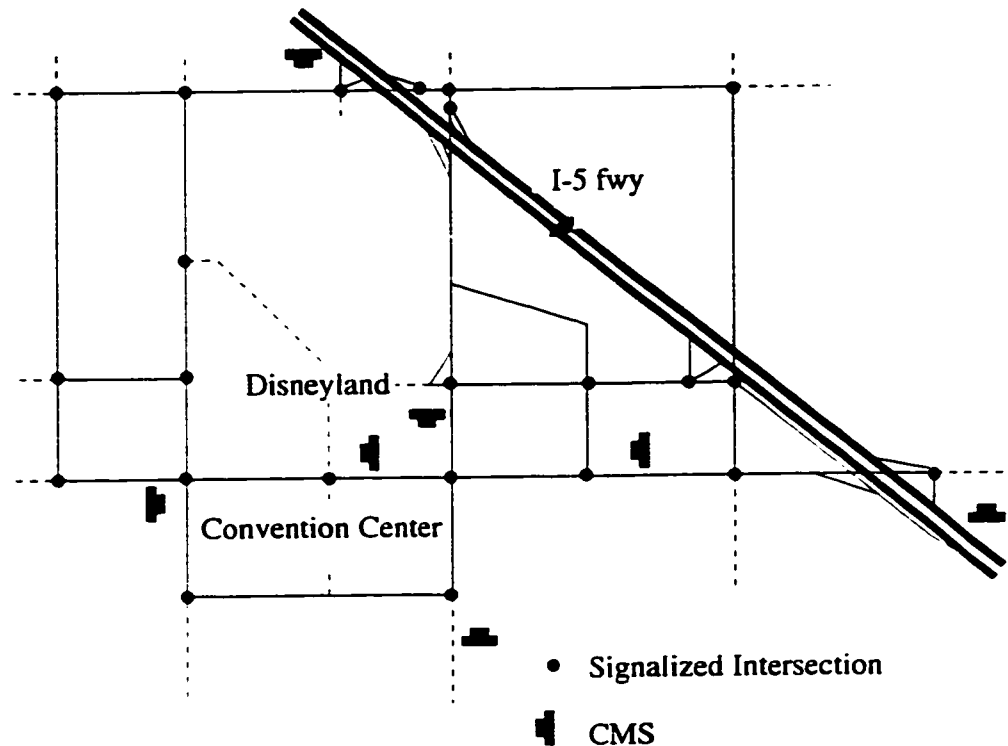


Figure 3.6: The test site in Anaheim, California

a 20-km² network of multi-lane surface streets, with 23 signalized intersections. Anaheim and its direct neighbors account for about 50% of the Orange County population. The area attracts a disproportionate amount of trip ends due to the intensity of special attractors that include Disneyland (about 15 million visitors per year) and a busy Convention Center and hotel industry which increase the total number of visitors to over 20 million per year (Planning Consultants Research 1996). The Anaheim TMC employs Urban Traffic Control System (UTCS) enhanced real-time software (FHWA 1987), with input from all controlled intersections, through mid-block detectors, and a CCTV system that allows the operator to monitor the flow of traffic and to detect or verify the occurrence of congestion. The Anaheim TMC also has control over several CMS, which are used mainly to direct travelers to Disneyland, the Convention Center and the Interstate-5 Freeway.

3.2 The Evaluation Process

Evaluating software means assessing its functionality, i.e. determining what it does and how well it does it, and possibly estimating its performance according to standard criteria. For knowledge-based systems, standard criteria have not been defined, and scientists still disagree on the most appropriate methodologies to evaluate knowledge-based systems (for a detailed review see O'Keefe & O'Leary (1993)). Most reports, however, acknowledge the importance of two aspects in the evaluation process: verification and validation (V&V). Verification involves assessing the system's adherence to the specifications and normally requires demonstrating the consistency and completeness of the knowledge base and the correctness of the inference process. Validation is concerned with assessing the quality of the solutions, thus focusing on the system's effectiveness. Important aspects of a system's evaluation are also related to how the system addresses end-users needs, in terms of user acceptance and system's usefulness in the field.

Verification is essentially a programming task, and sometimes is performed using traditional software engineering methodologies (Gupta 1990) that focus on demonstrating the logical correctness (logical verification) of the underlying knowledge base (Lee & O'Keefe 1994).

Validation involves testing KBS to ascertain whether they achieve acceptable performance levels: this has been primarily based on *ad hoc* techniques. In recent years it has been recognized as the cornerstone of the evaluation process (O'Keefe et al. 1987).

This section reports on the work performed in conjunction with and following a Federal Highway Administration (FHWA)-sponsored project for the development of guidelines for the verification, validation, and evaluation of expert systems (Wentworth, Knaus & Aougab 1995), for which TCM was used as an experimental case study (Logi & Ritchie 1997).

The evaluation does not address issues related to TCM's end-user acceptance and its applicability in the field since TCM, in its current prototypical version, is largely a research tool and its development represents an interim step in the realization of a distributed architecture composed of enhanced TCM agents. Some of TCM's characteristics, including its GUI, are likely to be modified so our interest was with evaluating TCM's problem-solving approach, rather than its immediate applicability in the field.

Verification

The evaluation of TCM involved the specification of the system's scope, through the definition of the problems addressed, the verification of the solution requirements and the correctness of the knowledge base, and the assessment of the system performance. Initially, the equivalence between the design and the specification was verified with the definition of case studies and the application of the system functions to them. Later, the correspondence between the system's design and the software code was verified, for each system component separately, through the definition of several input situations and comparison between the expected results and the function's output. This process was performed in a bottom-up manner by verifying first the correctness of each low-level function and then using the validated functions to verify the correctness of higher-level subsystems.

The logical verification of the knowledge base demonstrated its completeness and consistency. A system is complete if, for all its possible inputs, it produces an output. The verification of the completeness of the knowledge base was performed as a bottom-up process across the knowledge structure depicted in Figure 3.2. It started with the verification of the completeness of the knowledge units at the lowest level and proceeded upwards, level by level, until the completeness of the whole system was verified. A system is consistent if, for all its possible inputs, it produces both conceptually and mathematically consistent outputs. The consistency verification process started with the identification of possible sets of mutually inconsistent outputs and was carried out as a top-down process. If the system was found to yield inconsistent solutions, the process proceeded downward in order to locate the sources of inconsistency, according to the guidelines of Wentworth et al. (1995).

Validation

Several approaches to validation have been proposed. Lee & O'Keefe (1994) review recently developed V&V methodologies classifying them in three main categories: automated tools, that normally address logical verification issues; analytical modeling, mostly concerned with performance validation; and human support, which relies on efforts by developers, users or experts to aid V&V.

Automated tools in general are more suited for addressing logical verification, i.e., searching the knowledge base for gaps in knowledge or errors in the reasoning process. Generally, automated methods make rigorous assumptions about inference and have yet to

prove their applicability for non-rule-based knowledge representation.

Validating KBS using human support requires testing by developers, user or experts on the validity of the solutions proposed by a KBS (Buchanan & Shortliffe 1984, Chandrasekaran 1983a). The Turing Test (Turing 1950), which involves a comparison between the response provided by the system and that of a team of experts to the same set of test scenarios, falls in this category. The Turing Test is one of the most commonly adopted testing methodologies for those cases in which it is possible to clearly define test scenarios, judge the quality of a solution, and in general estimate the significance of the difference between system and expert solutions.

Analytical modeling methods are used to estimate measurable characteristics of a system, such as the logical correctness of a knowledge-base, the speed of an inference process, and the general performance of a system, when compared to some notion of expected or acceptable performance (Hall & Heinze 1989). Simulation model methods, where a KBS controls a simulation model, fall into this category. They involve the execution of multiple simulation runs, with different input configurations (different test scenarios) followed by the measurement of the system performance. Simulation-based validation can be only as good as the simulation model: it is important to remember that even if a system performs well in a simulated environment, it does not necessarily mean that it will perform as well in the real world. Nonetheless, simulation models can be used when real-life case scenarios can not be easily predicted, generated, or analyzed (see Prerau, Papp, Bhatnagar & Weintraub (1993) for an application to plant safety maintenance), or when the validity of a response can not be easily assessed by a direct comparison with that of a panel of experts.

In the case of TCM, the analytical model option was considered the most suitable. Indeed, the system composed of TCM, the TMC operator, and the traffic network can be viewed as an interactive structure evolving in time, with a feedback mechanism provided by the control suggested by TCM. The evolution of traffic after the occurrence of congestion and the implementation of adequate control depends on the control itself. Therefore, rather than testing the system's validity through a set of fixed scenarios and comparing its response with the one provided by a traffic expert, it was judged more interesting to evaluate the performance of the system in continuous applications. Simulated data had to be used, given the impossibility of directly applying the control on-line in a real environment, before a proper evaluation. The validation was carried out by simulating the traffic network for a 2-hour time period, providing TCM with simulated real-time traffic data, implementing

the control directives suggested by TCM, and studying how traffic was affected by them. A quantitative estimation of the impact that the system had on traffic was obtained by measuring the network performance both when TCM was used and when the default UTCS control (without TCM) was applied, and comparing the two. The traffic network performance was assessed through average and network-wide measures of effectiveness, such as the travel time, the stop time and the vehicle-miles driven.

3.2.1 The Simulated Environment

The validation of TCM's effectiveness in providing real-time control response to congestion, involved defining test scenarios consisting of different sources of congestion.

The Simulation Model

The Dynasmart simulation model (Jayakrishnan, Mahmassani & Hu 1994) was selected for the evaluation. Dynasmart is a time-driven mesoscopic simulation model that assigns time-varying traffic demand to a network, based on a user-specified time-varying trip table, and moves individual vehicles according to macroscopic flow relations. Vehicles are assigned a path at the beginning of their journey and they do not change paths unless they receive external information (in this case, in the form of CMS advisories). The equilibrium paths are selected from a set of equilibrium paths that can either be specified using an external assignment model or obtained from a user-equilibrium assignment run in an initialization period during which vehicles are assumed to have information on current traffic conditions.

A network of 75 nodes, 154 links, and 17 zones, surrounding the area controlled by TCM was modeled within Dynasmart. Traffic counts were collected from the Anaheim TMC for 82 one-way links on the major arterials and on 3 freeway ramps, based on different days' tube counts for two-hour periods during the morning peak, when commuter traffic interacts with traffic directed to Disneyland and the Anaheim Convention Center. Freeway hourly volumes were estimated based on data obtained from the California Department of Transportation. The observed traffic data and projections of attendance levels at Disneyland and the Anaheim Convention Center were used as input to the traffic modeling package CONTRAM (Leonard, Gower & Taylor 1989). Three dynamic, time-varying Origin-Destination (OD) matrices were developed, corresponding to different expected levels of demand (low, medium, and high, with the measured data corresponding to medium

traffic conditions). The two-hour test period was divided into eight 15-minute intervals and OD matrices were constructed to provide Dynasmart with time-varying traffic volumes.

Dynasmart simulates network flow at signalized intersections by adjusting the capacity for each movement through the intersection, based on the green time available to that movement as determined by the active phase during each time step. Dynasmart's main potential though lies in its ability of modeling driver response to information about the condition of the network. The default time-of-day signal timing plans implemented by the Anaheim TMC for the morning-peak were coded into TCM and simulated by Dynasmart, and used as the default control against which TCM's advisories were compared.

The original version of Dynasmart was modified to allow for the possibility of injecting user-specified incidents that reduce the link capacity by a given amount for a specified time interval.

Dynasmart does not have lane-changing or car-following models, but this was not considered a significant drawback for the analysis of traffic behavior and the computation of travel times in congested situations. Indeed, the effect of severe traffic congestion and of vehicle re-routing is best captured at a path level rather than at a link level, since variations in stop time and travel time due to vehicular lane changing and interaction are not significant in the computation of route travel times (Jayakrishnan 1992).

The Test Scenarios

Twenty test scenarios, grouped in the ten categories shown in Table 3.1 were simulated that included different sources of congestion with varying characteristics, at various locations and different times, under the three different levels of demand. For each of the three levels of demand, a scenario with no injected incident was simulated (scenarios 1, 5 and 10). In these cases, delay was a result of the formation of queues at signalized intersections. Other scenarios were simulated, where incidents were artificially injected by temporarily reducing the capacity of a link. In scenarios 2, 3, 6, and 9, one incident per scenario was injected. In each case, both the location of the incident (arterial, freeway, and freeway ramp), the percentage capacity reduction (30, 50, 66, 80), and the duration of the reduced capacity (10, 30, and 60 minutes), were varied.

It was also interesting to determine how well TCM recognized whether or not incidents were related and how it responded both to sources of congestion that had measurable effect upon each other and to unrelated events. The test scenarios thus included three

cases (scenarios 4, 7 and 8) in which multiple sources of congestion occurred within a short interval. In scenario 4, a first incident was simulated on the freeway, and a second one was injected along one of the paths that had been selected as an alternative route. In scenario 7, two incidents were injected on arterials, that affected traffic directed to Disneyland. The second incident was simulated along a path that had been selected as an alternative to the first one. In scenario 8, the congestion phenomena were unrelated, and had no measurable effect on each other.

A high compliance rate to CMS messages was assumed in most simulated scenarios. This allowed upper bounds to be established on the potential travel time reduction that can be obtained by integrating real-time traveler information and signal control policies, in response to major incidents (capacity reduction of up to 80% and incident duration of up to 1 hour). This was also partially justified by the absence of exact data on CMS compliance rates in the neighborhood of major tourist attractions, such as Disneyland, where many users are unfamiliar with the network and likely to comply with CMS recommendations. In the literature, measured or estimated CMS compliance rates range between 5 and 90% (Wardman, Bonsall & Shires 1997, Dudek, Weaver, Hatcher & Richards 1978).

The evaluation of TCM's potential to address non-recurring congestion was based on the comparison of the simulated network performance with and without the implementation of the control proposed by TCM. The average travel time and stop time, as measured by Dynasmart at the end of each simulation run, were considered significant measures of effectiveness. The difference in the average traveled distance was not significant, possibly due to the limited size of the network under study, and it is not reported.

Thus, for each test scenario, two simulations were performed, with the same input characteristics. The *before* case involved simulating the default control implemented by the Anaheim TMC, using time-of-day signal plans and no CMS messages. The *after* case involved implementing the control responses provided by TCM. Since no predefined specific incident response plan existed, it was not possible to simulate the response of TMC engineers: the characteristics of such response depend on current network conditions at the time of occurrence of the incident and during the recovery time, and they vary both in the control actions selected and in the duration of the selected control. The simulation of the response of TMC traffic engineers, would thus have required their presence in the laboratory during the simulation runs.

3.2.2 Evaluation Results

Quantitative Results

Table 3.1: Basic module: simulated network performance

#	Demand	Incident Characteristics			CMS Compliance Rate (%)	Average Travel Time (min)			Average Stop Time (min)		
		Location	Length (min)	Capacity Reduction		Baseline	TCM	Δ (%)	Baseline	TCM	Δ (%)
1	low	no incident			75	5.13	4.71	-8.2	1.37	1.02	-25.5
2	low	freeway	30	30	75	5.87	4.87	-17.0	1.69	1.15	-31.9
3	low	freeway	30	80	75	6.02	5.86	-2.7	1.91	1.61	-15.7
4	low	freeway	30	80	75	7.18	7.04	-1.9	2.29	1.95	-14.8
		arterial	10	50							
5	medium	no incident			100	6.45	4.74	-26.5	2.35	1.03	-56.2
6	medium	arterial	60	66	100	6.51	4.78	-26.6	2.20	0.98	-55.4
7	medium	arterial	30	66	100	7.13	5.06	-29.0	2.86	1.26	-55.9
		arterial	10	50							
8	medium	arterial	60	66	50	6.54	5.45	-16.7	2.37	1.41	-40.5
		arterial	10	50							
9	medium	ramp	30	66	75	8.00	6.01	-24.9	2.79	1.33	-52.3
10	high	no incident			75	7.11	6.42	-9.7	2.45	1.96	-19.9

Quantitative results, showing the average (per-vehicle) travel time and stop time reduction compared with baseline control using time-of-day UTCS signal control plans and no CMS, are listed in Table 3.1. The results show, in general, that the implementation of diversion strategies, combined with the adjustment of signal plan control has potential for a substantial reduction in the travel time, between 1.9% and 29.0% and a higher reduction in the average stop time, between 14.8% and 55.9%.

A more extensive set of test cases under a broader range of conditions, is recommended if a more precise assessment of the effectiveness of TCM is required. Nonetheless, these results and the match between the system's expected and actual behavior in the simulated scenarios demonstrates its ability to produce an improvement in the network performance under congested conditions.

For high and low levels of demand, the improvements were not as high as for medium levels, leading us to conclude that the closer the network is to saturation and the smaller is the spare capacity, the harder it is to find measures for balancing demand and capacity. On the other hand, the lower the demand the smaller is the adverse effect of

congestion, and the more marginal is the improvement offered by TCM.

TCM responded well to the simultaneous occurrence of multiple problems. In scenarios 4 and 7, when the second incident occurred, the control initially selected was correctly modified. TCM had to reassess the network conditions and propose an alternative solution that took into consideration both problems. In scenario 8, where two incidents occurred with no measurable effect upon each other, TCM, correctly, did not alter the control initially selected.

Table 3.2: Basic module response time

#	Demand	Incident Characteristics			CMS Compliance Rate (%)	Response Time (sec)
		Location	Length (min)	% Capacity Reduction		
1	low	no incident			75	61.5
2	low	freeway	30	30	75	22.3
3	low	freeway	30	80	75	35.4
4	low	freeway	30	80	75	28.5
		arterial	10	50		
5	medium	no incident			100	58.0
6	medium	arterial	60	66	100	91.5
7	medium	arterial	30	66	100	71.2
		arterial	10	50		
8	medium	arterial	60	66	50	61.4
		arterial	10	50		
9	medium	ramp	30	66	75	47.3
10	high	no incident			75	48.2

Table 3.2 reports on the system response time, i.e., the time that TCM required for the search of a set of feasible solutions (on a SUN Sparc 20 workstation) once a problem had been identified. In the simulated test scenarios, the TPS algorithm always took less than 92 seconds, with an average response time of 52.5 seconds. The response time is a function of both the input data, the knowledge base, (i.e. the number of problems analyzed, their location, etc.), and the magnitude of the plan database. The enhanced version of TCM, within the distributed architecture allows real-time computation of new timing plans based on actuated control, using existing algorithms for signal control (Gazis 1974, Akcelik 1981), rather than searching in a predefined database, see Chapter 6. This feature renders the search independent on the size of the plan database.

System Scalability

As noted in Section 3.1.3, it is difficult to estimate the complexity of the heuristic search algorithm. In general, though, several factors can be observed as responsible for such complexity: the size of the network under analysis is perhaps the most noticeable. Larger or more dense networks are more likely to experience a higher number of concurrent problems. Such networks have a higher number of paths and they have a higher number of control devices.

As part of the development of the distributed architecture, of which a revised version of TCM is the basic module, and partially as a result of the evaluation process, TCM has undergone several design modifications and software optimization, which increased its scalability. The following issues affect the scalability of TCM:

- The size of the database of paths, and the effect that each action has on them.
- The size of the database of network objects.
- The size of the database of predefined signal, meter and CMS plans.
- The number of concurrent problems that may need to be analyzed, and the relation between them.

Software optimization and design modifications of TCM include the restructuring of the heuristic search procedure, the modification of the path-based network demand computation, and the on-line construction of signal and meter plans, rather than selection from pre-specified databases.

Greater advantage can be taken of inheritance properties of the object-oriented computation. For example, a solution can be represented not by the state associated with a node, but by the path in the search tree that leads to that node (i.e., by the set of nodes and arcs from the root to that node), thus node N_{i+1} can contain only information required to represent action CA_{i+1} and its effect, reducing significantly the required number of operations and the amount of data storage. This reduction is particularly noticeable with respect to the assessment of the network demand since each control action, in general, has effect only on a limited number of routes.

In relation to the number of concurrent problems that may occur in larger networks, the observable mutual effect that concurrent problems have on each other may be

smaller because the problems are more physically apart.

Nonetheless, if the size of the network increases to such a point that the construction of the search tree or the analysis of all network objects becomes too cumbersome, two approaches can be taken to overcome the problem. The first one involves employing a coarser representation of the network by, for instance, considering only paths with large volumes, searching control plans for major intersections only, and considering incidents only on major arterials. The other, which is by no means a trivial task but is facilitated by the modularity of the architecture, involves extending the modular knowledge structure of the system to a higher level: TCM becomes an agent (a complex knowledge unit) within a hierarchy of interacting agents, each of which monitors and controls a portion of the global network. Such an approach, consistent with the work of Chandrasekaran (1983b) and, more recently for traffic-related applications, of Cuenca et al. (1995), is the one taken for the development of the distributed architecture.

Weaknesses

The evaluation procedure also identified some weaknesses in the TCM approach, regarding the way the system estimates the network demand and computes the expected effect of control actions. TCM computes the network demand using historical knowledge of the local traffic patterns under different conditions, and updates it according to the expected effect of the currently applied control. This allows a fast assessment of the current demand, but in some cases such assessment is not accurate enough. The network demand can not be solely based on measured link volumes because, in order to assess the effect of CMS messages, path-related information is necessary. Also, under heavily congested conditions, the measured volume is not representative of the demand, but becomes a closer measure of the available capacity (McShane & Roess 1990). An alternative to the current approach may be provided by the execution of a dynamic network assignment model to extrapolate path-related information such as the current or near-future travel times across the network.

Another issue that could be addressed relates to the way TCM assesses the expected benefits of control actions and evaluates alternative proposals accordingly. Currently, each control action is associated with its expected effect on a problem, and a combination of the effect of all the control actions that compose a response plan defines the benefit expected by a solution. This process provides a fast and efficient way to discard a large number of

control actions, and allows the system to select only a few meaningful response plans. A more realistic comparison of the effect of alternative control plans could be obtained by first using TCM's heuristic search to identify a small number of feasible solutions, and then executing a fast simulation of the future network performance in the area affected by the implemented control.

The combination of these two limitations prevents the system from being able to anticipate future congestion that may be caused by a sub-optimal control decision, and may lead to the onset of secondary congestion. TCM has a mechanism that checks the feasibility of a certain control decision, and its compatibility with the current state of the network. This avoids, for example, diversion of traffic towards a congested location, or decrease in the signal phase length corresponding to a congested approach, but it does not suggest how to optimize the control along the route traffic is diverted to, in order to prevent the potential onset of additional congestion. This limitation has been addressed in the distributed architecture, as described in Chapter 5.

Chapter 4

Distributed Problem Solving and Functionally Accurate Cooperation

This chapter describes the main concepts and the key research findings in a relatively recent and still evolving approach to distributed problem solving, called *Functionally Accurate, Cooperative* (FA/C) problem solving. FA/C research analyzes methodologies for cooperation among agents in a distributed environment. Agents formulate tentative, partial solutions and attempt to resolve the resulting uncertainties by applying several conflict resolution techniques. The main goal of this approach is to provide interacting agents with the means for effective cooperative reasoning that do not require each agent to have complete and up-to-date information about the problem-solving activity of other agents. The modeling of the problem of network-wide, multi-jurisdictional traffic management as a distributed problem solving process and the application of the FA/C approach thus provide a solution mechanism that limits the computation time for the formulation of control solutions and maintains a distributed organization of the data and information involved.

The main concepts and the most important results of this approach are reported here, because they inspired the development of cooperation mechanisms used by the distributed architecture for congestion management described in this dissertation.

4.1 Distributed Problem Solving

With the advent of large computer and telecommunication networks, and the recognition that much human problem solving and activity involves groups of people, the integration and coordination of several human or automated problem solvers has become a

pressing concern, and the interest in concurrency and distribution has grown. Just as conventional Artificial Intelligence (AI) research sometimes uses individual human psychology as a model, distributed AI (DAI), the sub-field of AI concerned with concurrency, considers concepts such as group interaction, social organization, and society as metaphors for both the definition of problems and the formulation of their solutions. On the basis of the historical interest of researchers in DAI (Davis 1980, Davis 1982, Fehling & Erman 1983), an initial classification identified two main approaches for distributed artificial intelligence systems that suggested different types of architectures suited to specific application needs: *distributed problem solving* (DPS) and *multiagent systems* (MAS). DPS research focuses on solving problems by creating a team of coarse-grained cooperating modules that act together to solve a single task (Decker 1987, Lesser & Corkill 1987). In a pure DPS, the problem is divided into tasks, and special task performers are designed to solve these tasks. Task performing modules, also called *agents*, generally share an overall global goal and have a common language and semantic. Moreover, a single module is rarely able to solve the given problem alone because only the agent community as a whole can accomplish that. In contrast, in a pure multiagent system, research is concerned with coordinating intelligent behavior among a collection of autonomous, potentially pre-existing units, defining how they can coordinate their knowledge, goals, and skills to jointly solve a given problem (Bond & Gasser 1988). Agents in a MAS are normally more independent units, that may have to compete for the use of resources, and since they do not necessarily use the same language, they have to cope with the problem of translating other agents' functions and mapping their behavior into their own individual representation (Wittig 1992).

The term multiagent system, currently in fashion among certain sectors of the distributed artificial intelligence community, has been generically applied to any system composed of a collection of interacting agents, where both the concept of *interacting* and *agents* are - sometimes too much - loosely used. This constitutes a deviation from the initial definition of MAS, but the use of this term has spread so widely that it is now hard to distinguish between the two concepts and, even among DAI researchers, the issue is not completely resolved. In general, one realizes that there is not such a clear boundary between these concepts: many DPS systems can be said to contain some elements of MAS and vice versa. According to the wider, more loose definition of MAS, the architecture described in this dissertation can be considered a multi-agent system, because it contains agents that interact for the solution of a distributed problem. If instead, the narrower,

original definition is considered, most of the architecture's characteristics, such as the fact that the agents share a common goal, and that they have the same language and the same semantics, cause it to fall in the category of DPS systems.

DPS borrows some key aspects from *distributed processing*, but it differs from this discipline both in the problems being addressed and the methods used to address those problems (Lesser & Corkill 1981). Both methodologies normally make use of networks of interacting task-solving entities. In distributed processing, multiple tasks execute concurrently, and they interact by sharing access to physical or informational resources. The goal of the network-operating system that manages the distributed processing is to preserve the illusion that each task has dedicated access, by hiding the resource-sharing interactions and conflicts. In contrast, in a DPS system, problem-solving procedures are explicitly aware of the distribution of the network components, and can make informed decisions based on that knowledge. This difference is closely related to the different characteristics of the applications these methodologies are normally used for. In traditional distributed processing networks, task decomposition is normally organized in such a way that a node rarely requires assistance from other nodes for the execution of its tasks, and emphasis is placed on achieving higher performance by effectively exploiting the concurrency inherent in certain applications. Thus most of the distributed-processing paradigms do not explicitly address the issue of cooperative task interaction to solve a given problem.

4.2 Functionally Accurate, Cooperative Distributed Systems

The FA/C paradigm for modeling cooperation among interacting problem solvers was introduced by Lesser & Corkill (1981) in response to the perceived inadequacy of conventional distributed systems to deal with certain classes of problems. Conventional models are characterized by their emphasis on maintaining correctness in all aspects of the distributed computation: tasks are decomposed so that each agent has sufficient data to solve its assigned tasks completely and accurately, with little interaction with other agents. These systems are composed of *completely accurate, nearly autonomous* agents (CA/NA), because the problem solving performed by each agent relies on complete and accurate (correct) information, that is normally stored in a local database, or requested from

another agent and received as a complete and correct result. For many applications, such as distributed planning, distributed interpretation, and resource allocation, that naturally lend themselves to a distributed implementation, this model is inappropriate, because it is not always possible to effectively replicate or partition algorithms and control structures so as to match the natural distribution of data and problem solving knowledge. In this situation, implementing a CA/NA system is not always practical and effective because of the high communication and synchronization costs required to guarantee completeness and consistency of the local databases. Despite the continuous increase in speed of both local and wide area networks, the computational cost associated with the transmission and reception of information can still be very high: if a large amount of information must be exchanged, especially among heterogeneous agents, the computational cost of selecting and packaging the information on the sending side, and the assimilation of the information on the receiving side can be significant. Furthermore, in real-time applications, if interacting agents must frequently synchronize their behavior, the delay incurred may significantly affect their overall performance.

The key issue thus becomes how to organize effective cooperation among agents, with low communication costs and synchronization delays, in order to produce an acceptable solution in reasonable time. The basic intuition behind the FA/C approach is that agents need not always have all the information necessary to completely and accurately solve their subproblems. For some applications, there exist inter-agent constraints among subproblems that can be exploited to resolve the inconsistencies that may arise due to the lack of accurate, complete and up-to-date information. Agents can produce tentative, partial results based on local information and then exchange them with other agents to resolve local uncertainties and global inconsistencies.

So far, most of the applications of the FA/C concept have been in distributed interpretation. The following example on sensor data interpretation for aircraft monitoring, illustrated in Figure 4.1 (Lesser 1991) provides an idea of the type of interaction that occurs between FA/C agents in a distributed system. Two acoustic sensors, covering two overlapping regions, provide information on the presence and tracks of aircraft, based on the spectrum of acoustic frequencies produced by each aircraft. There are two agents A and B, responsible for monitoring the two regions, each one receiving data from one sensor. The overall goal is to interpret the sensor data and identify the aircraft that are moving in the area.

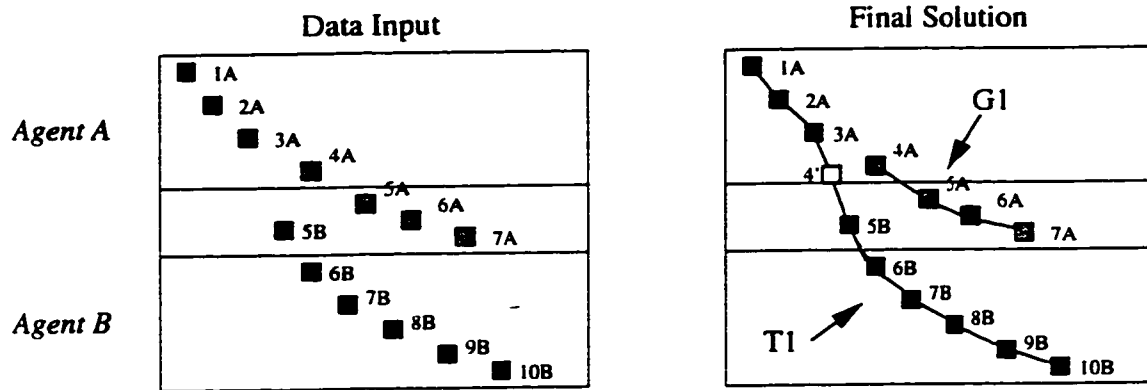


Figure 4.1: A distributed, two-agent aircraft-monitoring scenario: the different shading of the data points reflects the strength of the signal received (dark is stronger)

The uncertainty of a solution is due to several reasons, that include improperly sensed signals, environmental noise, and ghosting (the erroneous interpretation of data as a track, caused by the environmental reflection of sound). The agents must communicate in order to produce reasonable levels of certainty in their interpretation hypotheses and reach a consensus on a correct solution. In the example, without proper communication, agent A might wrongly interpret its data as a ghost track. In fact, while the first three data points are received as a strong signal, the remaining four are too weak to properly represent a trajectory. Therefore, since a true trajectory can not stop in mid-air, the most credible interpretation of the data, from agent A's local perspective, would be to catalog all received data points as a long ghost track. Receiving a possible interpretation of the data as a track T_1 from agent B, allows agent A to review its data interpretation: since a track T_1 exists, agent A interprets its first three points as the beginning of track T_1 (that continues in zone B) and the last four as a ghost signal G_1 caused by the environmental reflection of sound.

4.2.1 Early Experimental Results: Speech Recognition

The first major application of the FA/C concepts was a distributed version of the well-known Hearsay-II speech understanding system (Lesser & Erman 1980). Its implementation provided the developers with many significant insights, and highlighted important shortcomings of the newly defined FA/C paradigm. The system was composed of three agents, each of which was a complete Hearsay-II module (Lesser, Fennel, Erman & Reddy 1975), enhanced to allow the transmission of high-level interpretation results (e.g.,

phrase hypotheses). Each agent received acoustic data, with a certain degree of redundancy, in order to guarantee that all the information needed to recognize a word was locally available to at least one agent. This allowed the agents to formulate an interpretation without having to exchange the original acoustic data. Experiments showed that the agents were able to interact using a simple communication protocol in a cooperative fashion, and arrive to the same solution of the centralized version, achieving a slight processing speedup. The experiments also showed some weaknesses in the cooperative approach, mostly due to the incoherent behavior of the agents. Agents were sometimes observed to transmit information that was either not relevant or not timely, and to reach conclusions already reached by some other agent. Also, the agents did not focus on generating results aimed at improving other agents' processing, and they had a tendency to *distraction*, i.e., to pursue reasoning processes that were based on highly uncertain hypotheses and led to highly uncertain conclusions. This was caused by the sometimes too quick transmission of highly uncertain hypotheses to other agents, without first trying to confirm them using available information. These problems were attributed to a strongly self-directed behavior of individual agents, caused by the inadequacy of the methods of cooperation, that made agents unaware of the goals that other agents were trying to pursue. For example, an implicit form of cooperation occurred when an agent treated the results received from other agents in the same way as locally generated results, but such results were not accompanied by adequate information describing the reasons for their uncertainty and the reason why they were produced.

4.2.2 Explicit Goal Structures: Vehicle Monitoring

The first reaction to these issues was the understanding that effective cooperative control is based on exploiting relationships among agents' goal structures, and that an explicit representation of a goal structure is necessary to analyze the strengths and weaknesses of different cooperative control strategies. While experiments on the distributed version of Hearsay-II provided several insights on how problems are related and how problem solvers may interact for the formulation of a solution, the rather fixed control structure of the system limited the range of further experimentation. As a result, a new system was developed, for distributed Sensor Interpretation (SI) aimed at monitoring vehicle movements in a two-dimensional space. The new architecture, called Distributed Vehicle Monitoring Testbed (DVMT, (Lesser & Corkill 1983)) was designed in a more modular fashion, to allow for the

exploration of different organizational structures for configuring agents in the system. The aim of this project was to improve cooperative control by:

- more effectively exploiting constraints implicit in received information, with the effect of making agents' goal structures more consistent in terms of the importance associated to achieving interdependent subgoals;
- permitting agents to explicitly request information from other agents, thus increasing the timeliness and relevance of exchanged information
- clearly distinguishing between externally and locally directed activities, and evaluating the importance of different sources of problem solving data, thus allowing an agent to be guided by the information generated by the agent that is most likely to produce accurate information.

Extensions to the control cycle employed for the distributed Hearsay-II model involved explicitly constructing a partial representation of the agents' problem solving goal structure and creating a set of data-directed goals, i.e., associating to transmitted data, abstract information that represents an approximation of the potential results which can be directly produced using the new information.

Three important concepts for understanding the complexity of a distributed search and identifying methods to reduce it are those of solution, control, and cooperative control uncertainty.

- *Solution uncertainty* refers to uncertainty over whether the solution to a sub-goal is going to be part of the final solution of the system's top-level goal. This uncertainty may be due to several reasons. It may be caused by the unavailability of the information necessary to judge the acceptability of a goal's solution, at the time the solution is generated. It may derive from the difference between local and global criteria for rating solutions. Or it may originate from uncertainty on the validity of the data used to construct the solution.
- *Control uncertainty* refers to the uncertainty over how to organize the search for a local solution: which goals need to be pursued during the problem solving process, what partial solutions should be used, and how much effort should be expended to reach a certain goal. The uncertainty is due to the fact that there can be, in general,

multiple ways to achieve the top-level goal, and it may be difficult to assess the effort needed to produce a solution, thus making it difficult to decide which subgoals can and should be pursued.

- *Cooperative control uncertainty* is associated with the organization of the problem solving process in a distributed environment and relates to an agent's view of the state of problem solving in other agents: how complex are the subtasks other agents are working on; what progress has been made; what type of information may be beneficial to their problem solving. This type of uncertainty leads to uncertainties in decisions concerning the communication of local information and the selection of goals to be locally pursued, such as what type of results or goal should be communicated, when they should be transmitted, how should externally received information be regarded, and what is an appropriate balance between pursuing local activities and responding to external request.

In a distributed search, only a part of the goal structure is available for decision-making, thus increasing the solution and control uncertainty of the search process. To illustrate this point, consider Figure 4.2, that describes a centralized goal structure (Lesser 1991). This is a classical *and/or* tree (Shapiro 1987), that has been augmented to represent constraints among goals (or among the solutions for goals), and data or resources needed to solve low-level goals. Numerical ratings may be associated with goals that provide an indication of their current worth or applicability, based on the importance of achieving those goals within the problem solving process. Constraints can exist between solutions of sibling goals, but they can also be more distant in the goal structure, and associated with goals at different levels. Whenever constraints exist between two goals, these goals become interdependent, and the satisfaction of one of them may have repercussion on the other, make it redundant or required, and in general modify the current rating associated with it.

Let us assume that, within a centralized search process, based on the goal structure shown in Figure 4.2, the search has reached a point where the following four different goals can be achieved (the number in parentheses represent their associated ratings): $G_{k-1}(90)$, $G_{1,2}(60)$, $G_{1,3}(50)$, $G_{k,2,2}(40)$. The goal with the highest rating, G_{k-1} , is processed first, and based on its solution, the ratings associated with the remaining goals are reevaluated. New goals are also generated by the solution of G_{k-1} , and added to the list of goals to satisfy. Assume the list is now $G_{k,2,2}(60)$, $G_{1,1}(60)$, $G_{1,2}(50)$, $G_{1,3}(40)$ (these goals have

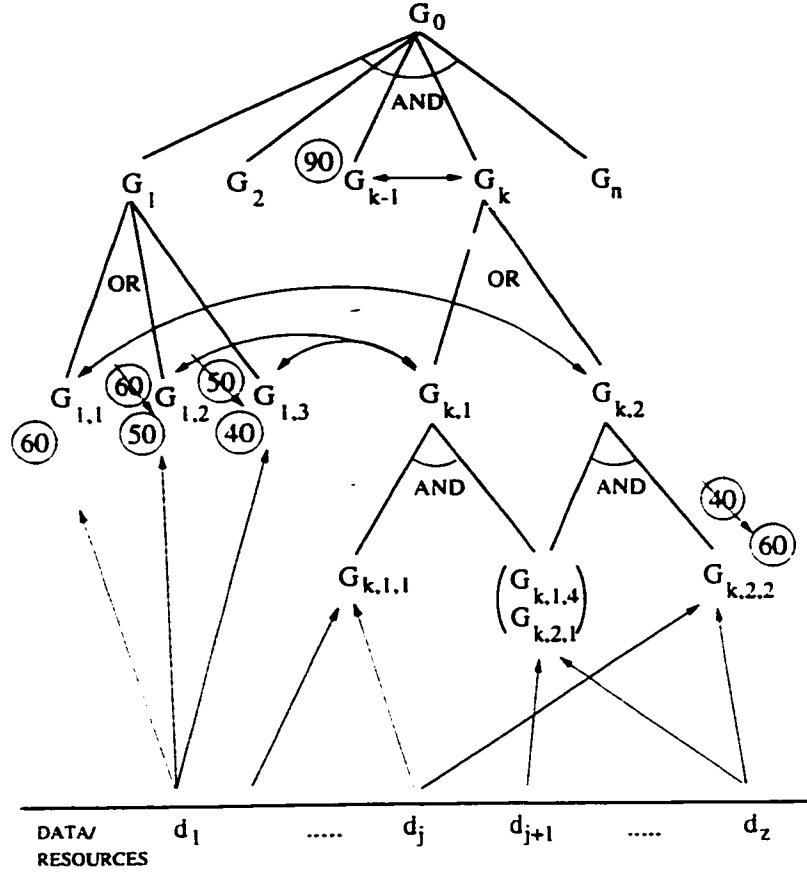


Figure 4.2: A centralized goal tree

been affected by the resolution of other goals and their ratings have changed during the process through interdependency chains). Assuming that solutions for G_{k-1} , $G_{k,2,2}$, and $G_{1,1}$ were found to be part of the final solution, it is evident how having solved G_{k-1} has provided information that resulted in a more informed control decision, by increasing the rating of $G_{k,2,2}$ and adding $G_{1,1}$ to the list of goals to process.

In a two-agent distributed environment, the same search may have a different, less efficient control flow, if the two agents do not properly coordinate their problem solving. Consider the same goal-structure, in this case distributed among the two agents, as represented in Figure 4.3 and assume the search process to be two steps earlier, with respect to the centralized case. Assume the list of goals to be processed contains the same goals as in the centralized case, plus goals $G_2^1(95)$ and $G_n^2(95)$. In this case *Agent*₁ has the choice between attempting to solve $G_2^1(95)$, $G_{k-1}^1(90)$, and $G_{1,2}^1(60)$. *Agent*₂ has the choice between

$G_n^2(95)$, $G_{1,3}^2(50)$, and $G_{k,2,2}^2(40)$ (the superscript refers to the agent the goal is associated with). The agents choose respectively G_2^1 and G_n^2 , because they have higher ratings. After processing G_n^2 , $Agent_2$ has the choice between $G_{1,3}^2(50)$, and $G_{k,2,2}^2(40)$, neither of which is highly rated nor well differentiated from the other. If no interaction with $Agent_1$ occurs, at this time $Agent_2$ processes $G_{1,3}^2$, because of its higher rating. If the agents are able to explicitly represent interdependency links among goals and update their goal structure based on other agents processing, the solution of goal G_{k-1}^1 increases the rating associated to $G_{k,2,2}^2$ (as it happens in the centralized case), thus making $G_{k,2,2}^2$ a better choice than $G_{1,3}^2$ for $Agent_1$.

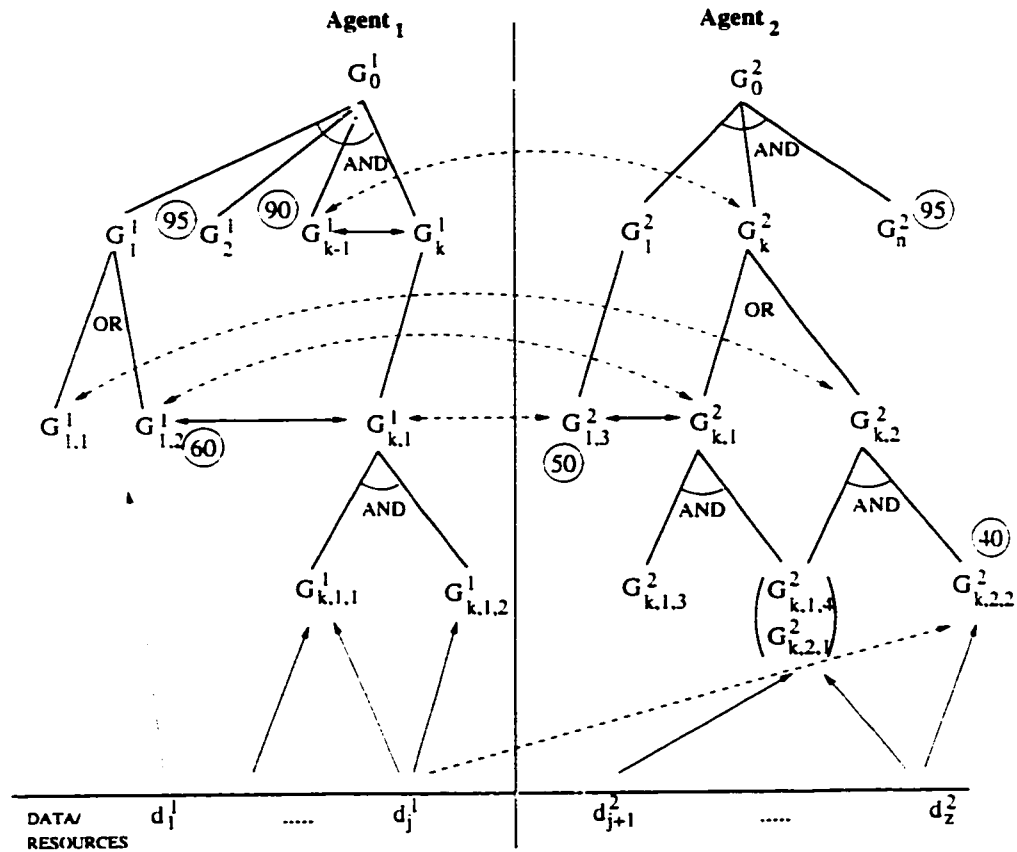


Figure 4.3: A distributed goal tree

This example shows how a well-coordinated search can reduce control uncertainty. One way to achieve this is for agents that have both highly rated goals and low control uncertainty to reorder their goal sequence, in such a way to balance the efficiency of their local search with the needs of other agents, by first processing those goals that can reduce

the control and solution uncertainty of other agents. In the example, if $Agent_1$ had chosen to process $G_{k-1}^1(90)$ rather than $G_2^1(95)$, $Agent_2$ would have been able to process its own goal in a better informed way. The lack of this type of coordination in the distributed Hearsay-II model was partially responsible for the generation of incorrect results and the distraction phenomenon.

Solution uncertainty in a distributed search can be caused by the unavailability, within an agent's database, of the data necessary to compose a certain solution. For example in Figure 4.3, the solution to $G_{k,2,2}^2$ is uncertain, because it requires datum d_j^1 that is not directly available.

When an agent requests another agent to pursue a certain goal, the ability to associate additional information and partial results to the requested goal helps reduce incoherent cooperative behavior among the agents. An informed, externally directed communication can be implemented by letting agents transmit results only in response to goals requested by other agents, or direct their problem solving towards the generation of information useful to decrease the solution and control uncertainty of the requesting agent, thus generating and transmitting information in a more relevant and timely fashion.

The rating of a goal indicates the estimated importance of achieving it, within the problem solving process. The way an agent computes the rating of externally received goals, or goals that were generated based on externally received information, determines the self-directedness of an agent, by balancing between satisfying the needs of other agents and pursuing the agent's own goals. The higher is the weight of externally received goal, the more externally directed is the behavior of the agent, and in general the higher is the potential for distraction. On the other hand, a system composed of highly self-directed agents is prone to a high control and solution uncertainty, because the information received from other agents will be only partially exploited. By choosing the appropriate balance, the agents' potential for distraction can be reduced while still guaranteeing the right exploitation of externally received information.

4.2.3 Multistage Negotiation: Distributed Constraint Satisfaction

In the first ten years since its conception, the FA/C approach has been applied almost exclusively to distributed sensor data interpretation. Its first significant application to a different domain is reported by Conry, Kuwabara, Lesser & Meyer (1991), where the

FA/C concept provides the basis for the definition of a cooperation protocol, called *multistage negotiation* for satisfaction of constraints that arise from the allocation of shared resources in a distributed environment. An architecture is developed, composed of a collection of cooperating agents, each of which has an incomplete view of the system resources and knowledge of only partial solutions to the overall procedures. The system is applied to the restoration of transmission paths for dedicated circuits in a complex communication network. The network is composed of sites interconnected by links, each containing several pieces of connection equipment. The sites are partitioned into geographic sub-areas with a single site for each sub-area. Each control facility monitors and controls communication within its sub-area, and corresponds to a single agent in the distributed problem solving network, as shown in Figure 4.4. An equipment failure or power outage may cause an interruption of service, and the loss of the communication link between two or more locations. Each interruption gives rise to a network task whose solution involves restoring a path between the disconnected sites by using alternative links.

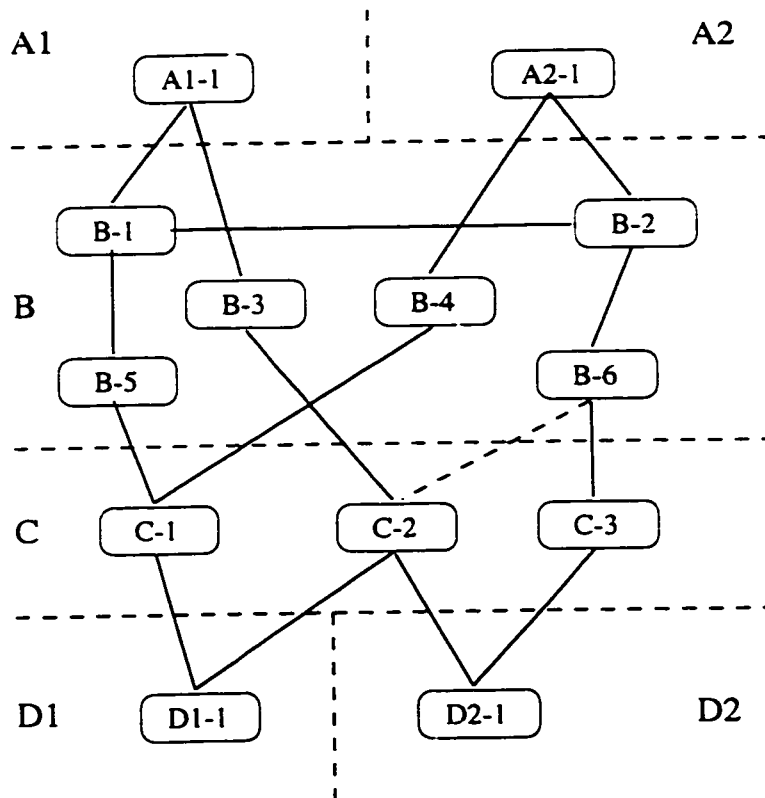


Figure 4.4: Multistage negotiation: example of a communication network

The lack of a global view of the problem and of the available resources may require substantial interaction among the agents to assess the set of indirect constraints that develop for the use of the resources. The agents must engage in simultaneously determining a solution to the overall problem and recognizing when parts of the problem are overconstrained. They must do so without falling into cycles of unproductive activity.

The multistage protocol is a three-phase process composed of an asynchronous local search phase, a coordinated search phase, and an overconstrained resolution phase. In each of these phases, agents handle goals in a progressively more coordinated way, based upon aggregated evidence determined through the construction of *conflict sets* and the explicit representation of a goal structure that makes conflict explicit. Conflict sets represent sets of subgoals that cannot be satisfied concurrently. Goals coincide with the restoration of interrupted paths and require the use of communication resources, such as links, and other equipment. They are structured into primary and secondary goals, depending on whether the corresponding circuit loss is local or remote to the agent. In the asynchronous search phase, each agent looks for solutions independently from other agents. When all primary goals have been analyzed, if a solution has not been found, the agents start the coordinated search phase. If an overconstrained situation is recognized, some goals need to be abandoned to resolve some of the conflicts. The recognition of these goals is performed in the overconstrained resolution phase.

4.2.4 More Sophistication: Aircraft Monitoring

A further next step in the research for reasoning capabilities to support effective cooperation was determined by the recognized need to explicitly consider why existing evidence for hypotheses is uncertain and what additional evidence is required. It involved the development of a new distributed architecture composed of agents with more sophisticated models of their evidence and their problem solving states. A centralized system called RESUN (Carver 1990), and a distributed architecture composed of RESUN agents, called DRESUN (Carver, Cvetanovic & Lesser 1991) were developed for the solution of a DVMT-like aircraft monitoring problem (see the example at the beginning of this chapter).

Carver & Lesser (1996) focus on the issue of *solution uncertainty*, i.e., on the uncertainty about the correctness of partial solutions due to agents having incomplete local information, rather than on *control uncertainty*, which, as defined earlier, is concerned

with determining what actions agents should take and when they should be taken, to achieve efficient, globally coherent problem solving. It is essential to remember that while the complexity of a centralized sensor interpretation problem is given by the existence of possible alternative explanations for a data set or by the uncertainty in the data set induced by environmental noise, masking, or sensor errors. Such complexity increases in a distributed environment, where each agent has direct access to only a subset of the input data, thus monitoring only a portion of the overall area of interest, and agents' local solutions must be combined in order to construct a consistent, conflict-free global solution. In many scenarios, the selection of a distributed approach is not driven by the desire for efficiency or speed improvement, but rather by the distributed nature of the problem domain itself. In such cases, cooperation is necessary because of the distribution of the available data, and the presence of multiple problem solvers. The interpretation of local data performed by an agent can be used by another agent to corroborate or reject its own hypotheses, that were constructed using only a subset of the input data.

RESUN agents maintain explicit representations of the reasons why their hypotheses are uncertain, of the state of their goals, and of the actions being taken to meet these goals. Symbolic statements describing *sources of uncertainty* (SOU) are attached to hypotheses when these are formulated or refined, that represent the reasons why these hypotheses are uncertain, based on local evidence. For instance, in the aircraft monitoring scenario, a hypothesis on a track may be uncertain because its supporting sensor data are incomplete, or because the data may support more than one interpretation. The interpretation process in RESUN is an incremental one, that tries to gather evidences to resolve those SOUs that keep the current answer from being sufficiently certain to satisfy prespecified termination criteria. Termination criteria that are not satisfied or have not yet been verified as satisfied, are viewed as SOUs, and goals which represent the need to resolve those uncertainties are created to drive the problem solving process.

RESUN agents were the basic modules for the construction of the DRESUN distributed architecture. They were extended for distributed problem solving, by providing them with the ability to represent global consistency termination criteria, inter-agent communication dialogues, and evidence coming from other agents. Compatibly with the RESUN model of control, driven by the desire to resolve uncertainty, the verification of global consistency is performed by generating appropriate SOUs, that represent the uncertainty about the global consistency of an agent's local solutions, whenever an agent realizes that its so-

lutions may interact with those of other agents. To represent these potential interactions, DRESUN agents create *global consistency SOUs* (GSOUs), that denote assumptions that the gathering of additional information by another agent may corroborate or contradict a certain local hypothesis. Carver & Lesser (194) define three types of GSOUs, corresponding to global consistency criteria. For each type of GSOU a specific resolution model is designed. To resolve a GSOU means to determine what information needs to be gathered and what processing needs to be performed to reduce or eliminate the source of uncertainty. Some of the models are modified versions of those used to handle SOUs in the centralized RESUN, while others are designed anew. The three consistency criteria and the corresponding GSOUs, examples of which are illustrated in Figure 4.5, are:

- Interpretation solutions involving overlapping monitoring regions must be consistent. The corresponding GSOU resolution is handled by the *consistent-overlapping-model*.
- Track hypotheses that can extend into regions monitored by other agents must be consistent. This situation is handled by the *consistent-global-extension* GSOU, an extended version of the model that is used by the centralized system for judging the completeness of tracks, within the region under its control.
- Agents must be able to find external evidence if a hypothesis requires evidence that could be in other agents' monitoring sectors (for instance if an agent interprets a ghost signal, the corresponding source signal may be in some other sector and has to be interpreted by some agent). The search for this type of external evidence is driven by the creation of *consistent-global-evidence* GSOUs and handled by the corresponding model.

4.2.5 Solution Quality

The development of the RESUN system and of its distributed version posed the basis for a more thorough analysis of the quality of the solutions that can be determined using a FA/C approach to cooperation. The analysis mainly focused on identifying necessary conditions for effective FA/C-based problem solving. The study was performed in the context of distributed SI, using the flexibility offered by the new models, but some of the conclusions can be extended to a more general context.

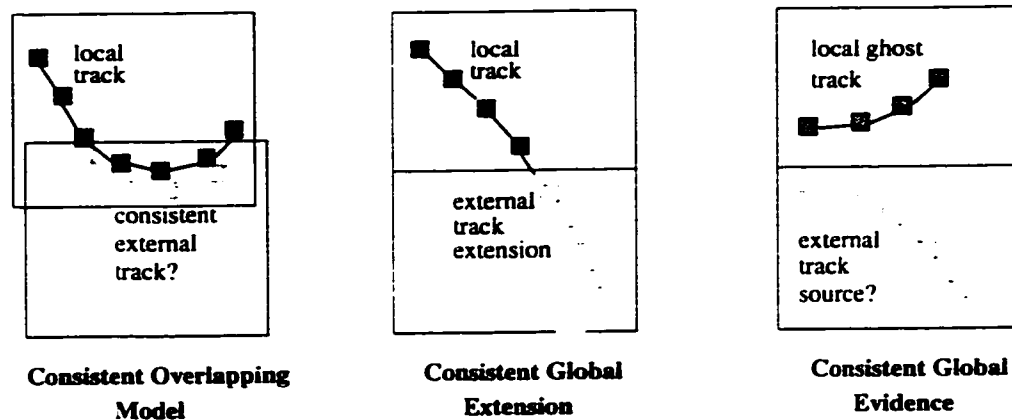


Figure 4.5: Examples of global consistency sources of uncertainty for vehicle monitoring

The analysis of the solutions in a distributed environment requires a standard for comparison. The “globally best solution” to an SI problem is an interpretation of the available data that is judged “best” according to some criteria. One possible criterion is the *most probable explanation* (MPE) (Pearl 1988), which is not necessarily the *correct* interpretation, i.e., what actually produced the data, but the interpretation that is most likely, given the actual data and a model of how data can be produced by different events. The problem with such a definition is that for many real-world problems, it is impractical to compute the true MPE, both in a centralized and in a distributed environment, because of the large volumes of data involved and because of what’s known as *correlation ambiguity*, i.e., the uncertainty about which potential explanation hypothesis a *support instance* should be associated with. A support instance is an instance of a predefined class of entities that is responsible for the generation of the measured data, like an aircraft or a vehicle responsible for the sound that is picked by the sensors. While typical diagnosis problems have a fixed set of causes and possible effects, with relatively fixed relations among them, interpretation problems, while normally having a fixed set of cause types and a fixed set of data types, can have an indeterminate number of instances of these types (e.g., aircrafts in an air space), that result in a combinatorial explosion of possible explanations for the data set. Systems that have to deal with this kind of complexity must use approximate, satisficing strategies to determine solutions.

The following are basic approximation techniques that can be used by SI systems:

- process only a subset of the available data;

- construct only some of the possible interpretation hypotheses for the analyzed data;
- estimate approximate conditional probabilities (relations between cause and effect) for the hypotheses;
- consider only some of the possible composite interpretations;
- use some methods other than maximum joint probability to select a solution (e.g., acceptance thresholds).

The adoption of these techniques results in the formulation of solutions that are only approximations of the MPE.

To analyze the difference between an exact solution methodology and an approximate one, it is important to understand what and how strategies can be applied to generate these solutions. A solution is generated by resolving SOUs and GSOU. Resolving an SOU is an intra-agent evidence propagation process that identifies a possible explanation for a piece of data or a hypothesis. Analogously, resolving a GSOU involves exchanging information between agents so as to propagate the effects of each agent's local evidence among the agents, by establishing evidential links among the hypothesis networks of the agents. SOU and GSOU resolution strategies may differ based on how completely and accurately local and global interactions are considered, and on different criteria used to decide which source of uncertainty (local or global) to pursue next.

An *exact local interpretation strategy* (ELIS) is defined as a strategy that determines the *true* MPE based on the complete local data set of the agent (ignoring global interactions), and it involves the resolution of *all* the possible SOUs. An *approximate local interpretation strategy* (ALIS) is any strategy that is not exact.

For GSOU, a key criterion imposed on GSOU resolution strategies is that the agents' local solutions must be "consistent" (i.e., non-contradictory) at termination. In other words, whatever strategy is used, it is necessary to resolve enough GSOU to guarantee that the local solutions will be either identical, independent, or corroborative. The most comprehensive resolution strategy requires *all* GSOU to be *always* completely resolved, i.e., all possible interpretation hypotheses from all possible joint interpretations must be considered. This is called *complete GSOU resolution strategy* (CRS). Any strategy that does not resolve all GSOU is incomplete. An example of an incomplete coordination strategy is the *local solutions GSOU resolution strategy* (LSRS), according to which, unless

local solutions are inconsistent, each agent resolves only those GSOU's that are associated with its current local solution, i.e., the "best" interpretation of its own local data set (using some common definition of best), and pursues evidence propagation using only the local solution hypotheses of the relevant external agents.

For a clarification, consider Figure 4.6. It shows examples of the application of ELIS and LSRS in two different, consistent and inconsistent, scenarios. The two agents *Agent*₁ and *Agent*₂ must determine an explanation of the input data. Input data sources are partitioned among the agents, and based on the input data each agent must determine a solution. ELIS requires all possible GSOU's to be resolved (in Figure 4.6, this is shown with arrows from all data sources to all possible explanations, h_1 and h_2). Whether the local solutions are consistent (Figure 4.6 a), or not consistent (Figure 4.6 b), all possible GSOU's are resolved, and hypothesis h_2 is found to be better. LSRS instead solves only the GSOU's associated with the local solution, and these guarantee consistency. In case of global consistency (i.e., if the local hypotheses reached by the two agents are mutually consistent), the final solution (h_1 in this case) is created without considering the complete set of global interpretations (i.e., without analyzing the GSOU's associated with h_2). If the two solutions are inconsistent, in order to satisfy the global consistency criteria, all GSOU's must be resolved. In such cases, the same conclusion obtained by ELIS is reached, and h_2 is the final solution.

The analysis of solution quality considers both exact and approximate interpretation strategies. For the reasons described above, exact strategies in many real-world scenarios are not practical or feasible, but the analysis of their solutions is used to draw some results on the quality of the solutions that can be determined using approximate ones.

Since the definition and the computation of the MPE is often an impractical task, and because a substantial component of the complexity of the problem of distributed sensor interpretation lies in the interaction among agents, Carver & Lesser (1996) compare the quality of distributed SI solutions to those of an equivalent centralized system. An equivalent centralized system is a (hypothetical) single-agent system that has access to all the data of the distributed system, and that uses the same interpretation strategies.

The analysis performed by Carver & Lesser (1996) is performed by formulating and demonstrating four theorems that qualitatively relate solutions obtained with a distributed system to those obtainable using a centralized approach, by applying the SOU resolution strategies described earlier. The analysis is a theoretical one and no experimental result is

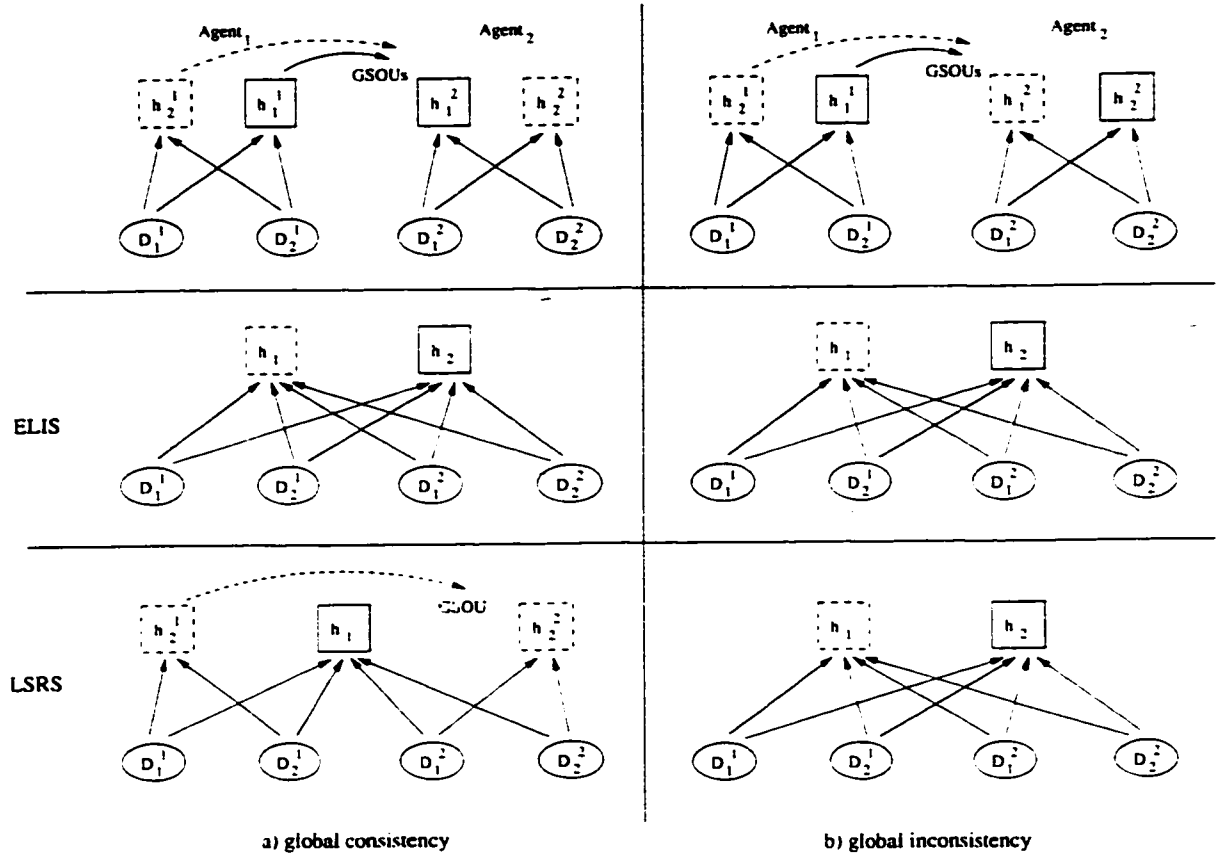


Figure 4.6: The application of different global resolution strategies

presented. This analysis can by no means be considered a conclusive one, yet it provides some interesting insights on the type of problem domains for which some of the FA/C strategies are more likely to be effective.

The first theorem compares the quality of FA/C global solutions, assuming that the centralized system and the distributed agents use the *exact local interpretation strategy* (ELIS) to resolve local SOUs, and that the distributed agents use the *complete GSOU resolution strategy* (CRS). Under such assumptions,

the FA/C distributed system produces the same exact interpretation as the centralized system, and that interpretation is the true MPE of all the globally available data.

This result may seem an obvious one, given the definitions of ELIS and CRS, and the assumption, inherent in the DRESUN model, that the GSOU's represent all the possible

interactions for the set of created hypotheses. Nonetheless it shows DRESUN's theoretical capabilities, if all inter-agent interactions are explicitly represented and a mechanism for controlling their resolution is used. Unfortunately, both exact local interpretation and complete GSOU resolution are not practical in many interpretation domains.

The second theorem thus looks at the potential loss of correctness that the *local solutions GSOU resolution strategy* (LSRS) may induce. LSRS is an important strategy, since, even though it does not always guarantee a correct result, it can be very efficient under certain circumstances, and it was one of the first ones used in the initial FA/C experiments.

If a centralized system and a set of distributed agents use the same arbitrary (approximate) local interpretation strategy, and the distributed systems use the *local solutions GSOU resolution strategy*, then the centralized and the distributed systems do not necessarily produce the same interpretation, and the solution produced by the distributed system is not necessarily the true MPE of the globally available data.

In other words, the solution that is *locally* most likely in all agents may not be the *globally* most likely solution. This result is true for all reasoning problems based on incomplete or uncertain information, and it is due to the inherent *non-monotonicity* of such domains. A non-monotonic domain is one where the reasoning process involves drawing conclusions that are based on incomplete sets of data and that may have to be discarded in light of new evidence, obtained by receiving or synthesizing additional data. In particular, those distributed problems for which it is not possible to guarantee that every task-solver has a complete and correct set of data available to perform its reasoning, are non-monotonic.

The justification behind the use of local interpretation strategies lies in the recognition that there are real-world problems for which once a local solution is achieved with a high degree of confidence, while it is possible that additional evidence can negatively affect such confidence, substantial additional evidence will be required for it to have a major effect on it. For instance, in vehicle monitoring, a considerable amount of evidence must generally be gained to hypothesize a track with high belief. And, even if additional evidence may reduce the degree of belief in that hypothesis, substantial additional evidence is required before majorly affecting such belief.

Given these premises, Carver & Lesser (1996) consider another approximate GSOU resolution strategy, based on the concept of *acceptability* of a solution: an interpretation is defined *acceptable* if all the hypotheses associated to it have a belief rating higher than a

given threshold. The strategy, called the *incremental GSOU resolution strategy* is defined as follows: each agent will resolve the GSOU's as in the LSRS, but if its resulting local solution is not *acceptable*, the agent must resolve other GSOU's associated with the hypotheses in its solution until either the solution becomes acceptable or all the GSOU's have been resolved (i.e., as in the *complete GSOU's resolution strategy*). The strategy is incremental because once a basic set of GSOU's have been resolved (as in LSRS), the GSOU resolution process will be iterated until the solution becomes acceptable or all GSOU's have been resolved.

The next two theorems basically prove that the implementation of the incremental GSOU resolution strategy does not necessarily provide any immediate improvement, but the formulation of the strategy and the analysis of its complexity helps the FA/C developer understand the effects that interpretation and coordination strategies, acceptance thresholds, the domain characteristics, and specific agent organization have on the *probability* of achieving global solutions that are comparable to those that are obtainable by an equivalent centralized system.

The third theorem analyzes the potential gain of the *incremental GSOU resolution strategy*, assuming that at the local level the *exact local interpretation strategy* is used:

assuming that a centralized SI system and the DRESUN agents use ELIS, and the agents use the *incremental GSOU resolution strategy*, if the centralized SI system is able to produce an acceptable solution for a data set, then so is the distributed system (using the same acceptance threshold).

Such a solution may not be the same as the one determined by the centralized system, and it's not necessarily the MPE.

The fourth theorem also consider the *incremental GSOU resolution strategy*, but in this case it is assumed that the agents use an *approximate local interpretation strategy*, rather than ELIS, as in the last theorem. The finding is that, under such assumptions:

even if a centralized SI system is able to produce an acceptable solution for a data set, the distributed DRESUN system may not.

Basically, if a satisficing approach to constructing solutions is adopted, then a FA/C-based system *can* produce results that are "comparable" to those of a centralized system, but only under particular conditions.

4.2.6 Nearly Monotonic Domains

Carver, Lesser & Whitehair's (1996) current research tries to analyze the conditions for which satisficing results can be obtained using a FA/C approach. This is an important concept in FA/C research: if agents can limit the communication to the transmission of mainly solution-level hypotheses rather than raw data, and these hypotheses can be used, with minor additional work, to produce satisficing global solutions, then effective problem solving can be achieved with greatly reduced communication.

The basic assumption is that *there are* domains whose characteristics justify the role of local solutions in producing global ones, thus supporting the adoption of certain types of incomplete coordination strategies. Carver et al. (1996) call such domains *nearly monotonic* and attempt to define criteria to recognize them.

A proper, formal definition of *near monotonicity* has not been found yet. Carver et al. (1996) adopt probabilistic models of the likelihood of a hypothesis being correct given its current belief, or rating, computed based only on the local data processed by the agent and define *nearly monotonic* domains as those for which that likelihood is sufficiently high, when the current (locally estimated) belief is high. In other words, letting

- $\mathcal{D}$ be the complete, globally available data set;
- D be a subset of $\mathcal{D}$;
- $BEL(H)$ be the current belief in hypothesis H , given data set D ;
- $BEL^*(H)$ be the "true" belief in hypothesis H , for data set $\mathcal{D}$;
- MPE be the current MPE solution, given data set D ;
- MPE^* be the "true" MPE solution, for data set $\mathcal{D}$;

a nearly monotonic domain is one in which the following is true: for any hypothesis H , if the locally computed belief of H is *fairly high*, then the probability that H will ultimately be part of MPE^* is also *fairly high*.

Carver et al. (1996) define a generalization of the incomplete global solution strategies described above, the *consistent local solution strategy*, (CLSS), for the formulation of a theorem that demonstrates the potentials of the application of the FA/C approach to the solution of problems that have near monotonic characteristics. CLSS can be described by the following steps:

1. Initially each agent uses only its own local data to develop a local solution.
2. Upon satisfying some solution criteria, each agent communicates its solution's hypotheses to all agents with which it has recognized subproblem interactions (for which it has created GSOU's).
3. The agent also sends its solution to all other agents from which it has received solutions.
4. Processing terminates when all agents have transmitted their solutions and have integrated their solutions with those received from other agents.
5. The global solution is the union of all the final, integrated agent solutions.

As in the other incomplete solution strategies, if two agents' local solutions are consistent, then they will be directly integrated into a final global solution. If local solutions are inconsistent, then the agents must engage in further communication, to resolve additional GSOU's. The definition of consistency is an evidential one: solutions are consistent if the hypotheses that comprise each of the local solutions are either pairwise identical, independent, or corroborative. Two hypotheses are corroborative when one is evidence for the other or when they are of the same type and can be merged into a single one (e.g., in the case of aircraft monitoring, when two consistent partial tracks can be merged into a longer one). The attractiveness of nearly monotonic problems is motivated by the realization that consistent local solutions can be highly predictive of the goodness of global solutions resulting from their merge.

The most recent finding in the analysis of the solution quality of FA/C-based reasoning is a theorem by Carver et al. (1996) that shows that

in nearly monotonic domains, the higher is the probability that a hypothesis H is in the MPE, given its belief (computed by a local agent using only its local data), the higher is the probability that the *consistent local solution strategy* will select H as part of the final solution.

In other words, incomplete solution strategies such as the *consistent local solution strategy*, under certain assumptions can *potentially* produce a global solution whose components (the hypotheses associated to it) are as likely to be in the MPE global solution as desired (the term *potentially* is used because the theorem is based on likelihood).

These assumptions, that were also recognized as basic requirements in Lesser (1991) are:

- That agents can produce local solutions whose hypotheses have high enough beliefs. This condition has to be verified because, otherwise, since the belief based on local data is a lower bound to the belief based on global data, the local belief would constitute an insignificant (too low) lower bound to the likelihood of the hypothesis of being in the global MPE.
- That local solutions are largely consistent. Whenever the hypotheses produced by each agent (using only its own local data) are not consistent, by the definition of the *consistent local solution strategy*, additional exchange of information is required to resolve uncertainties and produce consistent solutions, with a consequential loss of performance.

It is important to recognize the role that near monotonicity plays in the successful application of incomplete solution strategies, and the importance of producing consistent local solutions that satisfy certain properties, such as having a sufficiently high local belief, or allowing alternative explanations. From this perspective, the concept of near monotonicity can be extended from dealing with single hypotheses, to considering collections of interpretations, that have been evaluated based on their local consistency (Carver et al. 1996). Also, certain solution components, namely those with higher beliefs, are more useful than others in developing a global solution, confirming the intuition that communication of highly uncertain local solution components is not usually advantageous.

The next chapter will describe the application of some of the key concepts of the FAC: research to the problem of providing suitable real-time traffic control decision support in a distributed environment, in response to the occurrence of incidents.

Chapter 5

The Distributed Architecture

This chapter presents the distributed system CARTESIUS as a whole and describes the role of each agent within the cooperative problem solving process, showing how the basic FA/C principles were employed for the application of traffic management, and what alternative solutions were considered. The major functions of each agent are described in this chapter. A more detailed description of the functions within the distributed algorithm for the search of control solutions, which is the task that benefits most from the application of the FA/C concepts, will be presented in the next chapter.

5.1 The Architecture

The current architecture is composed of two modules that exchange high-level, i.e., highly processed, information for the real-time analysis of traffic congestion phenomena and the formulation of appropriate integrated response plans. The two units, or agents, are real-time decision-support systems for Traffic Management Center operators: one agent is responsible for the operation of a freeway subnetwork, and is able to interact with an operator at the TMC of the freeway management agency. The other is responsible for the network of the adjacent surface streets, and interacts with the operator at the local city TMC. Each agent is an enhanced instance of TCM, the centralized module described in Chapter 3, suitably modified to provide inter-agent communication, cooperative reasoning and conflict resolution capabilities.

5.1.1 Input/Output

Figure 5.1 shows a schematic representation of the organization of the two agents. The current status of the network is accessible to each agent through a real-time connection that makes traffic data available to it in short intervals.

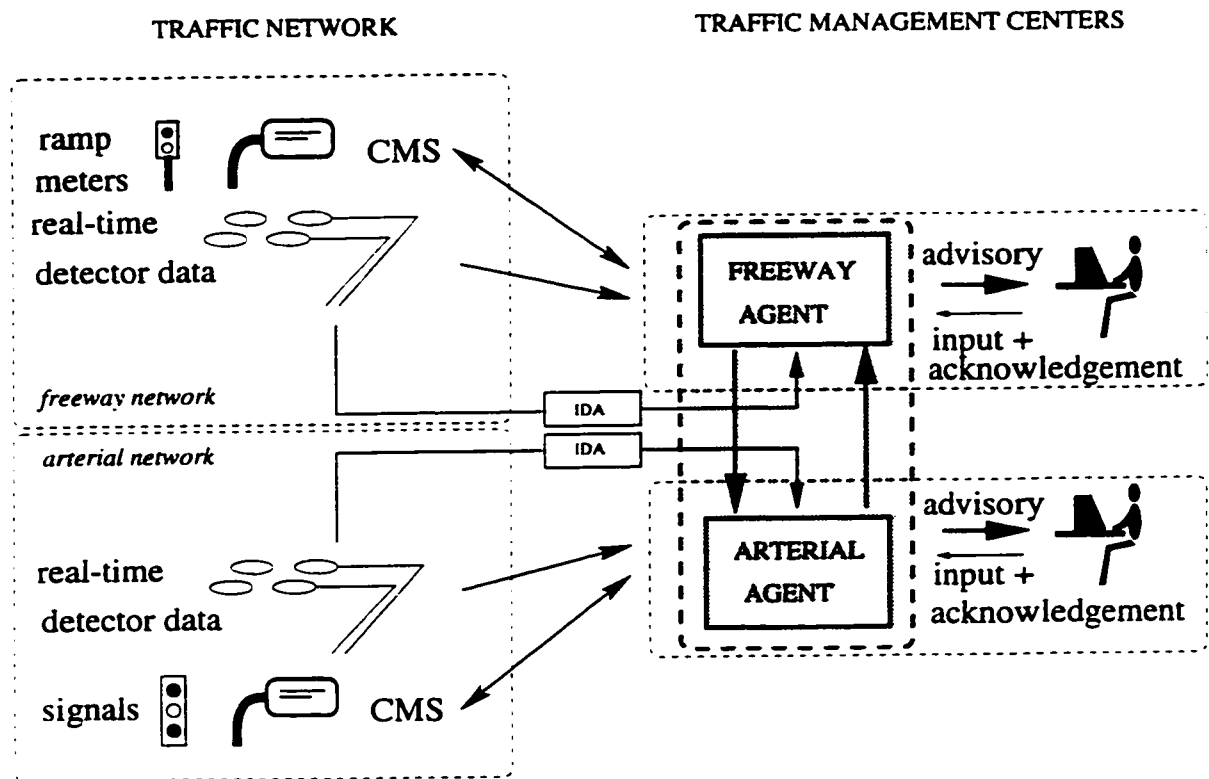


Figure 5.1: The distributed architecture

The agent's reasoning is, to a certain extent, independent of the exact format of these data, since it is always possible to define an interface between the agent and the medium providing the data. The agents are able to receive traffic data such as vehicle counts and occupancy rate as provided by loops detectors at regular intervals, such as 30 seconds (the interval used by Caltrans). The agents use these data sources, if and when they are available, as well as the results of a path-based network assignment, to assess the demand in certain parts of the network, and also to distinguish between different types of congestion. Traffic volume is expressed as the number of vehicles that pass over a point on a road segment (the pavement sensor) in a certain time interval. The 30-second input values are projected to provide the agent with hourly volumes. Volumes are aggregated

over all the lanes of a link. The occupancy rate is defined as the percentage of time during which a detector is covered by passing vehicles, and through appropriate adjustment, it is considered to provide a measure of the density of traffic on a link. The density is a measure of traffic conditions that expresses the number of vehicles that at any given time are traveling on a road section. It is a space measure, rather than a time measure, and it can be properly measured only by a snapshot of a highway section. Measuring the occupancy rate through a pavement detector is a commonly adopted methodology that attempts to provide an approximate indication on the density of traffic in the section surrounding the sensor.

A bidirectional real-time communication medium also exists between the agent and the control devices present within the network. Thus the agent receives the current control status from all the devices under the jurisdiction of the corresponding agency, and, at operator request, is able to transmit control directives for changes in the status of those devices. Again, the agent's functions are relatively independent from the exact format of this type of data, since an interface is defined between the agents and the devices. In the current implementation, signal and ramp meter plans are defined by a set of parameters that specify the plan cycle length, its offset, and for each phase, its minimum and maximum green, and unit extension, for use by traffic-actuated controllers. CMS messages are selected from a database of predefined plans, and are identified by an index in that database.

Each agent is able to receive external input about the occurrence of congestion, and respond to it. Input can be both acquired automatically, through a real-time connection with an incident detection algorithm, and accepted from TMC operators if additional information is available to them about the occurrence of congestion. The operator may be asked to provide additional information regarding congestion. For instance, if congestion is caused by a temporary lane closure, the operator is prompted to provide the estimated (or known) duration of such an event. If congestion occurs at a signalized intersection or at a freeway metered ramp as a result of inadequate signalization, the system prompts for an estimate of the current queue lengths, information that the operator may be able to provide by looking at CCTV cameras. A user-friendly graphical interface allows interaction between the agent and the operator by requesting acknowledgment of several of the agent's inference processes, by displaying detailed information about intermediate and final results and by providing communication between the operators at different TMCs.

The two agents communicate through a fast TCP/IP-based real-time protocol

(Black 1992), that allows them to exchange information. Compatible with the highly object-oriented organization of each agent's knowledge and inference capabilities, communication between the agents occurs by exchanging data structured as complex objects. The details of such exchange are introduced in the next sections.

5.1.2 Data Sharing and Agent Organization

The issues about the organization of the agents and the partition of data and control power among them constituted major decision points in the design process of the distributed architecture. Several options were taken into consideration, once the need for a distributed, multi-decision-maker approach was evident.

A possible design conceives a system composed of two agents interacting with an intermediate module, called the *coordinator*. Following this approach, each agent would interact with the operator of one of the TMCs, would be able to perform partial local processing on available data, and resort to the coordinator for remote data acquisition and network-level decisions. The functions and scope of the coordinator can be as simple as an *intelligent message passer* or as complex as a reasoning unit able to resolve conflicts among the agents.

An alternative agent organization may be conceived, in which the functions of coordination among the agents are transferred from the coordinating module to the agents themselves, by providing them with the ability to resolve inconsistencies through the definition and verification of inter-agent constraints, and to decide when it is necessary or convenient to interact with another agent and what type of information should be exchanged. Several reasons determined the selection of this approach. First, the need to have control decisions ratified by TMC operators and the lack, within the administrative organization of transportation management agencies, of an authority able to coordinate and potentially override control decisions made by either of the agents, limited the power of the coordinating module. Once the functions of the coordinating unit reduced to mere message passing and automatic (unmanned) decision-making, it was decided to allow the agents to share some information, still guaranteeing a sufficient level of autonomy, and introduce a certain degree of computation redundancy, in order to completely eliminate the need for a coordinator. Furthermore, this organization, even though it requires a more elaborate interaction scheme, is in agreement with most of the literature in distributed problem solving

and multiagent systems (Wittig 1992), in which agents are conceived as units at the same hierarchical level that contain interfaces that enable them to interact. Finally, on a strictly practical note, this scheme reduces the number of knowledge bases that must be developed, and therefore the number of real-time shells required for the software implementation of the system.

The issue of data sharing is closely tied to the agent organization. The database accessible to the agents can be:

centralized, i.e., data is collected and made available to a centralized unit that administers it and lets the agents have shared access to it;

distributed to all agents in such a way that all agents have private access to copies of the same data;

partitioned and local, in such a way that each agent has exclusive access to a portion of the data (for instance the one local to its own jurisdiction) and provides other modules with abstractions of the data that is considered relevant for the accomplishment of its tasks.

hybrid, i.e., some of the data is shared, or redundantly distributed, while the remaining is partitioned among private access databases.

The first two options require extensive data communications. In particular, the first one requires data to be assembled into a unique database that can be accessed by all agents, thus originating potential access delays and maintenance complications. The second requires maintenance of multiple copies of the data, thus calling for complex mechanisms to guarantee consistency and at the same time severely limiting the system adaptability. More importantly, the first two options preclude the interacting agencies to have reserved data access, thus interfering with their desire of relative autonomy and exclusive control of their jurisdictions. These solutions have the advantage that, directly or indirectly, each reasoning unit has access to complete and exact data, thus making the problem solving process easier to deal with, compared to the case in which the information available to each agent is not complete.

The third option allows each of the interacting agents to preserve dedicated control over its portion of data, by controlling the amount and the quality of the information that is made available to remote agents. Another important advantage of this option involves

the reduction of data processing that can be achieved, by having one agent process its data locally, and then making intermediate or final results of such processing available to the other agents. The apparent superiority of this approach is balanced by the complexity of the interaction between the agents. Given the potential lack of completely specified and globally accessible information, agents must have means for satisfying constraints, resolving inconsistencies, and developing a globally compatible and efficient solution.

The fourth option allows agents to share some of the input data, when the loss of autonomy associated with it is not considered significant, and a substantial reduction in the complexity of the constraint satisfaction process can be obtained. As is often the case for hybrid approaches, such a solution has the potential of combining the advantages and limiting the drawbacks of both approaches.

Whatever the partition of the input data (traffic data or control device status) among the interacting agencies, it is important that each agency maintain dedicated control over the devices under their jurisdiction.

The fourth option was deemed the most efficient, practical and elegant, based on the mentioned criteria. As introduced at the beginning of this chapter, input data describing the status of the network is made available to each agent through access to detector data on links that are part of the subnetwork controlled by the corresponding agency. A moderate redundancy is necessary for the agents to assess the status of the network at the boundaries between the freeway and the arterial network. For this reason, detector data on freeway ramps is made available to both agents.

Control related information is partitioned among the agents as follows. The agent responsible for the operation of the freeway subsection has authority over the CMS on the freeway and sets appropriate rates at freeway metered on-ramps. The arterial agent is responsible for posting CMS messages on surface streets and for the operation of intersection signals. The knowledge used to determine control plans is partitioned as follows: inference methods and data describing the control status of signalized intersections are local to the arterial agent, while the corresponding knowledge and data for selecting an appropriate metering rate at freeway ramp meters are local to the freeway agent. Data related to CMS are treated in a different way: in order to guarantee consistent traveler information, messages are stored in groups that include settings for CMS both on the freeway and on surface streets. Groups of messages that initiate traffic diversion from the freeway are part of the knowledge of the freeway agent, while those that initiate traffic diversion from

surface streets are part of the knowledge of the arterial agent. This is consistent with real-life scenarios, for example in California, where often Caltrans, the agency responsible for freeway operation is aware of the possible messages that the local City TMC can use, and vice versa. Nonetheless, each agency has exclusive authority over the CMS within the network under its jurisdiction.

5.1.3 The Rationales for a Functionally Accurate, Cooperative Approach

When analyzing the behavior of an agent it is important to remember the basic reasons for the chosen distributed approach and the main goals of the distributed architecture: agents interact in order to produce an efficient and conflict-free global solution in an environment that is inherently distributed. Their interaction is constrained by the lack of complete information, that results from the data and control knowledge being distributed according to geographical and administrative criteria. Input data available to the TMC operator, such as detector data describing traffic conditions, visual data coming from CCTV cameras installed at key locations, and incident reports or confirmations, are normally partitioned according to the institutional organization of the management agencies, which must administer the information in their possession as efficiently as possible, avoiding the transmission of irrelevant or ill-timed data. Another constraining factor is the need of each agency to preserve dedicated control over its jurisdiction, maintaining its decisional power, and applying local criteria and guidelines. At the same time, the agencies, as interacting components of a regional, integrated management organization, must attempt to converge towards a common goal, that of ameliorating network-wide traffic conditions, and can do so by effective collaboration and resolution of potential conflicts.

A mechanism that provides the means for effective cooperation by allowing interacting agents to coordinate their efforts and resolve their conflicts, and that takes maximum advantage of existing infrastructures and technology, is required. As seen in Chapter 4, recent developments in the theory and practice in Distributed Problem Solving methodologies have the potential to provide the required capabilities. The ability to use a suitably enhanced version of an already developed knowledge-based system, as the basic congestion management model, and to develop a system composed of agents that share a common language and knowledge representation formalisms, makes the use of a DPS approach particularly attractive.

In particular, several characteristics of the problem of real-time traffic control in response to non-recurring congestion in a distributed environment, and their affinity with those of the domains traditionally addressed by the FA/C problem-solving style, described in the previous chapter, provide the rationale for an exploration of the benefits of the application of those concepts to the problem at hand.

As shown in Chapter 4, the idea of providing interacting agents with the ability to cooperatively solve tasks, using only limited and uncertain knowledge of the processing performed and the results obtained by other agents, was developed in response to what were perceived as deficiencies of traditional DPS approaches. These focused on maintaining correctness and completeness in all aspects of the distributed computation, to guarantee each agent sufficient data to solve its task completely and accurately. For many applications, decomposing a problem into a set of tasks that ensures a perfect match between the location of information, expertise, processing and communication capabilities, and the computational needs for effectively solving each subproblem, is not possible. The maintenance of accurate, complete, and up-to-date information requires heavy communication of input data and frequent exchange of intermediate processing results, thus burdening agents with performance reduction due to communication and synchronization delays, that may results in resolution times that are not acceptable for real-time applications.

Distributed traffic control in a complex urban network is one such application. Providing each task solver with complete data would require not only the transmission of all input data to all agents, but also sharing the expertise required to address arising problems. Input data (coming from sensors, videocameras, police reports, automatic incident detection algorithms, etc.) can be exchanged among several agencies, and in fact, as seen in Chapter 2, some deployment projects make use of complex data sharing links, that allow each of the agencies involved to have access to such data. But, beside the fact that such networks have very high installation and maintenance costs¹, input data covers only one aspect of the requirements for maintaining processing completeness. The impracticability of sharing expertise and decision-making power in a real-time context limits the flexibility of those applications, by requiring the adoption of predefined, previously established cooperation plans.

The application of the FA/C concepts to distributed control allows one to take ad-

¹the total cost of the Santa Monica Smart Corridor project amounted in 1997 to \$48 million. See Chapter 2

vantage of local data processing performed by cooperating agents, while limiting the amount and increasing the informative quality of data exchanged, by performing approximate or incomplete processing on raw data and transmitting, when possible, highly informative abstractions of such data, expressed as local solutions, intermediate goals, and inter-agent constraints that must be satisfied.

There are analogies between the problems addressed by FA/C problem solving, namely distributed SI and resource allocation, and real-time traffic congestion management and control. An SI solution is a composite of a set of hypotheses of different types, each of which explains a certain portion of the input data. An agent can derive it by processing local data, and validate, adapt, or reject it, based on constraints and additional information that arise from hypotheses derived by remote agents after processing other portions of the input data. In general, there can be multiple possible interpretations of the same data, and a "good" solution is one that offers a "good" explanation (or interpretation) of the data.

In an analogous way, solving a traffic control problem involves determining adequate settings for control devices that are aimed at ameliorating current traffic conditions on the network. As mentioned in Chapter 1, in oversaturated conditions it becomes necessary to balance the demand/capacity distribution for a smoother traffic flow, by increasing segment capacity via signalization, and adjusting the flow distribution by information and advice to motorists. In this respect, a control solution is a composite of settings for several types of control devices present on the network, that affect the capacity and the flow of certain links. Settings for control devices are inter-related, and while an agent may be able to determine an appropriate control plan for a certain portion of the network, additional information on local conditions and on the control being applied onto another section may generate and propagate constraints that in general can not be ignored. In general, multiple alternative composite control plans can be developed, and the goal is to construct them and assess, with a reasonable degree of confidence, the benefits that can be expected by their implementation.

Both problem domains have to deal with a significant amount of uncertainty, generated by the potential inaccuracy of the input data (sensor errors) and the lack of complete knowledge of the causal relationships between events (data and hypotheses in SI, control plans and response in traffic control). This, in conjunction with the virtually unlimited magnitude of the solution space justifies the adoption of approximation techniques and the research for *satisficing* rather than *optimal* solutions.

Finally, the concept of near monotonicity, described in the previous chapter as a characteristic of SI domains can be tailored to the control problem, as shown in section 5.3.5, providing some highlights on the effectiveness of the solution approach adopted in this work.

There is also a correspondence between some of the methodologies employed in FA/C-based applications and the basic structure of the solution process within TCM, the previously developed centralized module that serves as the building block for the design of the agents. When the reasoning process in TCM had to be adapted to support FA/C-based problem solving, TCM's highly modular and fairly flexible task-solving structure facilitated the adaptation of those mechanisms and their incorporation within the distributed system. Some of those mechanisms, as described in the following sections, require the ability to identify subgoals and develop dynamic basic goal structures, to distinguish between different forms of interaction, both within and among agents (analogous to SOUs and GSOU's), and the capability to balance redundancy of computation with problem complexity and experiment with different levels of control uncertainty.

The next three sections describe the main functions of the two agents. Since each agent is an enhanced version of the centralized module TCM, described in Chapter 3, the presentation follows TCM's original hierarchical knowledge-modeling structure with the description of the functions related to the analysis of traffic congestion described in section 5.2, to the formulation of control solutions in section 5.3, and to the monitoring of congestion dynamics in section 5.4. The presentation will describe only the innovative aspects, with respect to the original TCM centralized module. The distributed algorithm for the formulation of control solutions is the most complex function of the agents, and the one that has required the most extensive changes. Section 5.3 will focus on the main characteristics of the algorithm and on how the FA/C approach has been interpreted and applied for its development.

5.2 Problem Analysis

As noted earlier, the formulation of an appropriate control solution to the problem of a sudden onset of congestion, requires a clear understanding of its characteristics, namely, its location, its causes, and the impact that it is expected to have on the regular flow of traffic, both at local and at network-wide levels. The analysis of congestion is performed following the methodology adopted by the original TCM. The modifications that were

introduced, both to remedy to some of its shortcomings, highlighted by its initial evaluation, and to adapt the system to a distributed reasoning, will be described below.

The frame matching procedure used for the diagnosis of the problem type, and the identification of a critical section can still be successfully performed by each single agent on the sub-network under its control. An issue that was not taken in consideration in the original TCM, concerns the concept of a *derived problem*, which assumes particular importance in the case of distributed control. A derived (or secondary) problem (Longley 1968) is the occurrence of congestion associated with the propagation of the effects of an initial problem, to a different section of the network. In the original TCM, this was not considered a critical issue, since it was assumed that when the original congestion dissipates, the derived problem will also clear. In reality, as was shown by the simulation-based evaluation performed on the original TCM (see Chapter 3), there are cases in which it may be useful to treat the primary and secondary onsets of congestion separately, and try and address both problems. Especially when no control action can be implemented to mitigate the primary congestion, it is imperative to try and reduce its spreading across the network. As part of the recovery process, it is important to identify a secondary congestion, and this is especially critical in a distributed environment, where, given the lack of complete information, a secondary type of congestion that spreads to the network controlled by the remote agent might be wrongly classified and erroneously treated. Indeed, congestion can evolve either within one single agent's subnetwork or across the border between the subnetworks. Examples of derived problems include:

- freeway off-ramp bottlenecks: if a freeway off-ramp becomes congested, slow traffic conditions may propagate to the upstream mainline freeway section, thus affecting traffic traveling along the freeway. The corresponding primary congestion may be caused by an accident, by signal timing problems or limitations, if the ramp terminates at a signalized intersection or if it merges with a high-volume highway section, as shown in Figure 5.2.
- insufficient capacity of the downstream link(s) at a signalized intersection: if a downstream link of an intersection is congested, one or more of the upstream links may be affected, since there may be no physical space for traffic to move forward (a *de facto red* condition). The primary congestion on the downstream link, again, may originate from an accident downstream, or, as it often occurs in densely signalized

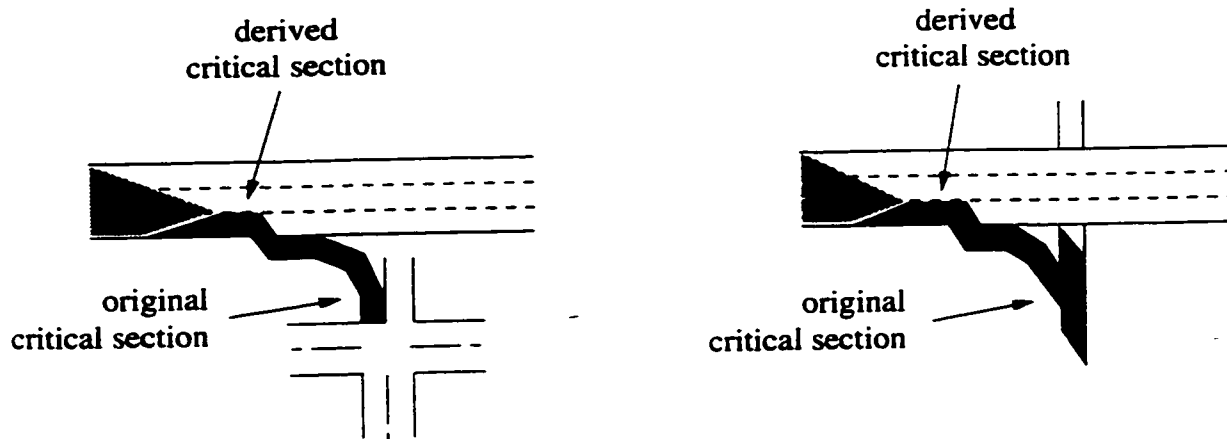


Figure 5.2: Spillback at off-ramps

urban networks characterized by short links, from the downstream intersection being already saturated. If the downstream link leads to a freeway on-ramp, the congested ramp may be responsible for the initial congestion. A schematic example is shown in Figure 5.3.

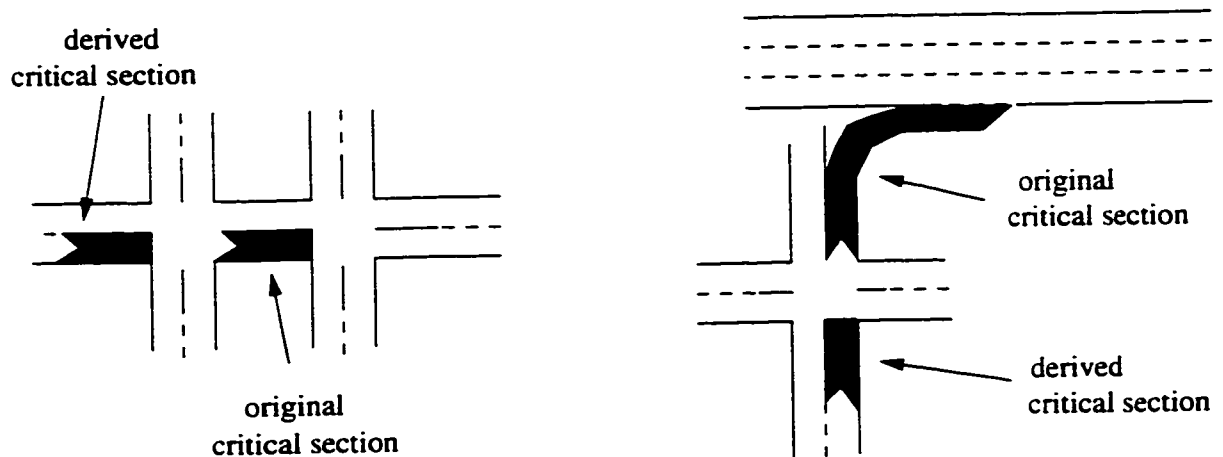


Figure 5.3: Spillback at signalized intersections

The analysis of secondary congestion requires one to identify the problem and to determine its characteristics that, in general, are related to those of the corresponding primary congestion. The problem identification is carried out during the Problem Diagnosis phase, through the definition of three additional problem frames (see Section 3.1.2), that describe the two cases above (a spillback at a signalized intersection requires two frames,

since the congested downstream link may coincide with either the left-turn or the through movement). The simplicity of this adaptation incidentally confirms the flexibility of the modeling methodology. In order to determine the characteristics of the secondary congestion, the algorithm tries to match the *derived* problem with the one corresponding to the primary congestion. If both problems occur within the subnetwork controlled by the same agent (e.g., at two consecutive signalized intersections) then, once the original problem has been analyzed by the system, the relevant attributes of the secondary problem can be inferred with no need for agent interaction. Otherwise, the two agents need to communicate their findings, in order to determine a relationship among congestion phenomena. In the terminology adopted by the developed software system, a secondary type of congestion corresponds to a *spillback* problem type. A *local* problem is one occurring on the agent's sub-network, while a *remote* one occurs on the other agent's sub-network.

To maximize the effectiveness and limit the amount of the interaction among the agents and allowing operators to acknowledge the occurrence of the problems within the network under their control, the analysis of the relationship among problems and their description involves the following procedures:

Intra Agent Relate Derived Problems: this procedure determines a relationship between original and derived problems, within the list of problems that are local to the agent. At the time this procedure is executed, problems have not yet been fully analyzed, i.e., the corresponding critical section demand, capacity, and the problem's expected duration have not yet been assessed, so the only known attributes are the problem's location and type. The analysis of the relationship among problems is performed by verifying, for each spillback, if any of the other local problems can be its original problem, by looking at their location and differentiating over their type. The procedure receives in input the lists of local current and previous problem descriptions, respectively *CP* and *PP*, and returns a *derives_from* relation between problems. The procedure is defined as follows:

Algorithm 5.1 (Intra Agent Relate Derived Problems)

```

1   begin
2       for  $p \in CP$  s.t.  $problem\_type(p) = SPILLBACK$  do
3           for  $q \in \{CP \cup PP\}$  s.t.  $q \neq p$  do
```

```

4      DERIVE = verify_derivation(p, q);
5      if DERIVE then derives_from(p) = q
6          fi;
7      od;
8  od;
9  end

```

where *verify_derivation*(*p*, *q*) is defined by the following logic function:

$$\begin{aligned}
 & (\text{problem_at_r_imp}(p) \wedge \text{ram}_r(p) \in \text{cs}(q)) \vee \\
 & (\text{problem_at_intersection}(p) \wedge \\
 & \quad ((\text{cs}(p) = \text{left_mov} \wedge \text{left_link}(p) \in \text{cs}(q)) \vee \\
 & \quad (\text{cs}(p) = \text{through_mov} \wedge \text{through_link}(p) \in \text{cs}(q))))
 \end{aligned}$$

Inter Agent Relate Derived Problems: this procedure performs a similar task to the previous one, but it tries to establish a relationship between a local spillback and remote problems. The task is more complex because each agent does not yet have information about remote problems. For each local spillback, an agent invokes a remote procedure call (RPC) (Birrel & Nelson 1984) to the other agent, passing the characteristics of the spillback as arguments and receiving as output some of the identifying characteristics of the problem that is found to be the corresponding primary congestion. The remote procedure uses the same internal procedures as the ones used by Intra Agent Relate Derived Problems. The procedure receives in input the lists of local current problem descriptions, *CP*, and returns a *derives_from* relation between local and remote problems. It is defined as follows:

Algorithm 5.2 (Inter Agent Relate Derived Problems)

```

1  begin
2      for p ∈ CP s.t. problem_type(p) = SPILLBACK do
3          q = RPC(compute_derivation, p);
4          if q ≠ ∅ then derives_from(p) = q;
5          fi;
6      od;
7  end

```

where $compute_derivation(p)$, executed by the remote agent, verifies the relation between p and its local (remote for the first agent) problems $\{CP \cup PP\}$, as follows:

$$compute_derivation(p) = \begin{cases} q & \text{if } \exists q \in \{CP \cup PP\} \\ & \text{s.t. } verify_derivation(p, q) \\ \emptyset & \text{otherwise} \end{cases}$$

Describe Problems: after having derived the proper relationships among problems, the attributes of each local problem, such as demand, capacity, estimated duration, etc., are determined, in the same way as in the centralized TCM. The only problems whose attributes cannot be determined using only local information, are spillbacks, because both their capacity and their expected duration may depend on attributes of the corresponding original problems. This task is performed by the next procedure.

Import Remote Problems: each agent executes this procedure to *import* the description of the problems detected and analyzed by the remote agent. The description of remote problems is used both for updating the attributes of local spillbacks deriving from remote problems and to provide each agent with a view of the global network status, so that both agents can cooperatively tackle both local and remote problems. The capacity of a secondary type of congestion is computed as follows:

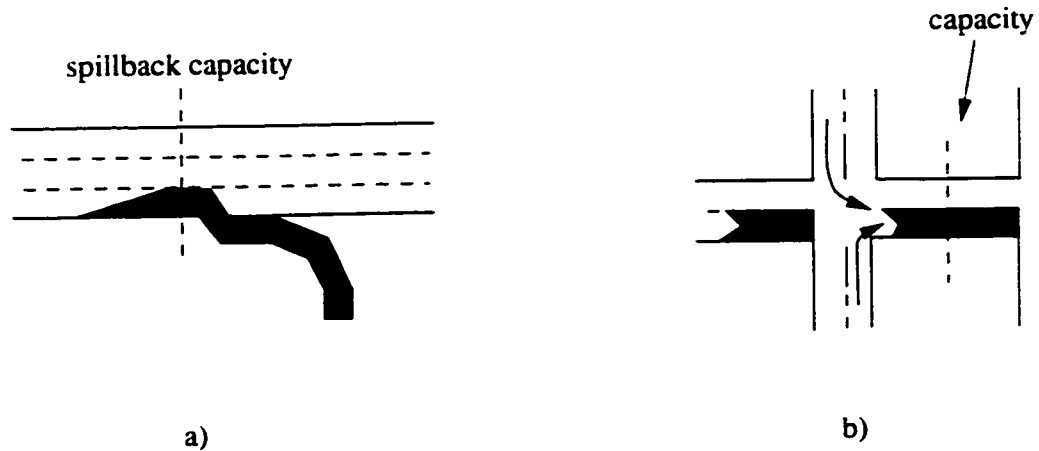


Figure 5.4: The capacity of a secondary congestion

- if the spillback is a bottleneck at a freeway off-ramp, the capacity of the upstream link is estimated by applying the same criteria used for a lane-blocking accident,

assuming that the number of closed lanes coincides with the number of lanes of the off-ramp (Figure 5.4 a)).

- if the spillback is at a signalized intersection, the capacity of the congested upstream link is estimated as the amount of traffic, if any, that can flow through the congested downstream link, and is computed as the difference between the capacity of the downstream link and the flow directed from the remaining approaches to the congested downstream link (Figure 5.4 b)).

A derived problem can be interpreted as an incomplete (partial) classification of a problem, since it is not immediately known what the corresponding original problem is and what its characteristics are. Communication in this case helps completing the analysis of the problem's attributes.

At the end of this phase, the agents exchange information with each other, describing their *local* problems, that have been, by now, fully analyzed, for which a solution has to be determined. This step allows each agent to become aware of problems detected by the other agent, that may in general affect its search for a solution, and that the agent can help solve.

5.3 Problem Response

After the current problems have been analyzed, the agents start a complex process for the formulation of suitable control solutions.

As already noted, a functional formulation of the solution to the problem of network-wide traffic management does not exist. In general, the ultimate goal is to ameliorate overall traffic conditions, and from a global perspective, this can be achieved by a reduction in travel time and delay. How to achieve such goals under congested conditions is not clear and requires further research. This is due to several sources of uncertainty, inherent in traffic modeling and control: if it were possible to measure the exact travel time of each vehicle and if the effect of control plans on travel times and delay could be correctly computed, then it would make sense to adopt total or average travel time as the objective function and look for control plans that minimize them. Unfortunately, the exact travel time can be measured only in simulated environments. Also, the flow distribution across the network can only be partially represented, and the effect of control and traveler infor-

mation on traffic can be at best only estimated. At the same time, several valid algorithms for modeling the traffic behavior and the effect of control actions have been developed. In general, the complexity of these algorithms grows exponentially with the dimension of the input data, thus only problems of limited size can be correctly addressed. A frequently used solution lies in the adoption of assumptions and the use of heuristic techniques that limit the size of the input data, and in the application of those algorithms only in restricted areas. Such a compromise obviously yields a solution that is at best an approximation of the optimal one, but if the assumptions are realistic and the heuristic techniques adequate, the obtained solution may still be a satisfying one.

As in the case of distributed sensor interpretation, other sources of uncertainty are due to the distributed nature of the problem and the solution. As seen at the beginning of this chapter, agents have only a partial view of the network status, because they receive real-time traffic data and incident information according to criteria based on the existing administrative and geographical distribution of responsibilities. Also, due to the partitioning of human and automatic problem solving expertise among the various management centers, agents have only partial knowledge of how to solve problems. A further source of complexity is due to the distribution of control and the division of authority over control devices.

All these considerations highlight the need to select an overall objective that can be evaluated and a methodology that provides indications on how to reach that objective. Under congested conditions, a reasonable goal is the avoidance of the spreading of congestion, that otherwise may lead to oversaturation (Longley 1968), and the achievement of a balanced ratio between network capacity and traffic demand.

5.3.1 The Problem Response Algorithm

A decomposition of the original problem into subtasks in general allows partial isolation of subproblems in order to make them more treatable by specific task-solving algorithms. The subtask isolation is only a partial one, because in complex problem domains, many of the subtasks interact with each other, either through the resources used by the task-solvers, or through the constraints they generate. An appropriate algorithm must be able to identify these interactions and efficiently deal with them. For example, in traffic control, the selection of a traffic diversion plan aimed at reducing the flow directed towards

a congested location may affect the control implemented on the set of routes that receive the diverted traffic; the selection of a particular signal plan may also cause spillback that changes the effective capacity of the upstream links and intersections.

In order to reduce the amount of inter-agent communication and synchronization, each agent initially processes only local data, that is directly available, and abstract information received from the other agent. An agent is able to recognize when it is necessary or convenient to interact with the other agent, and transmits or requests information that can guide the processing performed by the other agent. Analogously to the *multistage negotiation* protocol described in Chapter 4, a point is reached where each agent has processed all or a sufficient portion of its local information, and partial results are exchanged and combined to determine compatible and efficient global solutions.

The algorithm, called DTPS (Distributed TPS) is composed of the following two phases:

1. *Construction of Local Solutions:*

This step involves determining a set of solutions composed of control actions local to the agent, starting from an abstract representation of the network conditions, through a path-based estimation of the current network demand and information describing the current "problems", i.e., the location, the type, and the severity of the congestion phenomena that must be addressed. A control action local to the agent is a list of settings for control devices that are under the jurisdiction of the agency corresponding to the agent.

The construction of solutions is performed by dynamically defining opportune strategies that prescribe different, complementary or alternative, ways to achieve corresponding subgoals. Typical strategies (or the corresponding subgoals) are for example, *increase capacity at congested location by signalization*, or *decrease flow at congested location by diversion*. For each strategy the algorithm looks whether suitable settings for control devices can be determined that are expected to satisfy, partially or completely, the corresponding goal. In general, a strategy may require the modification of the status of more than one control device. Also, depending on the type of the strategy, more than one alternative control action may be constructed that "translates" it. The selection of a strategy or the construction of a control action may require the satisfaction of particular conditions, that define dynamic constraints that propa-

gate among the agents. The satisfaction of those constraints determines the definition of additional subgoals, or the modification of existing ones, thus modifying the goal structure of the agents, both at local and remote level, as part of a distributed search process similar to the example presented in Section 4.2.2.

Adequate control settings are obtained by applying specific algorithms, that will be presented in detail in the next chapter. For instance, to determine the appropriate signal plan at a saturated congestion, a queue-management algorithm, based on Gazis's (1974) theory is employed. For each control action, specific algorithms, based on previous established research on delay (Fambro & Rouphail 1997) and on original heuristic techniques, are used to estimate how the corresponding control is expected to reduce the current problem and in general affect the network conditions. For instance, the selection of a certain signal plan for a specific time interval, by increasing the capacity of a congested approach at an intersection and decreasing the capacity of an underused one, is expected to reduce, in time, the length of a queue, thus reducing the overall expected intersection delay.

During this stage each agent is able to establish the compatibility of control actions only at the local level, since the information necessary to verify global compatibility may not yet be available. During the processing, however, additional information is stored with the local solutions, to facilitate the future verification of global compatibility criteria, which will be performed in the next stage.

As in the centralized algorithm, a search process is performed that involves the construction and traversal of a heterogeneous search tree (Rich & Knight 1991), as shown in Figure 5.5. Each node of the tree represents a problem state, and each arc represents a relation between states, and corresponds either to a possible strategy (or the corresponding subgoal) or to a control action that processes a subgoal. The root corresponds to the initial state, i.e., to the problems to be solved. A path from the root to a node represents a possible local solution whose corresponding control actions are determined by the arcs composing the path. The search is an iterative process that each agent executes, composed of two nested loops. The internal loop is the construction and traversal of the local search tree. The external loop is used to check, at the end of every iteration of the internal loop, whether any new information has been received by the other agent. If new information is received, the visit of the current

tree starts over, because new subgoals may be defined by the additional data.

Such an organization is due to the relative asynchrony of the two agents, aimed at letting an agent process its data without having to stop during the search process, but, at the same time, providing each agent with the opportunity to update its goal structure if external information is received. The internal loop proceeds by sequentially visiting nodes. For each node, the current problem status determines the specification of suitable subgoals and correspondingly, of applicable strategies. For each strategy, all feasible control actions that may implement it, are developed, and a new node is constructed, that describes the expected problem status after the implementation of the action.

Criteria for the termination of the local search have to take into account several factors: it may not be necessary to consider all possible solutions, but the more extensive the search is, the more likely it is to find a better solution; and the more complex the process becomes. Analogously to the definition of acceptance thresholds in the LCRS, as described in Chapter 4, thresholds can be defined to terminate the search and combine the necessity to produce an acceptable solution in a short time (due to the real-time nature of the problem), with the search for a truly effective solution.

At the end of this phase, each agent has determined a list of local solutions, ranked according to the effect that each of them is expected to have on the initial problem. The solutions are computed based only on locally accessible data and information received from the other agent. Each solution stores information that will be used, within the next stage, to establish its consistency and compatibility with the partial solutions constructed by the other agent, for the construction of global solutions. The local search procedure is described by Algorithm 5.3.

Algorithm 5.3 (Local Search)

```
1      begin
2           $N_0 = \langle PS_0, \{\}, H_0, 0 \rangle;$ 
3           $RS = import\_remote\_strategies();$ 
4          while  $\langle TRUE \rangle$ 
5               $unvisit\_all\_nodes(N_0);$ 
```

```

6         curr_node_ptr =  $N_0$ ;
7         while < cur_node_ptr  $\neq \emptyset$  > do
8              $N = \text{cur\_node\_ptr}$ ;
9             if  $\neg \text{visited}(N)$ 
10                then
11                     $\text{visited}(N) = \text{TRUE}$ ;
12                     $\text{expand\_node}(N)$ ;
13                    if  $\text{left\_child}(N) \neq \emptyset$ 
14                        then  $\text{cur\_node\_ptr} = \text{left\_child}(N)$ 
15                        else  $\text{cur\_node\_ptr} = \text{sibling}(N)$ ;
16                    fi;
17                else  $\text{cur\_node\_ptr} = \text{sibling}(N)$ ;
18            fi;
19        od;
20         $RS = \text{import\_remote\_strategies}()$ ;
21        if  $RS = \emptyset \vee < \text{termination\_condition} >$  then  $\text{exit}()$ ; fi;
22    od;
23 end

```

The *expand_node* subroutine is essentially the same as the one described in Algorithm 3.2, except that after the creation of a new node M and its insertion in the expansion subtree, the routine described by Algorithm *refcondhandler* is executed on node M , control action CA and strategy S , to handle the generation of conditions and the translation of conditional strategies.

Algorithm 5.4 (Condition Handler)

```

1     begin
2         if  $(C = \text{require\_condition}(CA)) \neq \emptyset$ 
3             then  $\text{push}(C, \text{required\_local\_conditions}(M))$ ; fi;
4         if  $\text{conditional\_strategy}(S)$ 
5             then
6                  $C = \text{create\_condition}(S)$ ;
7                  $\text{broadcast\_condition}(C)$ ;

```

```

8      fi;
9      end

```

where $require_condition(CA)$ returns a condition C if control action CA requires the verification of a condition, and the empty set otherwise.

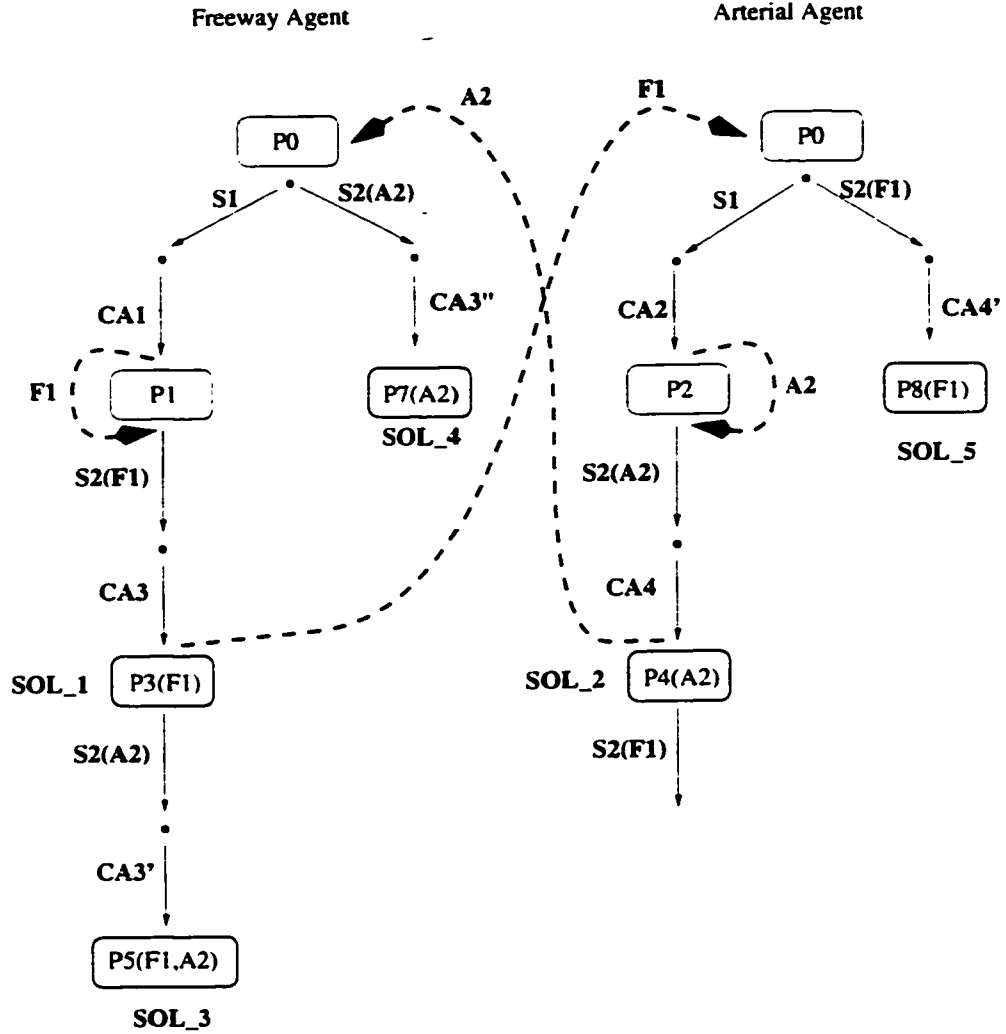


Figure 5.5: The search trees in the distributed problem-solving process

In the example depicted in Figure 5.5, the two agents must solve a problem $P0$. Both agents initially determine strategy $S1$, which is translated into control actions $CA1$ by the freeway agent and $CA2$ by the arterial one. The selection of $CA1$ and $CA2$ requires the verification of conditions, respectively $F1$ and $A2$. Each condition

requires the achievement of a goal, which is, in turn, translated into a corresponding strategy ($S2(F1)$ and $S2(A2)$). The freeway agent determines a control action $CA3$ that translates $S2$, while the arterial agent determines control action $CA4$. Now, since both conditions $F1$ and $A2$ are locally satisfied, they are exchanged among the agents, at the end of the internal control loop. Since no other strategy or control action can be determined based only on local processing, the two agents attempt to satisfy the received remote conditions. The freeway agent translates condition $A2$ into strategy $S2(A2)$ and, applying it to the nodes of its search tree, selects control actions $CA3'$ and $CA3''$. The arterial agent translates condition $F1$ into strategy $S2(F1)$ and, applying it to the nodes of its search tree, it can find only one node for which the condition is verified, determining $CA4'$.

At this point, since the search trees can not be expanded any further, this phase terminates. The freeway agent has three solutions $SOL1$, $SOL3$, and $SOL4$. The first solution satisfies only local condition $F1$, the second satisfies both local condition $F1$ and remote condition $A2$, while the third one only satisfies remote condition $A2$. The arterial agent has only two solution, $SOL2$ and $SOL5$, which satisfy, respectively, local condition $A2$ and remote condition $F1$.

2. Construction of Global Solutions:

This step involves the construction of consistent and efficient global solutions, and their ranking according to effectiveness criteria, before they are presented to the operators for final selection.

Starting from the two sets of local solutions developed in the previous stage, it is necessary to determine a limited set of pairwise matchings that are mutually compatible and consistent, and assess their effectiveness, i.e., the benefits that can be expected by their implementation. The complexity of this process stems from the following considerations:

- Depending on the number of local solutions, and on their degree of mutual compatibility, the number of possible combinations of pairs of them may become very high.
- As seen in Chapter 4, strictly speaking, every distributed problem is non-monotonic, therefore the best global solution is not necessarily the composition of the two

best local ones, even assuming they are compatible. In traffic control, this is due in part to the non-linearity of the laws that govern traffic flow and in part to the approximation embodied in the techniques used for assessing the quality of local solutions.

- The number of solutions proposed to the operators can not be too large, otherwise too long a time may be required to analyze all of them.
- At the same time, a proper decision support tool must be able to propose more than just one feasible solution, to give the operator a certain degree of freedom in the selection of a solution, especially in a distributed environment, where not one, but two operators must agree on the final solution.

Respecting these considerations calls for the definition of suitable criteria for global consistency, for mutual compatibility, and for global effectiveness, that will be discussed in the next subsections.

At this level it was judged appropriate to introduce a certain degree of redundancy. Thus, the agents mutually exchange their sets of local solutions and both of them perform the same procedure on copies of the same data to construct an initial set of global solutions. Since all the information necessary to establish global compatibility is now contained in the local solutions, the composition of compatible global solutions is a rather straightforward process that is performed without the need for additional data exchange. The computational overhead induced by executing the same operation on the same data is not considered particularly burdensome. Additional information exchange, instead, is needed for the assessment of the quality of global solutions because local data, such as the current traffic volumes, are necessary.

While the verification of global compatibility criteria and solution composition is a simple process, the evaluation of solution efficiency is not. Reflecting the above considerations, the goal here is to consider as many solutions as possible, and at the same time use the available information to reduce the computation necessary to select a limited number of highly effective ones. The tradeoff is, once again, between accuracy and computability: the more extensive is the computation, the more informed is the decision making, and accurate the solution, but the more time-consuming is the algorithm.

This stage is carried out by a sequence of steps aimed at progressively reducing the search space of the solutions that must be analyzed in the following steps, using as efficiently as possible the available information. At each step the accuracy of the evaluation criteria is higher than that of previous steps. The algorithm can be summarized as follows:

- (a) Each agent while transmitting the set of its local solutions, reduces it, by eliminating local solutions that by the end of the local search are not locally consistent.
- (b) Each agent receives the reduced set of solutions local to the other agents and looks for pairwise mutually compatible matches of local and remote components. This step entails verifying whether two solutions are mutually consistent and they do not recommend different setting for the same control device.
- (c) The set of global solutions is further reduced, by eliminating solutions that according to currently available information are considered redundant. The criteria in this case is based on the value of a heuristic benefit function that measures the expected capacity/demand balance associated with the solution, and on the corresponding list of control actions.
- (d) For each remaining global solution, a macroscopic assessment of their expected delay reduction is computed. The network delay associated with a certain control is compared to the one corresponding to the *status quo*. The network delay computation is partitioned among the agents, depending on the location of congestion and the locality of the control actions.
- (e) For each global solution, the total delay and total delay reduction are obtained by summing the delays computed for that solution, by the two agents.
- (f) Each agent orders the list of global solutions according to the associated expected delay reduction.
- (g) A limited number of the most highly-ranked solutions is selected for simulation. Short, faster-than-real-time simulations of the network performance associated with those solutions are executed to verify which one of them is more likely to provide the greatest improvement. Among the simulated scenarios, the one with the currently implemented control is also considered, to make sure that the evaluation considers also the possibility of implementing no change in the current

control.

An interesting issue in this context, that can be further explored concerns termination criteria, i.e., the criteria according to which the search for solution is concluded. The current termination criterion is the simple exhaustion of possible strategies and control actions, which provides, at least in a simulated environment, a satisfying performance in terms of execution time. In order to accelerate the response process, it might be worth considering alternative criteria, such as terminating the search after the search tree reaches a certain dimension, or depth, or after the projected delay reduction is considered satisfactory. The increase of resolution speed in this case may be counterbalanced by a loss of performance, if possible solutions are overlooked. It is interesting to experiment on the potential solution deterioration associated with a reduction of the search process. An empirical analysis can be performed, by executing simulations and observe the quality of the solution obtained adopting different termination criteria.

5.3.2 Strategies

As pointed out earlier, in the DTPS algorithm, there is a one-to-one correspondence between strategies and subgoals. Having recognized the difficulty of formally defining a suitable global objective function, the problem-solving process is thus driven by the definition of subgoals, aimed at reducing the future demand at specific locations and adapting local and remote capacity, in the attempt to adjust the network capacity/demand distribution. Specific strategies are used to achieve such goals. Strategies are defined as parametric objects that are instances of a hierarchy of predefined classes, and are dynamically created through the assignment of values to particular attributes. Each strategy stores a certain amount of information that varies according to the type (class) of the strategy, but that, in general, can be roughly classified as information describing the problem and the desired subgoal, and meta-level information, used to direct the problem-solving process. Whenever, during the problem-solving process, a strategy is constructed, its attributes are filled with an abstract representation of the location affected by congestion (*critical section*), a list of the paths that carry significant portions of the demand through the critical section, a description of the expected remaining capacity/demand imbalance, that will be obtained by the implementation of the control actions selected so far in the search process, and a *priority index*. The latter two pieces of information determine a rating of subgoals and

drive the task-solving process by specifying priorities among possible strategies and providing evidence on the current status of the solution process. Depending on the class it belongs to, a strategy may contain additional information. For example, a strategy that regulates intersection capacity requires information about the existing queue length at the intersection.

Currently the following strategies, and corresponding methodologies to implement them have been defined. Their class hierarchy is shown in Figure 5.6:

Decrease flow through traffic diversion: the corresponding goal is to divert traffic upstream of one or more congested locations to alternative, non-congested routes. This can be achieved by means of CMS located upstream of the congested area along major paths and on alternative routes. It obviously requires the availability of feasible alternative routes, in which the average travel time, even considering the additional load resulting from the diversion, is low enough to justify the detour. It may require the adjustment of signalization along the alternative path to provide additional capacity for the diverted traffic. This strategy is justified only for major disruptions, such as accidents during high-volume conditions.

Decrease flow through upstream metering: the corresponding goal is once again to reduce traffic flowing through a congested section, but it is achieved by reducing the capacity upstream of a congestion, according to well-known, but still under discussion, theories of traffic metering. The implementation of the strategy is currently limited at freeway metered ramps, but the principle can be adapted to intersection signal control for the implementation of *gating* control (Quinn 1992).

Increase capacity through signalization: here the goal is to increase the capacity of a controlled junction, by modifying the current control plan, compatibly with the local demand at the various approaches and, if necessary, with network-wide consideration. Controlled junctions include signalized intersections, operated by the arterial agent, and freeway metered ramps (either freeway-to-freeway or, more frequently, arterial-to-freeway). Important factors, for deciding the appropriate control, are the expected flow, the measured volumes, and the existing queues which the strategy is trying to reduce. Information about the expected flow is necessary for several reasons: first, by providing path-based data on the expected arrival volumes at the junction, it provides

an indication of the expected splits (through, left, and right movements), that can not always be obtained by detectors; secondly, in oversaturated conditions, the measured values are often more indicative of the demand rather than the capacity. Finally, in the case of long queues, whose tails reach the upstream traffic detectors, the information received may cease to be meaningful, if for instance a vehicle is stopped on a detector for a long time. Information on the existing queues is an input to the system, and it is requested from the TMC operator.

Adjust capacity on alternative paths: this is a *conditional strategy*, i.e., a strategy constructed as a consequence of the formulation of a condition, as described in the next subsection. Here the goal is twofold: to verify whether a set of non-congested paths can offer enough capacity to accommodate the diverted traffic without giving rise to additional congestion, and to determine suitable control settings along those paths. The path capacity is given by the capacity on the links that compose it, and it is a function of the link characteristics, of the control implemented at signalized junctions (intersections and metered ramps), and of the current link demand.

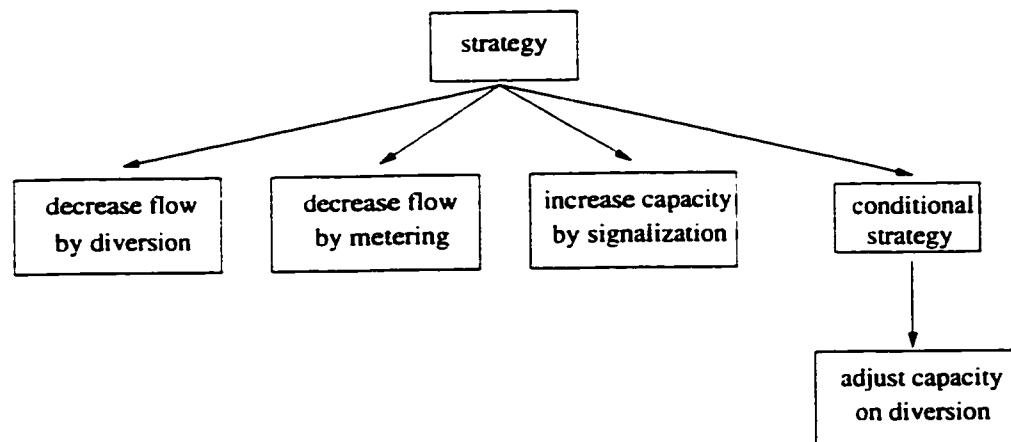


Figure 5.6: The strategy class hierarchy

Beside facilitating the problem decomposition into treatable subtasks, the hierarchical class definition of strategies and their dynamic instantiation support a flexible control structure that allows one to define additional strategies. The off-line definition of additional strategies requires the identification of the possible benefits that can be achieved, and the formulation of appropriate methodologies to implement them. Additional strategies may

include adjusting capacity downstream of an accident, by reducing the green time assigned to an intersection approach whose flow is reduced by the occurrence of an upstream incident and increasing the green time assigned to other approaches. A possibility is also to couple that strategy with one that assigns higher capacity to alternative routes that bypass the incident location. Another possible strategy is to implement *reverse progression* signal offset, for closely spaced congested intersections. It is also possible to define further conditional strategies, as a result of additional conditions. For instance, if a freeway on-ramp is downstream of a signalized intersection, and the two are closely spaced, the adequate metering rate may depend on the demand and the capacity assigned to the lane movements that feed the freeway ramp, and the choice of a certain rate is conditional on the selection of a certain signal plan.

The assignment of the *priority index* is a very important factor in the problem-solving process, and one that offers ground for further research. The priority index, by determining the selection of strategies, influences the way a problem is tackled, and characterizes the interaction among subgoals, at both intra- and inter-agent level. It is also responsible for avoiding redundant computation at the intra-agent level, and for making sure that potentially effective solutions are not ignored or lost. The priority index is an integer that is assigned to a strategy at its creation. A *maximum priority* index is also assigned to the current state of each problem, during the traversal of the search tree, in the DTPS algorithm. Priority is organized in an increasing order, according to which, the highest priority is given to the strategy with the lowest index. The maximum priority index of a problem corresponds to the highest among the priority indices of the strategies selected so far for that problem, and provides information on all the strategies that have been applied for the resolution of that problem. This mechanism thus allows to define dynamic criteria for strategy selection by an automatic dynamic assignment of priority indices.

Another advantage of this methodology is that, by keeping track of the strategies already processed (and the subgoals already achieved), it allows the algorithm to avoid redundant computation and, at the same time, to avoid overlooking possible solutions. Consider the following example, illustrated in Figure 5.7 a). Assume that a problem P_1 has to be solved, and that both strategies S_1 and S_2 can be applied, resulting, respectively in control actions CA_1 and CA_2 . The visit of node N_i results in the construction of two strategies S_1 and S_2 and of nodes N_{i+1} and N_{i+3} , corresponding respectively to the selection of CA_1 and CA_2 . Without a maximum priority index, the visit of node N_{i+1} is likely to

result in the selection of strategy S_2 while the visit of node N_{i+3} is likely to determine the selection of strategy S_1 , and the implementation of the corresponding control actions, which would result in the creation of a redundant search branch. If strategies S_1 and S_2 have different indices, with S_1 having a higher priority (lower index) than S_2 , the maximum priority index of node N_{i+3} becomes equal to the priority index of S_2 , thus preventing the selection of strategy S_1 from N_{i+3} , and avoiding the redundant search.

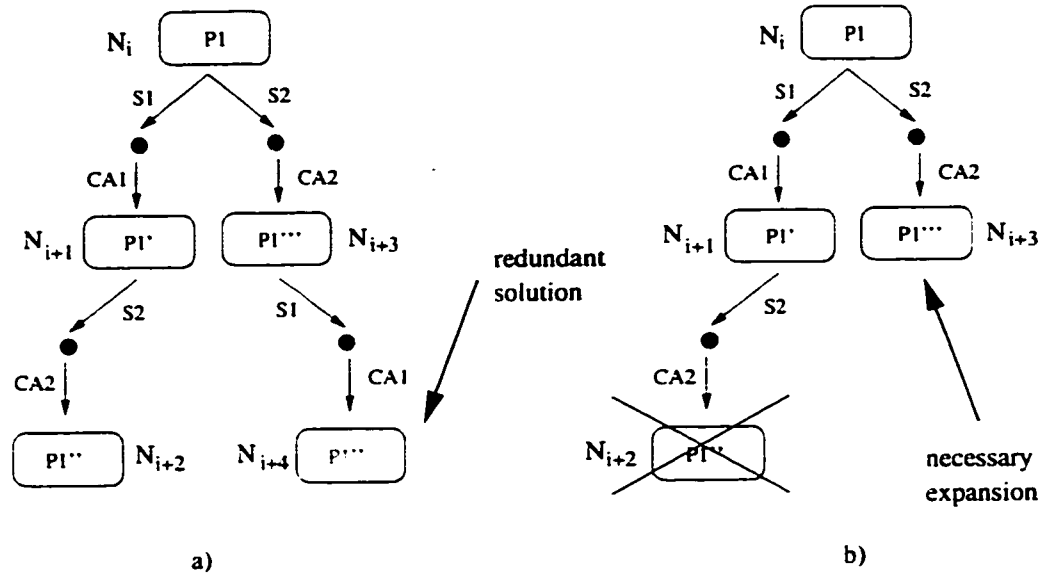


Figure 5.7: An example of redundant strategies

It may seem redundant, once node N_{i+1} has been expanded into N_{i+2} using control action CA_2 , to expand node N_i into N_{i+3} , using again CA_2 , but it is not. The need for such an expansion arises from the need of avoiding potential losses of solutions. Indeed, assuming that only the left branch of the tree was developed, using CA_1 and CA_2 , if additional information obtained in a later visit of the tree (e.g., the satisfaction of a local or global condition), made control action CA_1 incompatible and therefore inapplicable, thus removing the corresponding branch from the search tree, as shown in Figure 5.7 b), the effect of strategy S_2 and control action CA_2 would be lost. This concept is particularly important in the context of a distributed search, in which it is possible that an agent determines a set of solutions, each of which is composed of a set of control actions, and a component of one of the solutions is not compatible with any of the solutions determined by the other agent, while all the other components are. In such case, if solutions composed by other subsets of

control actions were not constructed, potential globally compatible solutions would be lost. This is a consequence of the inherent non-monotonicity of a distributed search.

Currently, a priority index is assigned to each strategy according to a very simple criterion: the priority is based on the type of the strategy, and both agents use the same numbering scheme. But the approach is very flexible, allowing the introduction of new strategies, of different numbering criteria for the two agents, and even of modifying the levels of redundancy or approximation, by dynamically assigning priorities and allowing different strategies to have the same priority, or skipping certain types of strategies.

5.3.3 Conditions

A *condition* defines a property that a solution may or may not satisfy. It specifies a constraint that propagates among the agents' search processes, and it is one of the concepts on which the compatibility between agents' solutions is based. In accordance with the object-oriented organization of the knowledge and the inference process, a condition is an instance of a class, which is dynamically constructed when the search algorithm recognizes that the applicability of a solution is contingent to the satisfaction of certain constraints. Constraint propagation is obtained by broadcasting the condition across the agents.

The satisfaction of a condition is translated into the implementation of a corresponding subtask, whose goal is indeed to satisfy the condition. A *conditional strategy* is created to process such task. Analogously to the FA/C approach, the recognition of the need for a condition represents the identification of a source of uncertainty. Indeed, the final acceptance of a solution depends on whether the corresponding condition, at local (SOU) and global (GSOU) levels can be satisfied. The definition of the conditional strategy associated with a condition represents a form of SOU resolution. By transforming a condition in a conditional strategy, a homogeneous approach is obtained for directing the agents' search solutions towards a reduction of cooperative control and global uncertainty. As seen in Chapter 4, the FA/C paradigm is based on the assumption that cooperation can be cost-effective if it is possible to exchange solution-level hypotheses rather than raw data, without losing much information, in order to take advantage of the local processing performed by each agent. In a similar way, the formulation, exchange, and satisfaction of conditions provides the cooperating agents with a method for exploiting inter-agent constraints, without the need of exchanging low-level data and without stringent synchronization requirements.

Additionally, the representation of a condition as a composite object allows the agent that generates the condition to associate with it specific types of abstract information. This will help the other agent translate the condition in a strategy, thus allowing the exchange of limited, well-timed, and relevant information, and increasing the overall problem-solving coherence.

The reason behind the use of conditions lies in the necessity to represent and propagate semantic constraints among solutions, without having to explicitly perform pairwise compatibility checks among control actions. In general, within the resolution process, there are several types of constraints that solutions must satisfy, both within and among agents. Some of these are specifically related to control actions, like the requirement that no two compatible solutions recommend different settings for the same control device (e.g., two different messages for the same CMS). Some other are related to subgoal-level decisions, and, since in general a subgoal can be achieved in several alternative ways using different combinations of control actions, it becomes unreasonable and impractical to express these types of compatibility criteria among control actions, especially in a distributed environment, where conspicuous interaction and synchronization would be required. It is more efficient, instead, to simply have an agent specify a condition and broadcast it to the other agent whenever a particular type of GSOU is encountered, and let the other agent define subgoals for the fulfillment of that condition and look for ways to achieve those subgoals. In the same way, a condition can be propagated within the search of the agent that requires it. Analogously to what happens in the remote agent, the generation of a condition will affect the construction of further local solutions.

Conditions are thus heavily responsible for both the amount and the type of interaction between agents. The lower is the number of required conditions, the more loosely coupled (and thus simpler) is the distributed search. At the opposite end, though, failing to recognize constraints among solutions may lead to the development of inefficient, or even contradictory solutions. We face once again a compromise between approximation and computability, since, as observed in Chapter 4, requiring an exact solution is impractical in many real-life domains. It is therefore important to balance the real need for cooperative conditions with the increase in the complexity of the solution algorithm.

Currently, only one type of cooperative condition, called *diversion cooperative condition* has been implemented. The condition is created whenever a control action is selected that recommends the rerouting of a certain volume of traffic, and is translated into

the conditional strategy described in section 5.3.2. Information describing an approximation of the expected effect of the control action is associated with the condition and used for the construction of the corresponding strategy. The rationale behind this protocol is twofold: first, the decision of partially diverting traffic has to take into consideration the overall network conditions, or at least the conditions of the subnetwork directly affected by the diversion; secondly, the control implemented on the subnetwork affected by the diversion should take into account the resulting variation of path flows. If the diversion affects both agents (as often happens in corridor applications, when freeway traffic is partially rerouted through adjacent arterial segments) the transmission of the condition, along with the information about the expected diversion effect, is potentially beneficial to both agents. Indeed they can try to adjust their control using relevant information, in the attempt to maximize the benefit expected by the diversion policy, both for the diverted traffic and the traffic that is indirectly affected by the diversion.

A simplifying assumption, that may introduce a certain degree of approximation is made with regard to the effect of diversion advice on traffic: the effect of CMS messages is known (at least in approximation), or can be inferred without significant interaction with the other agent. In other words, that effect is largely independent of non-local information that is not readily available to the agent selecting the message. From a global (system-optimal) point of view, the best control solution is one that results in an overall travel time decrease, both on the congested paths, on the alternative paths, and on those paths that are indirectly affected by the diversion (in the corridor example, the travel time of vehicles on streets crossing the arterial used as the alternative route). On the other hand, a purely system-optimal solution is not feasible, because travelers will follow rerouting indications only if they perceive that it is to their advantage to do so (according to a user-equilibrium criterion). In reality, given the approximate nature of the input data, the inherent inaccuracy of existing algorithms for travel time computation and estimation of response to traffic information, and the computational cost of information exchange, neither of the two solutions can be practically obtained, and heuristic reasoning is used instead.

The following issues drive the formulation and satisfaction of conditions: first, diversion is implemented only for moderate to major disruptions, that result in significant delays; secondly, alternative paths are considered based not on finely balanced equilibrium criteria (such as the comparison of travel times on the original and alternative paths), but rather on coarser guidelines, such as the avoidance of additional oversaturation on the alter-

native paths, decided and regulated by the agency or agencies responsible for the subnetwork to which the alternative paths belong. Still referring to the freeway/arterial corridor example, in case of heavy delay on a freeway section, if the freeway agent selects CMS messages to divert a portion of traffic towards adjacent arterials, and notifies the arterial agent of the possible additional demand on those arterials, the arterial agent will check whether the resulting demand can be served without reaching oversaturated condition, and based on this criteria will decide whether a compatible control (i.e., a solution satisfying the requested condition) can be determined. The currently used criteria for avoiding oversaturation uses acceptance thresholds on the degree of saturation and maximum expected queue lengths at all the signalized intersections and metered ramps along the alternative paths.

The way the algorithm handles conditions is a very important factor in the cooperative process, responsible for the degree of interaction between the agents. It defines the balance between local and remote processing and the degree of solution and cooperative control uncertainty of the whole process. These issues assume particular importance especially if different (approximate) termination criteria are adopted.

The first and simplest degree of freedom concerns the priority assigned to conditional strategies. A conditional strategy is derived from either a local or a remote condition. A distinction between local and remote search can be obtained by assigning different priorities, depending on whether a condition is local or remote: a high priority to a local conditional strategy determines a more self-directed behavior, according to which agents first look for the satisfaction of local conditions and subgoals. Vice versa, a high priority to remote conditions determines a more externally directed behavior. In general, as observed by FA/C researchers, the more externally directed the behavior of an agent is, the higher is the potential for distraction. On the other hand, in a highly self-directed environment, cooperative control and solution uncertainty tend to increase, because the information received from other agents will be only partially exploited. It is therefore important to choose the right balance, in order to reduce the agents' potential for distraction while still guaranteeing the right exploitation of externally received information.

Another possible variation is related to *when* conditions should be exchanged among agents. The simplest approach is to transmit a condition as soon as it is detected. In this way, the receiving agent is notified as early as possible of the need to satisfy the condition. On the other hand, an agent might want to wait to transmit a condition until it finds at least a local solution that satisfies it, thus reducing its uncertainty, since this

avoids to request the satisfaction of a condition at a remote level, when in reality, no local solution satisfies it. Researchers in the FA/C protocol have found that an agent should first process goals with low uncertainty that can reduce the control and solution uncertainty of the other agent, and also that an early transmission of highly uncertain solutions does not allow to take full advantage of the nearly-monotonic characteristics of certain domains.

Several different levels of accuracy in the resolution process can be obtained by allowing the agent that is requested to satisfy a remote condition, to return information about the satisfaction of such condition. Such a model of interaction allows the agent that has requested a condition to alter its search for local solutions, depending on whether and by what types of remote solutions, that condition has been satisfied. Since the resolution process is incremental, this type of interaction requires a continuous, or periodic update of the returned information, because in general, the synthesis of new data may affect the satisfaction of certain conditions. The simplest form of interaction involves returning a *boolean* answer, that basically states whether there exists at least one remote solution that satisfies a given condition. On the other end of the complexity spectrum, the most accurate solution (*completely accurate*, according to the definition given in Section 4.1) would require to return detailed and updated information about all the remote solutions that satisfy the given conditions. As previously observed, such an approach may become very computationally expensive, since every time a new solution is created or a new condition is received, extensive communication and synchronization are required. An approach in between the two involves returning a measure of how likely the condition is to be satisfied by the set of final solutions, or how beneficial it is expected to result. A measure of the condition adequacy can be obtained using a weighted sum of the number and type of the solutions that satisfy it and of the degree of benefit expected by those solutions.

Further research and empirical analysis of the resolution process may help determine the right balance among approximation and computability, and determine the interaction protocol that is most suitable for the problem domain.

5.3.4 Termination Criteria

Related to the issue of the complexity and efficiency of the relationship among agents' goal structures is the issue of terminating the cooperative problem solving process. Termination of the global problem solving process requires similar reasoning capabilities to

those needed for basic coordination strategies, including identifying what part of the goal structure have been analyzed, and the uncertainty associated with current solutions, and understanding whether all or enough system-wide constraints have been developed.

The current algorithm, as explained in section 5.3.1, is a two-phase process, organized as a local solution search and a global solution composition. The first stage defines all possible combinations of subgoals that can be achieved with locally compatible and consistent control actions. The second is a multi-stage process in which compatibility and effectiveness criteria are progressively refined to select a limited, ordered set of alternative solutions.

While for the network under study the current termination criteria seem to be adequate, requiring an acceptable computation time, for larger networks or larger systems it may be necessary or convenient to devise approximate methodologies, able to limit the computation time of either of the two phases of the problem solving process. Alternative methodologies can result from further exploitation of properties of the problem domain and empirical observations of the solution process. Possible termination criteria may be as simple as reaching a threshold, for instance a maximum number of iterations, or a maximum size or depth of the search tree, or even a maximum execution time. More complex criteria may include the analysis of which subgoals have been completely achieved, to what extent other goals have been explored and so on. At this moment, these are all open questions.

5.3.5 Near-Monotonicity

Most reasoning processes that involve uncertain or incomplete information are non-monotonic. As reported in Chapter 4, Carver et al. (1996) introduced the concept of *near monotonicity*, to distinguish those problem domains for which approximate global solutions can be produced from local components computed using only a portion of the data and exploiting opportune inter-agent constraints. They suggest that in nearly-monotonic domains a FA/C-based interaction protocol has the potential to produce approximate but efficient global solutions from locally computed components. In other words it is possible to develop an approximate problem-solving process in which local solutions can be used as a substitute for the raw data, without requiring complete re-processing of the initial data when new information becomes available.

The potential for a FA/C-based approach to exploit this characteristic, in the con-

text of distributed traffic control, can be shown considering the way solutions are constructed by the DTSPS algorithm. Each agent, using local data and global constraints resulting from the exchange of conditions, develops a search tree composed of alternative combinations of control actions. Each local solution is given by a path in the tree from the root node N_0 to a node N_j . In general, if there exists a local solution S_j that is incompatible with all or a large number of the solutions developed by the other agent, this is because there are one or more control actions, among those that compose S_j , that are incompatible with a large group of remote solutions (for instance, they require the satisfaction of a condition that can not be easily satisfied). Let CA_j be the control action among the ones responsible for the incompatibility of S_j that is closer to the root node. Assume N_k is the node that is obtained, within the path from the root to S_j , by applying CA_j , as shown in Figure 5.8. The subtree of N_k is then composed of solutions that are all globally incompatible, and therefore it coincides with an “unused” search process. But the above part of the path, from N_0 to N_k , may be part of compatible solutions, like the ones whose corresponding paths go through N_{j+1} . In this case the local processing performed to build the corresponding control actions is not useless.

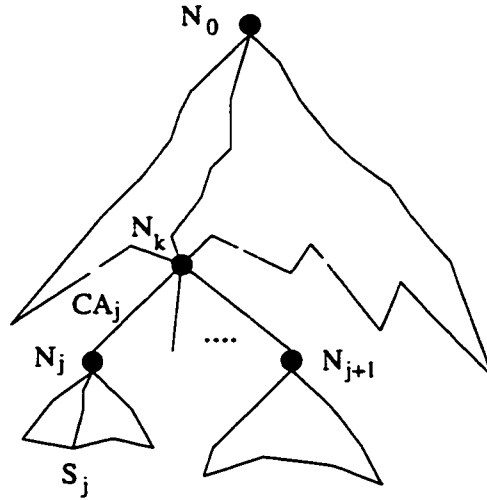


Figure 5.8: The local search tree

An efficient approach is thus one that exploits local processing in an effective manner, i.e., one that reduces the portion of unused searches to the minimum. This basically corresponds to reducing the size and the number of “unusable” subtrees, like the one of N_k in Figure 5.8. Nodes like N_k , before information about the compatibility of the condition

required by CA_j is received, correspond to highly uncertain hypotheses in the field of distributed sensor interpretation. If information able to reduce the node uncertainty can be gathered during the local search in the first phase of the DTPS algorithm, without the need for extensive synchronization and delay, considerable efficiency may be gained. A problem-solving process that postpones the development of nodes like N_k until information about the likelihood for its global compatibility is received, has the potential to achieve this efficiency. Possible solutions, with different degrees of effectiveness and complexity, are the ones highlighted in section 5.3.3.

5.3.6 Computational Complexity

The analysis of the complexity of the DTPS algorithm is similar to the one performed for the centralized algorithm (TPS), described in Chapter 3.

DTPS is computationally more complex than its centralized version. The added complexity is given by the creation and propagation of intra- and inter-agent conditions, during the first phase, and by the construction and ranking of global solutions, during the second phase. The complexity of the first phase of DTPS depends largely on the number of control actions that can be selected, since for every applicable control action a new node, in the developed search tree is created. Highly exponential upper bounds in the algorithm complexity are typical of search algorithms like the one used in this approach, which is the main reason why heuristic techniques are used to reduce the dimension of the solution space.

In the first phase of DTPS, as described in section 5.3.1, each agent performs a recursive procedure that develops and visits a search tree, iterating this process until no new external condition is received from the remote agent. The number of iterations thus depends on the number of conditions that can be transmitted among the agents. In its current formulation, DTPS associates conditions to diversion control actions. A diversion cooperative condition depends on the diversion frame that generates it (the group of CMS messages responsible for traffic diversion) and on the characteristics of the problem that is being analyzed (the type of problem and the expected delay determine the expected CMS compliance rate). Therefore, if p is the number of problems that are concurrently being analyzed by the algorithm and D is the size of the database of diversion frames, a maximum of $p * D$ conditions can be exchanged among the agents.

If the problem description of each tree node has dimension p (the number of problems currently analyzed), for each problem, a maximum of $3 + c$ strategies (those described in section 5.3.2 and the conditional ones, assuming $c < D * p$) can be selected. A diversion strategy can be translated into a maximum of D control actions, while any other strategy is translated into a single control action. Thus a node can generate a maximum of $(D + 2 + c) * p$ children nodes. The maximum depth of the search tree l is given by the maximum number of strategies that can be applied to a node ($p * (3 + c)$). The maximum number of nodes is thus:

$$n = \frac{[(D + 2 + c)p]^{l+1} - 1}{(D + 2 + c)p - 1} \in O[(Dp)^{2p^2 D}]$$

In the most complex scenario, a new condition is generated at each iterative step, thus the maximum number of iterative steps is $p * D$. An upper bound for the complexity of the first stage of the algorithm (the number of visits to the tree times the size of the tree) is thus

$$O[(Dp)^{2p^2 D} Dp] = O[(Dp)^{2p^2 D}]$$

which is highly exponential and virtually unbounded, since in theory the number of concurrent problems p is unbounded.

In practice, however, the adoption of heuristic techniques significantly reduces the complexity of the algorithm. The use of a priority index reduces the number of possible combinations of strategies, as explained in section 5.3.2. Also, the assessment of the expected effect of diversion frames reduces the number of applicable diversion control actions, reducing both the total number of nodes in the tree and the total number of conditions. The verification of compatibility among control actions further reduces the number of applicable control actions and therefore of the tree nodes. Finally, heuristic techniques such as the verification of the satisfaction of intra-agent conditions prior to their broadcasting to the remote agent reduces the number of conditional strategies, as described in section 5.3.2.

In all simulated scenarios, a maximum of five concurrent problems were simulated for testing purpose. In general, it was observed that, while the complexity of the algorithm grows with the number of problems being addressed, a higher number of problems reduces the number of possible diversion solutions, thus reducing the size of the search tree.

Given the real-time nature of the problem and the need for providing decision support in a *reasonable* time, termination criteria are used, as described in section 5.3.4,

to ensure a computation time that falls below acceptable thresholds. In general, the higher is the number of analyzed solutions (i.e., the higher are the tree size and depth), the more likely it is to find a better solution, since the portion of the analyzed solution space is larger. However the ranking of strategies and the possible ranking of solutions according to the value of a heuristic function (such as the one based on the capacity/demand imbalance at critical sections) drives the solution process towards selecting the most promising solutions first, in accordance with the general characteristics of the BFS technique described in Chapter 3.

The computational complexity of the second stage of the DTPS algorithm is also, indirectly, dependent on the number of local solutions determined by each agent. The most computationally intense operations are the pairwise matching of local solution for the construction of mutually compatible global solutions ($n * m$ operations, if the number of local solutions found by the agents are respectively n and m) and the assessment of the expected delay reduction associated with each non-redundant global solution.

The first operation involves matching the set of local and remote satisfied conditions. The complexity of this task is polynomial ($O(n * m)$). For what concerns the second operation, it must be observed that, in general, step (c) in the second phase of DTPS allows to significantly reduce the number of solutions for which a macroscopic delay reduction must be assessed. Nonetheless, the estimation of the delay reduction is a complex task, since it requires each agent to compute the delay associated with each control action (the delay expected for each suggested signal plan and metering rate) and at each critical section (details on these operations are provided in Chapter 6). In its current formulation, the algorithm estimates the expected delay reduction for each non-redundant solution. As reported in Chapter 7, the current dimension of the problem (the network size) is small enough that a final set of solutions is obtained in what can be considered a reasonable time (less than 30 seconds). More sophisticated formulations can be envisioned, where only a limited number of the most highly ranked solutions (based on the value of the heuristic function) are considered for the macroscopic assessment of the delay savings.

As the size of the network controlled by CARTESIUS increases, the number of concurrent incidents that the system has to manage is likely to increase. Furthermore, the number of control devices under the jurisdiction of each agent is expected to augment. Finally, the total number of possible diversion schemes may increase due to the availability of both a higher number of alternative routes and a higher number of available CMS.

To analyze the algorithm's scalability, the following considerations should be taken

into account: for what concerns the availability of alternative routes, if the network *density* (i.e. the total length of links per square unit area) and the CMS density (the number of CMS per square unit area) remain roughly constant, a similar number of applicable diversion schemes is to be expected for each problem. The estimation of the effect of a CMS frame (a diversion scheme) with respect to a problem allows to disregard all those groups of settings that are not relevant or applicable to a given problem. The limited number of applicable diversion strategies is also contained by considering that effective diversion strategies do not normally involve long and complex detours.

For the same reason, the number of control devices affected by diversion (e.g. signalized intersections or metered ramps for which a new plan must be computed) is expected to remain roughly constant. Furthermore the number of control devices affected by metering strategies is not expected to grow significantly, since the further away a controlled junction (metered ramp or signalized intersection) is from a congested location, the smaller is its effect on congestion.

5.4 Current Problem Monitor

The Current Problem Monitor, as described for the centralized module in Chapter 3, has the task of monitoring the evolution of those congestion phenomena that have already been addressed. These include phenomena for which a control solution has been selected, but which, due to changes in traffic patterns or the possible inadequacy of the solution, have not yet dissipated. The existence of such a function is driven by the need of some sort of feedback on the implemented control, that provides an indication of how well the current solution is addressing the current problems. Also, it renders the system able to analyze causal relationships among congestion events, and devise control solutions that take those relationships into account. Furthermore, by helping the system recognize input data describing problems that have already been analyzed, it avoids continuous repetitions of the same analysis, alleviating the system's burden and relieving the operator from repetitive interaction with the machine.

The complexity of this module may vary according to the depth of the analysis performed. In general, whatever measure of effectiveness is used, automatically recognizing how well a certain control is performing is a complex task. This is because differences between the expected and the actual performance may be due not to the inappropriateness

or inefficiency of the control, but rather to fluctuations of the demand, or to variations in travelers' response to control. This module currently performs only a rather coarse analysis, mostly based on recognizing the onset of additional congestion and the dissipation of previous problems.

Several functions and complex objects for the representation of knowledge and of the inference process are used, but the main, high-level reasoning protocol for dealing with current problems can be summarized in three procedures and in one complex data structure, (one per agent). This structure stores an abstract representation of current problems, of the corresponding currently active control solutions, and on conditions to determine when congestion dissipates and whether the occurrence of other congestion events is related to it. Analogously to the structure described in Chapter 3, the following terms describe the information stored in the data structure, that the procedures deal with:

Previous problems: an abstract representations of the (list of) problem(s) for which a global control solution has been found and is currently being implemented.

Expiration time: the time until which the current demand pattern is expected to be valid. The time-varying network demand is partitioned in a list of time intervals, of 15 minutes each, resulting from a time-varying traffic assignment algorithm. When the validity of the current flow assignment expires, it may be necessary to reassess the control plans.

Presence conditions: a logical expression that indicates the conditions that mark the presence of previous problems, such as the congested status of links or paths, and whether the expected interval for the dissipation of congestion (the *minimum recovery time*) has been completed.

Context conditions: a logical expression that describes conditions that have to be satisfied for the currently implemented control to be still valid. The verification of the context conditions helps determine whether it is necessary to reconsider the problem and its control. Context conditions describe the status of alternative paths, if the current control recommends diversion, of intersection approaches that were not congested at the time a certain signal plan was implemented, and of freeway sections downstream of metered ramps, that were not congested at the time the current metering rate was chosen.

The information related to the previous problems is stored in a distributed fashion: an agent maintains only the abstract representation of those problems that are local to the network under its jurisdiction. This choice is dictated on one hand by the desire to preserve the agent's privileged access to local information, on the other by the need to reduce, whenever possible, the system redundancy. Information about context conditions is distributed according to the locality of the resources it refers to. If it refers to the status of paths, since paths may span across the whole network, it is stored by both agents. If it describes the status of intersection approaches, it is stored by the arterial agent, while if it refers to the status of freeway sections, it is part of the knowledge of the freeway agent.

The next three procedures are responsible for the inference performed by the Current Problem Monitor.

Update Previous Problems: this procedure is executed at the end of the reasoning cycle, after both operators have agreed on a global control solution. The tasks of this procedure involve defining the context conditions using information about the implemented control, and translating the "current" problems into "previous" ones, by specifying their presence conditions and their minimum recovery time.

First Filter: this procedure is executed at the beginning of a new reasoning cycle, right after input data describing traffic conditions have been updated. The procedure is performed before the interaction with the operator, and it is used to check whether the recovery time has expired, or the presence conditions have been invalidated from the newly updated data. If either of the above situations has occurred, then the corresponding problem is flagged for removal, but it is not deleted until confirmation from the operator is received.

Second Filter: this procedure is the most complex, and the one that requires most interaction among the agents. It has the task to update some of the information about previous problems and determine which control actions have to be preserved until the termination of their recovery process, and which actions have to be reevaluated. It is executed after the operators have confirmed the occurrence of new problems and the clearance of previous ones, information has been gathered about them, and functions within the *Problem Analysis* knowledge units have determined relationships (derivation) among existing congestion phenomena. In light of the new data or the

new interpretation of old data, this procedure checks once again the presence conditions and expiration time of previous problems, and verifies whether the context conditions stored in the knowledge bases of the two agents are verified. If previous problems have disappeared, they are deleted. If they are recognized to have evolved in an unexpected manner, or additional problems related to them have occurred, they are inserted once again among the list of problems to be solved. In both cases the control that had been previously selected is reset. If a problem has not yet dissipated but no additional related congestion has occurred, the corresponding control must be preserved.

Chapter 6

Control Algorithms: Software Design and Development

This chapter presents a description of the methodologies used by the DTSP algorithm, for the selection of appropriate control settings for signals, ramp meters and CMS, the estimation of the network demand and the assessment of the corresponding expected delay. The final section describes a possible methodology to partially represent the uncertainty about driver's compliance to CMS messages.

Most of the algorithms presented in this chapter use methodologies defined in the literature, that, under most circumstances have demonstrated their validity. Indeed, as mentioned earlier, the aim of this research is not in developing new signal and ramp control algorithms, but rather to construct a tool that efficiently embodies existing methodologies into a well-structured, distributed architecture. The modularity of the system architecture allows to substitute with minor adaptations, these methodologies with more sophisticated algorithms, provided that their interface with the system is respected.

6.1 Signal Control

The formulation of control plans for actuated signals utilizes algorithms developed by Akcelik (1981) and Gazis (1974). A distinction is made between the cases of oversaturated intersections and undersaturated ones, whose control plans need to be adjusted as a consequence of planned diversion control. For a signalized intersection along a path that is used as an alternative route, it is necessary to a) determine whether enough capacity is available to support the additional flow that is expected to be diverted through it; and b)

define a signal plan that maximizes the capacity and reduces the overall delay, both for the diverted traffic and for the portion of traffic indirectly affected by the diversion. On the other hand, when an intersection becomes oversaturated, vehicles at one or more approaches may form a queue. In this case, especially if the approach is closely spaced downstream of another intersection or of a freeway exit, it becomes important to avoid the spreading of congestion. An appropriate management of arrival volumes and existing queues becomes therefore necessary, in order to obtain a reduction in the network delay. The algorithms used are described in the next two subsections.

6.1.1 Signal Control on Diversion Routes

The travel time on a path is determined by the travel time on the links that compose it and by the capacity at junctions (ramps and signalized or unsignalized intersections). Signalized intersections use traffic-actuated signals. Varying levels of capacity may be provided by properly setting minimum and maximum green and unit extension times for the various approaches, and determining an appropriate phase sequence. Many of the demand-responsive traffic control schemes proposed in the literature have not been thoroughly tested for oversaturated conditions, neither has their capability to operate in conjunction with traffic diversion been yet verified. The hypothesis underlying the methodology used in this research is that traffic diversion, in response to severe congestion, may determine a sudden and substantial shift in the network demand that is not captured promptly enough by demand-responsive control.

There are two issues that, in this respect, are considered fundamental:

- The agent in charge of allocating the adequate capacity to the network, through signal or ramp meter control, should be able to use and take advantage of the information about the predicted shift in network demand caused by traffic re-routing.
- The agent in charge of determining alternative paths that bypass the congested location should be able to judge, either through its own knowledge, or using knowledge coming from another agent, whether these alternative paths are feasible and can provide total delay and travel time reduction.

As noted in Chapter 5, these two issues determine the following (interdependent) requirements:

1. When an agent plans to recommend a set of CMS messages (whose effect is to divert a portion of traffic from a set of paths to another set of paths), before the control directive can be safely implemented, it is necessary to make sure that the alternative paths have enough capacity to accommodate the extra demand.
2. It is necessary not only to accommodate the extra demand (plus the original one), but also to find the "best" control setting, or a satisficing one, under certain constraints and according to certain criteria, i.e., the capacity allocation that minimizes total delay, or that provides volume/capacity ratios below a given threshold, or one that guarantees equitable queue lengths.

The analysis of the cost/benefits associated with a control setting is based on the computation of the lane movement delay for each intersection, given the demand distribution at each approach, for the time T during which the control is active. The analysis of the delay uses a generalized formula proposed by Fambro & Rouphail (1997), that takes into account actuated signalization and oversaturated conditions. This formula, that requires the computation of the average cycle length and green splits, as described by Akcelik (1994), has been proposed as a modification of the latest version of the Highway Capacity Manual (HCM 1994)).

After the occurrence of an incident and the consequent selection of a set of CMS messages, the near-future expected volume at intersections along alternative paths changes. In order to determine the feasibility of the diversion control strategy, it is thus necessary to look for plans that satisfy specific constraints on volume to capacity (v/c) ratios, average delay, maximum cycle length, and that minimize the total delay. In general, a diversion action will require to accommodate extra flow, therefore increasing the total delay compared to the original situation, but if the "costs" incurred are smaller than the global expected benefits, then the network-wide effect is to ameliorate the disruption caused by the incident.

The algorithm proposed by Akcelik (1981) is an iterative process, used to determine the required green times for lane movements. Once the maximum required green times have been determined, given an expected demand distribution, a function defined by Akcelik (1994) allows to estimate the corresponding average green time associated with each movement. The computation of the average expected delay is then used to establish the level of service (LOS) of the intersection, and thus the feasibility of the control plan.

In the work presented herein, no distinction is made between displayed and effective

green time, i.e., it is assumed that for each movement the initial lost time and the final intergreen intervals are the same (their difference is variable and is in general estimated between 0.5 and 1 seconds (Webster & Cobbe 1966)).

The algorithm for the selection of maximum green times is based on the search for the *critical movements*, i.e., those movements that determine the capacity and timing requirements of the intersection, and on the selection of appropriate parameters for them. If sufficient green time is allocated to each critical movement so that its capacity requirement are satisfied, then all movements will have sufficient capacity. Actuated control allows for the existence of overlap movements, i.e., movements that receive right of way during more than one phase. If all movements were non-overlap movements, there would be one critical movement per phase, the one requiring the longest green time. Once the required time for the critical movements has been determined, it is possible to compute the required cycle time. For simplicity an eight-phase plan based on the dual ring shown in Figure 6.1 is used. The iterative process for the computation of the required movement times starts assuming an initial value for the cycle length. It then computes green times based on the assumed cycle length, and then readjusts the cycle length and if necessary the green times iteratively, until an equilibrium is reached. The required movement time is calculated by

$$t = \frac{c}{u} + l \quad (6.1)$$

where l is the movement lost time, u is its required green time ratio, and c is the initial value of the possible cycle length. The required green time ratio that allows to achieve maximum acceptable (practical) degree of saturation x_p , is given by

$$u = \frac{y}{x_p}$$

where y is the movement flow ratio, i.e., the ratio of arrival flow q to saturation flow s . The practical degree of saturation is the desired ratio of arrival flow to capacity. Typical values for the practical degree of saturation range between 0.85 and 0.95. The movement time computed from equation 6.1 must satisfy the constraint

$$t \geq g_m + l \quad (6.2)$$

where g_m is the fixed minimum green time. If equation 6.2 is not satisfied, then the required green time is set to

$$t = g_m + l$$

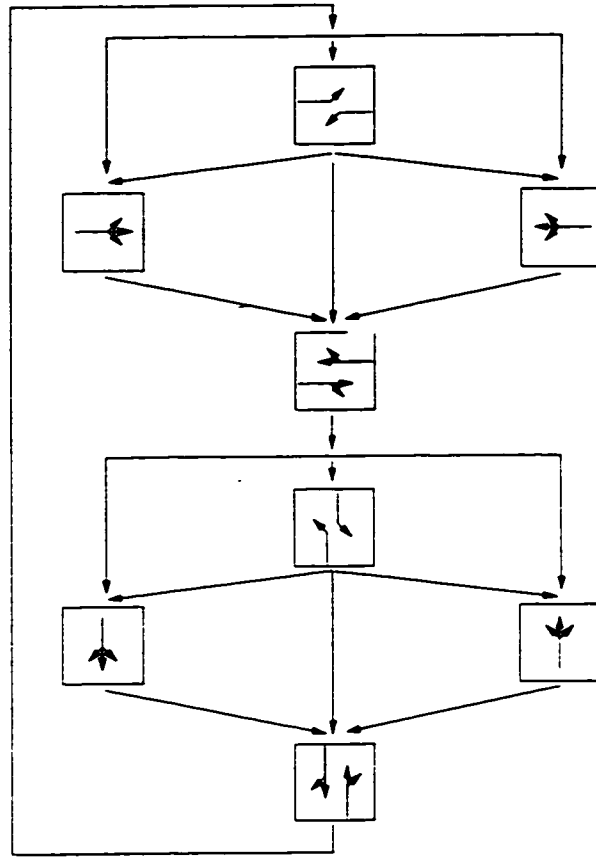


Figure 6.1: An eight-phase dual ring

The identification of the critical movements is based on the comparison of the required movement times, for all possible combinations of phase sequences in a cycle. The sequence that has the highest total required movement time is the one that determines which movements are critical. The possible sequences, or *critical paths* can be represented through a graph, as shown in Figure 6.2 b). The four sequences that may result are:

- 1 - 2 - 4 - 3
- 1 - 2 - 7 - 8
- 6 - 5 - 4 - 3
- 6 - 5 - 7 - 8

Once the critical movements have been identified, the practical and approximate optimal cycle times are computed using the intersection values of lost time L , flow ratio Y ,

and green time ratio U . These parameters are computed as the sum of the corresponding parameters for the critical movements. If the green time of a critical movement is specified by its minimum time (equation 6.2), the movement's green time is added to the intersection lost time (i.e., g_m is treated as lost time for other critical movements), and its flow and green time ratios are not included in the corresponding intersection parameters.

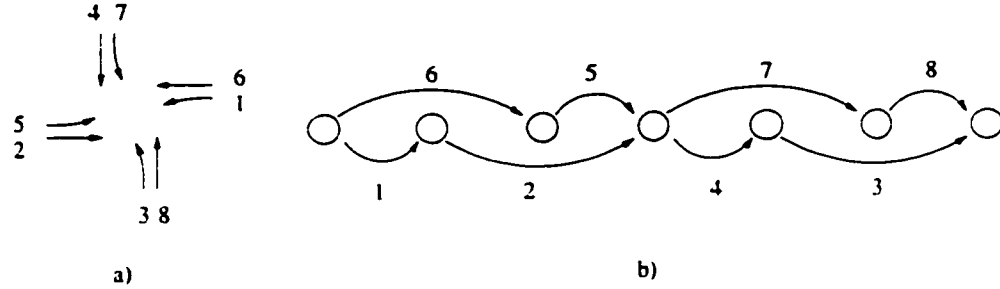


Figure 6.2: The movement numbers and the critical movement search diagram

An approximate value for the optimal cycle time is given by Akcelik's (1981) parametric formula:

$$c_0 = \frac{(1.4 + k)L + 6}{1 - Y}$$

where k is such that $k/100$ is a stop penalty parameter that varies according to the performance measure the equation is trying to optimize. Akcelik suggests a value of $k = 0.2$ for the optimization of total travel cost.

The practical cycle time is the minimum cycle time that ensures that the degree of saturation for all movements falls below a specified threshold ($x < x_p$), and is computed as

$$c_p = \frac{L}{1 - U}$$

If c_0 and c_p fall below a maximum acceptable cycle time (normally between 150 and 200 seconds) and $U < 1$, a cycle time is chosen between the two. If $U \geq 1$, or if the practical cycle is too high, then the intersection demand is too high for the available capacity, i.e., there exists no signal plan that can satisfy the demand without giving rise to the formation of queues, and the routes that pass through the intersection are deemed inappropriate as alternative routes for diversion.

However, if the chosen cycle length falls below acceptable thresholds, it is possible to determine a plan that is able to conveniently serve all demand. The required green times

are computed using the new value of the cycle time (that, in general, will be different from the first estimation of 100 seconds). Because of the new cycle time, the critical movements in general may change, so it is first necessary to check whether they are the same as the ones previously identified. If they have changed the procedure must be repeated, considering the selected cycle length as the starting cycle length. Normally, however, critical movements are not significantly affected by the length of the cycle (unless the cycle is so small that the minimum green time makes a difference), so only few, if any, repetitions of the procedure are necessary. Once a cycle length and a corresponding sequence of critical movements have been determined, the actual green time computation follows, analogously to the traditional methodology, by first partitioning the total available green time ($C - L$) among the critical movements, proportionally to their required green time ratios

$$g_i = \frac{c - L}{U} u_i$$

and then assigning the green times to the corresponding non-critical movements.

The last step involves verifying whether the degrees of saturation corresponding to the selected green times, fall below the desired thresholds for all movements. This allows to determine whether the levels of capacity associated with the computed green times are sufficient to serve the expected demand, and thus whether the desired flow can be diverted through the intersection without generating additional congestion. The threshold degree of saturation may vary across the intersection and from intersection to intersection. It is specified, according to local or global criteria, by the arterial agency, that thus holds control of the acceptable saturation levels.

6.1.2 Signal Control at Oversaturated Intersections

In undersaturated conditions, most of the literature is in agreement on the need of minimizing total delay per cycle, by partitioning the green time among conflicting approaches in such a way to provide them with the same degree of saturation (Webster 1961). If the demand on some of the approaches is such that not all vehicles can be served within one cycle, and the total demand becomes greater than the total capacity, different strategies are necessary. In other words, if M is the set of movements, when for a prolonged time interval

$$\sum_{j \in M} \left(\frac{q_j}{s_j} \right) > 1 - \frac{L}{c}$$

not all vehicles will be served within a cycle as they approach the intersection, and queues will form at some of the approaches. A strategy is thus required that takes into account not only the arrival rate per cycle, but also the existence of queues of waiting vehicles.

Gazis (1974) developed a simple, macroscopic method for determining the right balance of green splits at congested intersections. This is not truly an on-line highly responsive policy, but it has the advantage of simplicity and of not requiring continuous updates of queue length measurements, when compared to other methods (Longley 1968, Pignataro, McShane, Crowley, Lee & Casey 1978).

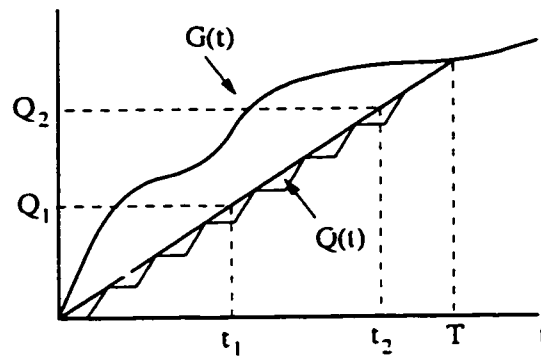


Figure 6.3: Demand and service rates

Gazis's (1974) store-and-forward approach is based on the Automatic Control Theory and assumes the knowledge of the demand function which, in DTSP is estimated from a previous traffic assignment. The cumulative arrival curve Q and cumulative service curve G , shown in Figure 6.3, are defined by

$$Q(t) = \int_0^t q(\tau) d\tau \quad G(t) = \int_0^t \gamma(\tau) d\tau$$

where $\gamma(\tau)$ is the average service rate. The total intersection delay is measured by the difference between the two curves. Thresholds for the minimum and maximum green time at each movement j provide constraints for the region of admissible solutions, as follows

$$g_{min} \leq g_j = \frac{Q_j}{s_j} \leq g_{max} \quad (6.3)$$

Assuming the simplified case of only two competing traffic streams at the intersection, any feasible strategy involves time-dependent $\gamma_j(t)$ that satisfy equation 6.3 and the equation

$$\frac{\gamma_1(t)}{s_1} + \frac{\gamma_2(t)}{s_2} \leq 1 - \frac{L}{c} \quad (6.4)$$

where s_j is the saturation flow of stream j , for $j = 1, 2$.

The problem of optimizing the performance of the intersection can then be formulated as follows:

Problem: Minimize the delay function

$$D = \sum_{j=1}^2 \int_0^T (Q_j(t) - G_j(t)) dt$$

where $\gamma_j(t)$ are subject to equations 6.3 and 6.4 and the termination time limit T , assuming that after T the demand will fall again below the capacity, is defined by the equation:

$$G_j(T) = Q_j(T) \quad (6.5)$$

Feasible green splits and the earliest possible time T at which oversaturation terminates, are determined by assuming a single setting for the traffic light (i.e., constant γ_j) for $0 < t < T$, in such a way that both queues dissolve simultaneously. The service rates and the termination time are determined by the following system of equation

$$\begin{cases} \frac{\gamma_1(t)}{s_1} + \frac{\gamma_2(t)}{s_2} \leq 1 - \frac{L}{c} \\ \gamma_j T = Q_j(T) \end{cases} \quad j = 1, 2 \quad (6.6)$$

If a constant arrival rate B_j is assumed for the interval under analysis, and the initial queues A_j are known or estimated, so that the cumulative approach demand is approximated by

$$Q_j(T) = A_j + B_j t,$$

then

$$T = \frac{A_1 s_2 + A_2 s_1}{-(B_1 s_2 + B_2 s_1) + s_1 s_2 (1 - L/c)}$$

In order to extend Gazis's (1974) algorithm for fully actuated control of an intersection with eight movements, it is necessary to utilize once again the concept of sequences of pairs of critical movements. Given the dual ring shown in Figure 6.1, for the computation of the termination time and appropriate splits, it is necessary to consider $2^4 = 16$ possible combinations of pairs of related movements. In this case, a critical movement is determined not only by its arrival rate but also by the length of the queue on the corresponding approach. The 16 combinations of movements are:

1234 1238 1274 1278

1634	1638	1674	1678
5234	5238	5274	5278
5634	5638	5673	5678

For each quadruplet of movements $CM = \{i, j, k, l\}$, the minimum recovery time is computed using the following formula, which is a direct extension of Gazis' formula:

$$T_{ijkl} = \frac{\sum_{m \in CM} A_m \prod_{n \neq m} s_n}{-\sum_{m \in CM} B_m \prod_{n \neq m} s_n + (1 - L/c) \prod_{m \in CM} s_m}$$

The maximum value for the recovery time, among all possible quadruplets $\{i, j, k, l\}$ is the one that corresponds to the "critical sequence", since that sequence is the one that has the highest capacity requirements, and is chosen as the minimum recovery time. The corresponding appropriate service rates for the critical movements are determined using an extended version of Gazis' formula. Let $\bar{CM} = \{\bar{i}, \bar{j}, \bar{k}, \bar{l}\}$ the set of such critical movements, then

$$\gamma_{\bar{k}} = \frac{s_{\bar{k}} \left[A_{\bar{k}} \prod_{\bar{n} \neq \bar{k}} s_{\bar{n}} (1 - L/c) + \sum_{\bar{n} \neq \bar{k}} (B_{\bar{k}} A_{\bar{n}} - B_{\bar{n}} A_{\bar{k}} \prod_{\bar{m} \neq \bar{n}, \bar{k}} s_{\bar{m}}) \right]}{\sum_{\bar{n} \in \bar{CM}} A_{\bar{n}} \prod_{\bar{m} \neq \bar{n}} s_{\bar{m}}}$$

from which the required green for critical movements can be directly computed as:

$$g_k = \frac{c \gamma_k}{s_k}$$

The computation of green times for non-critical movements follows the same procedure used for the case for undersaturated conditions, described in section 6.1.1.

Gazis suggests that for an intersection serving two competing demands, a solution that further minimizes the aggregate delay can be obtained by "trading off" some delay for the major stream, i.e., the one associated with the higher saturation flow s_1 , for a smaller amount of delay for the other stream. The optimum strategy then is a two-stage operation, also called *bang-bang control*, that involves serving the major stream with the maximum (allowed) green time and the other stream with the minimum green, during the first stage, and reversing the control on the second stage. The "switch-over" point is given by

$$\tau = \frac{\left[\left(\frac{c}{s_1} \right) Q_1(T) - g_{min} T \right]}{g_{max} - g_{min}}$$

where T and $Q_1(T)$ satisfy equations 6.5. In the context of actuated control, for an eight-lane movement (eight streams) intersection, there can be switch-over points for all possible

combinations of conflicting points. Since not all movement are necessarily congested, i.e., not all of them have a discharge rate equal to the saturation flow, this extension would require a probabilistic approach based on the computation of the average green times (and average discharge rates), in order to assess the benefits of a bang-bang control.

6.1.3 Estimation of Average Green Time and Delay

The most widely used models for estimating delay at signalized intersections are those of the Highway Capacity Manual (HCM 1985, HCM 1994). The methods used in this research are based on a revised version, proposed by Fambro & Roupail (1997) and validated by Engelbrecht, Fambro, Roupail & Barkawi (1997) and Roupail, Anwar, Fambro, Sloup & Perez (1997) aimed at addressing certain limitations of the original models. Specifically, the new model takes into account parameters for actuated control, oversaturated conditions, and variable demand distributions. Basic variables for the computation of intersection delay in a traffic actuated control scheme are the average duration of the green time allocated to the movements and, strictly related to them, the average length of the cycle. In order to estimate such variables, a methodology introduced by Akcelik (1994) is used.

In order to compute a suitable average cycle length and green splits, and the delay associated with it, given the new demand distribution, the arrival distribution, and a specified unit extension (this is assumed to be an exogenous variable, depending on the detector setback and the average speed of incoming vehicles), Akcelik's method computes the average green time allocated to a lane movement as follows:

$$g = yc + (1 - y)e_g$$

where

y is the movement flow ratio (ratio of arrival flow rate to saturation flow rate);

c is the average cycle length (to be computed);

e_g is the average extension time, to be computed as:

$$e_g = \frac{\exp[\Lambda(e_0 - \Delta)]}{\Phi q} - \frac{1}{\Lambda}$$

where Λ , Φ , and Δ are parameters of the assumed arrival distribution, q is the arrival flow (veh/sec) and e_0 is the gap time setting, or unit extension.

Akcelik assumes a bunched exponential distribution for the arrival pattern whose cumulative distribution function (CDF) is defined as follows:

$$\begin{aligned} F(t) &= 1 - \Phi \exp[-\Lambda (t - \Delta)] & \text{for } t > \Delta \\ F(t) &= 0 & \text{for } t < \Delta \end{aligned}$$

where

Δ is the minimum headway in arrival stream (sec);

Φ is the proportion of free (un-bunched) vehicles = $\exp[-b \Delta q]$;

Λ is a model parameter = $\Phi * q / (1 - \Delta * q)$;

Calibrations of this distribution (Akcelik 1994) provided the following values for Δ and b :

single-lane: $\Delta = 2.0$; $b = 1.5$;

2-lanes: $\Delta = 1.0$; $b = 1.0$;

multi-lane (> 2): $\Delta = 0.5$; $b = 1.0$;

For lane movements on which a queue exists, the average green time is simply the maximum green allocated to the movements, since as long as queueing vehicles are present, the corresponding phase will always use its maximum allocated time. The average cycle time is the sum of the average green times of the critical movements.

Once the average cycle length and splits have been determined, the average delay that each vehicle will experience for the analysis time interval T , during which demand is assumed to be stable, is computed using Fambro & Rouphail's (1997) generalized formula:

$$d = d_1 + d_2 + d_3$$

where

d_1 is the uniform delay;

d_2 is the incremental delay component due to random and overflow queues;

d_3 is the incremental delay component due to oversaturation queues at start of analysis period.

The first two components of the delay are computed as follows:

$$d1 = \frac{0.5c(1 - g/c)^2}{1 - (g/c)\min(1, X)} PF$$

$$d2 = 900T \left[X - 1 + \sqrt{(X - 1)^2 + \frac{8kIX}{Tc}} \right]$$

where

X is the degree of saturation for the lane group (volume/capacity);

T is the analysis period (expressed in hours);

k is a parameter for the given arrival and service distribution, depending on the unit extension ue and the degree of saturation;

I is a parameter for the variance/mean ratio of arrivals.

In case of oversaturated intersections, the $d3$ component of delay measures the additional delay due to existing queues. This is computed as

$$d3 = (3600 \frac{N}{c}) - 1800 T(1 - \min(1, X))$$

where

N is the initial queue length. The queue length at the congested approaches is an input from the operator. It is assumed that the operator can gather this information, using additional data sources, such as CCTV or incident reports.

6.2 Ramp Metering

Analogously to the case of signal control, the algorithm for the control of free-way access through ramp metering makes use of different strategies, according to global network considerations, and distinguishes between incident-free and congested conditions. The methodologies used are based on simple, state of the practice algorithms, such as the traditional *demand/capacity* strategy (Koble, Adams & Samant 1980), but more sophisticated algorithms, such as ALINEA (Papageorgiou et al. 1991) can be implemented, provided that the interface with the rest of the system is preserved. Two different conditions are considered: a) meter control at freeway on-ramps on alternative paths or at congested on-ramps; b) meter control upstream of an incident.

6.2.1 Ramp Metering on Diversion Routes and at Congested On-Ramps

As noted in the previous section, if bypass paths are sought to provide alternative routes in case of congestion, it is necessary to determine whether those paths can provide enough capacity, and a suitable control strategy that guarantees the required capacity. Analogously to signalized intersections, metered freeway ramps may constitute a bottleneck, thus it is necessary to determine whether a suitable metering rate exists, that satisfies minimum capacity requirements, given the expected demand after diversion. The strategy used, known as the demand/capacity strategy is based on measuring the mainline flow upstream of the ramp q_u , comparing it with the downstream capacity cap_d , and determining a ramp metering rate r proportional to their difference, as shown in Figure 6.4.

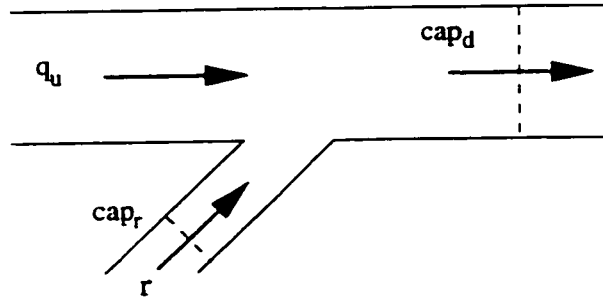


Figure 6.4: Ramp metering variables

The ramp metering rate r (expressed in vehicles/minute) is set to the minimum between the maximum capacity cap_r of the metered link and the difference between the downstream capacity and the upstream volume.

$$r = \begin{cases} cap_r & \text{if } cap_d - q_u \geq cap_r \\ cap_d - q_u & \text{otherwise} \end{cases} \quad (6.7)$$

In order to determine whether the available capacity is enough to justify a diversion control, the corresponding degree of saturation is compared with a pre-specified threshold value, determined by the agency responsible for the control of the meter. If the resulting degree of saturation is higher than the maximum acceptable degree of saturation, the diversion proposal is rejected, since the meter can not serve the diverted demand without giving rise to additional congestion. Otherwise, the corresponding green time ratio is computed as the ratio between the ramp capacity, r and the ramp saturation flow s

$$u = \frac{g}{c} = \frac{r}{s}$$

The specification of the green time and cycle length depends on the control policies adopted by the local authority. The duration of the green can be specified in such a way to allow passage to only one, two, or more vehicles per interval. The traffic light at the meter can be turned off if no control is desired, or set to a continuous green state, to reach the same effect. The policy normally adopted in Southern California is to allow passage of only one vehicle per interval per lane. Minimum and maximum thresholds are specified for the metering rate. If the desired rate is higher than the maximum rate (Caltrans uses 15 vehicles/min/lane), the signal is turned off.

The demand/capacity strategy is also used to increase the ramp capacity in case a queue forms at the ramp and the current rate is judged inappropriate to serve the ramp demand.

6.2.2 Ramp Metering for Congested Freeways

When ramp metering is implemented as a measure to decrease the flow upstream of a congested freeway section, the basic demand/capacity algorithm sets the ramp metering rate to a predefined minimum rate. The green ratio, computed once again as the ratio between the selected capacity (minimum rate) and the ramp saturation flow, determines the setting of the required cycle length and the green time: the cycle length is

$$c = \frac{3600}{min_rate}$$

and the green time, selected in order to allow passage to only one vehicle per interval ($g = 2, 3$ seconds), is

$$g = c \frac{min_rate}{s}$$

which allows to set the cycle time, i.e., to specify the duration of the red time, accordingly.

6.2.3 Delay Computation

The average approach delay per vehicle associated to the selection of a certain metering plan is computed using a formula for isolated fixed-time signals that combines steady-state and saturated conditions, as proposed by Akcelik (1981). The formula (Kimber & Hollis 1979) is essentially the same as the one used in the TRANSYT optimization model (Robertson 1979), based on queueing theory, and it approximates Webster's (1961) formula

for undersaturated conditions. As pointed out by Hurdle (1984), when estimating intersection delay, no formula, among the ones proposed in the literature, yields the “correct” result, but rather they provide answers that do not violate elementary logic in the troublesome region of degree of saturation approaching unity, where neither the steady-state models, nor the oversaturation models can be expected to produce reasonable results.

For each lane movement, the formula estimates the threshold degree of saturation x_0 , below which the average overflow queue is approximately zero, as

$$x_0 = 0.67 + \frac{g}{600}$$

The average overflow queue (the total number of vehicles queued in all lanes) is

$$N_0 = \begin{cases} \frac{QT}{4}((x-1) + \sqrt{(x-1)^2 + \frac{12(x-x_0)}{QT}}) & \text{if } x > x_0 \\ 0 & \text{otherwise} \end{cases}$$

where

Q is the capacity (in vehicles per hour) = $s(g/c)$

T is the analysis period (during which demand is assumed constant)

x is the degree of saturation.

The average delay (expressed in vehicle-hours) is computed as

$$d = \frac{c(1-u)^2}{2(1-y)} + \frac{N_0}{Q}$$

where y is the movement flow ratio.

The computation of N_0 is used to determine whether the current levels of demand and capacity are likely to cause spillback of vehicles. If N_0 is greater than a prespecified maximum queue length, for instance the number of vehicles that can be stored in the ramp without causing spillback, then the metering rate is likely to cause oversaturation.

If a queue has already formed at a congested metered ramp, information about its length, that was requested to the operator at the time the problem is diagnosed, is input in the computation of the delay.

6.3 Traffic Diversion

Traveler information is a fundamental component of ATMIS, especially with respect to congestion management, since it is a means to provide travelers with information describing current network conditions. In particular, in case of non-recurrent congestion, providing information describing unusual network conditions or suggestions for alternative paths, has a potential role in managing demand to match the available capacity, by allowing travelers to perform a better informed routing decision.

Information can be broadcast to travelers in several ways, that range from the use of in-vehicle navigation systems to the transmission of traffic-related messages by commercial radio stations. The use of CMS is a relatively inexpensive and effective way to communicate a limited amount of information to all drivers passing a specific location, that does not requires vehicles to be equipped with on-board guidance devices. The use of CMS has also several drawbacks, mostly related to the limited amount of information that can be carried by each message and to the fixed location of the signs. But in those applications where the main interest focuses on few bi-directional high-volume routes and few feasible alternative routes, such as in most freeway-corridor applications, the use of a limited number of CMS at strategic decision points both on the freeway and along those routes that can be used to bypass a serious incident, provides the potential for broadcasting very relevant information to a wide number of directly affected travelers.

An important issue when dealing with traveler information and in particular with CMS concerns the degree of compliance to information and advisory. As noted in Chapter 2, some research in this field, especially with respect to applications of the Automatic Control Theory (Diakaki et al. 1997, Diakaki et al. 1998, Papageorgiou 1995) is based on the assumption that, once an optimal, "desired" assignment of traffic among alternative routes has been determined, it is possible to select adequate CMS messages that affect traffic in such a way to obtain the desired distribution. Some of these researchers use transformation functions that allow to select the appropriate message according to the desired diversion degree. A correct definition of such functions, though, is a very hard, if not impossible task, since these functions are too coarse and rely heavily on a very accurate estimation of driver response.

The research described in this dissertation takes a different, almost diametrically opposite approach. The DTPS algorithm examines a database of existing CMS settings,

previously defined based on information gathered from TMC operators and local traffic engineers. The “expected effect” of each setting that satisfies specific applicability conditions, is analyzed to determine whether the corresponding control action can help redistribute traffic flow in order to ameliorate traffic conditions around congested areas. Each “useful” setting becomes part of a control action. The selection of the most appropriate control is part of the global DTPS algorithm, and it depends on local and global compatibility issues and on the predicted overall network performance.

The control strategy used by Diakaki et al. (1998), based on the selection of the CMS message whose expected effect is closest to the “desired” one, can lead to the diversion of too high a volume of traffic, potentially causing congestion on the alternative path. Indeed, especially if a small number of CMS messages are available, the transformation function will be a very coarse one. Thus, the gap between the desired effect and the one obtained using the “best” CMS message (the one whose expected effect on traffic is closest to the desired effect), may be high enough to cause the diversion of too high a volume of traffic to the alternative path. Given the highly non-linear behavior of travel time with respect to volume, it is thus not appropriate to select a CMS message whose effect is “closest” to the desired one. Or at least, it is necessary to define the concept of “proximity” that takes into account the non-linear characteristics of traffic flow.

Even if a number of researchers have explored driver’s response to traffic information via CMS, no algorithm or model has yet demonstrated its superiority over the others. According to the literature, estimation of driver response to CMS varies extensively. Several authors report quantitative results, obtained by direct traffic counts (Dudek et al. 1978), or using different modeling (Wardman et al. 1997) or simulating methodologies (Brocken & van der Vlist 1991), that vary between 10% and 90%. Several variables have been identified as determinant for affecting driver response and route change. These include the expected delay on the traditional route, the type of the problem causing the delay (e.g., whether delay is due to an accident, to road works, etc.), traffic conditions on alternative routes, the provision of direction for alternative routes, and previous experience with the reliability and accuracy of the advisory information. Dudek et al. (1978) report measurements of diverted flows induced by CMS messages in occasion of some special events. Response varies depending on the type of the message (traffic-state descriptor messages have a higher compliance), the type of the problem, and, in general it increases when potential delay savings are provided. The percentage diversion reported by Dudek et al. (1978), measured during

three special event experiments, is shown in Table 6.1. where the basic model is given by

Table 6.1: Measured percentage diversion

basic model	explanatory	delay levels
29.8 ÷ 42.8	46.4 ÷ 66.0	16.8 ÷ 57.9

CMS messages that simply suggest a diversion as a better alternative to reach a destination, the explanatory model contains information describing the reason for congestion, and the delay levels model provides indications on the expected delay on the traditional route.

Wardman et al. (1997) report results from a model based on a stated preference approach. These results confirm that information about the amount of expected delay and the stated cause are significant variables: delays due to accidents have the highest impact, while if no cause is quoted the messages have a relatively low effect. A summary of such results is reported in Table 6.2.

Table 6.2: Modeled percentage diversion according to cause and delay

	5 min	10 min	20 min	30 min	likely	long
accident	9	24	62	89	59	84
no reason	6	17	46	74	22	72
road work					15	74
congestion					53	68

The development of a sophisticated driver behavior model falls outside the scope of this research, but since an estimation of driver response to CMS is necessary for the control algorithm, a basic model was developed. As in the case of signal and ramp control, the modularity of the architecture provides the possibility of replacing it with more sophisticated and accurate models, when available.

The selection of suitable CMS messages is based on the analysis of a database of frame objects. Each object, or *CMS-frame* contains information related to the use and usability of a coherent set of CMS messages, i.e., a set of CMS messages that provide indications along a certain route, and that are consistent with each other. Three types of information are particularly relevant in this respect:

Control action is a list of CMS settings for a set of CMS. It is expressed as a list of couples $\langle CMS, message \rangle$. The control action does not necessarily contain a setting for

all CMS on the network, but only for those used to redirect travelers along a set of alternative paths.

Preconditions is a logical expression that specifies conditions that have to be verified for the CMS-frame to be taken into consideration. Conditions are statements of various type that the system is able to verify. They may regard the state of the network (the status of a link, of a path or a set of paths, etc.), the level of demand, the time of day, the presence of special events, etc. The verification of conditions allows the system to quickly determine whether a certain frame is applicable to a particular situation, before getting involved in the more onerous task of determining its effect on traffic.

Diversion effect is a description of the effect that the corresponding CMS messages are expected to produce on the current traffic distribution. It is expressed as a list of *diversion* objects. Each diversion describes the original and the alternative path for which diversion is expected, and contains a method to estimate drivers compliance to the set of CMS messages, given the current network conditions, based on a simple driver behavior model. This method uses a parametric compliance procedure, defined as part of the knowledge of the frame. Whenever the effect of a frame must be determined, the corresponding compliance procedure is invoked using as arguments the type of the problem that the system is trying to address and the estimated average delay per vehicle incurred to travelers. The procedure returns a percentage compliance rate which, when applied to the current demand, causes a redistribution of traffic from the original paths to the alternative ones, whose amount depends on the estimated current demand on the original paths. The definition of a compliance function for each CMS-frame allows to represent the variability of the response according to the location of the CMS. For instance, CMS messages that aim to divert traffic from the freeway may prompt a different response compared to those attempting to divert traffic on the arterial network, even given the same type of problem and the same expected delay. The compliance procedures used in the model currently return a compliance rate ranging between 5% and 65%. Additional testing and calibration based on observation of real-data may be used for their fine-tuning.

The simple driver behavior model currently used by the DTPS algorithm does not require any additional data exchange among the agents, since the necessary information,

namely the type of the problem and the average expected delay incurred by vehicles, at the time this information is used are already part of the knowledge base of both agents. A more sophisticated model may require additional data, such as, for example, the current travel time on alternative paths, which in part depends on the control implemented, and therefore may rely on information local to another agent. If a more sophisticated model is deemed necessary, a possible solution may require the agents to exchange this kind of data, but as seen in Chapter 5, the more information is exchanged among the agents, the higher is the potential for computation and synchronization time increase. A more approximate but faster solution (less *completely accurate* and more *functionally accurate*) might involve the use of approximate or estimated data, based on the available information and the refinement of such data at a later moment in the solution process, once correct data becomes available. Once again, the tradeoff is between accuracy on one side and speed of execution and computability on the other.

6.4 Demand Computation

When attempting to assess the demand over the networks, as noted in the introductory chapter, the following issues must be taken into account:

- in order to determine suitable diversion strategies, path-related information describing network demand, rather than link-related information is often needed;
- under congested conditions, data describing volumes measured by loop detectors do not always yield a measure of the true demand, but rather of the available capacity;
- loop detectors, especially on the arterial network, are available only for a limited subset of the network links.

Because of the above considerations and the necessity of a basic approximate estimation of the network demand, the result of a time-varying static traffic assignment is used as a base “backbone” demand distribution, but heuristic techniques are also applied to make use of real-time measured data, whenever available, for data completion and update.

The development of a sophisticated *data completion* module is beyond the scope of this research, but simple techniques that adjust the backbone demand using measured traffic data are implemented as follows.

When computing the demand that is to be expected at a signalized intersection (by lane movement) or at a metered ramp as a result of a diversion control action, the following three types of volume measures are considered.

- **The measured demand:** if a link l is equipped with system loop detectors, the measured demand is the average volume of traffic crossing that link, as measured by the set of detectors. The detectors transmit hourly volumes as projections of 30 seconds sets of data. Each agent, when receiving detector data, averages the input data across the measurements of the last 5 minutes, thus producing a smoothened rolling horizon estimate of the most updated data. In general, if measuring the demand at a signalized intersection, depending on the type and location of the detectors, it is not always possible to obtain disaggregated volumes for the various lane movements, but rather total link volumes. The volume measured by detectors on a link l from which movements i and j start, is named $meas_{ij}$.
- **The Δ -demand,** i.e., the number of vehicles (per hour) that are expected to divert from their original route as a result of the diversion control action. For a lane movement i , the Δ -demand is called Δ_i .
- **The backbone demand,** computed as the summation of the demand on all the paths that cross the location under analysis. This quantity includes the expected Δ -demand, i.e., it takes into consideration the expected effect of a diversion control action. In order to estimate the expected demand prior to the diversion effect, which is compared to the currently measured volumes, it is thus necessary to subtract the Δ -demand from it. The backbone demand on on lane movement i from link l is named dem_i .

The adjustment of the current demand to estimate the future distribution is then computed as follows: for lane movements i and j (left and through movements), starting from a link l , if a detector exists on link l , the total demand among the two lane movements that should be currently expected on link l (i.e., before the diversion is initiated) is

$$cur_{ij} = dem_i - \Delta_i + dem_j - \Delta_j \quad (6.8)$$

The volume on lane movement i is estimated as

$$vol_i = \begin{cases} meas_{ij} \frac{dem_i - \Delta_i}{cur_{ij}} + \Delta_i & \text{if } cur_{ij} > 0 \\ dem_i = \Delta_i & \text{otherwise} \end{cases} \quad (6.9)$$

This algorithm partitions the demand measured on a link across its lane movements, according to the demand split resulting from the backbone traffic assignment. It also assigns the additional (positive or negative) demand as expected from the diversion control.

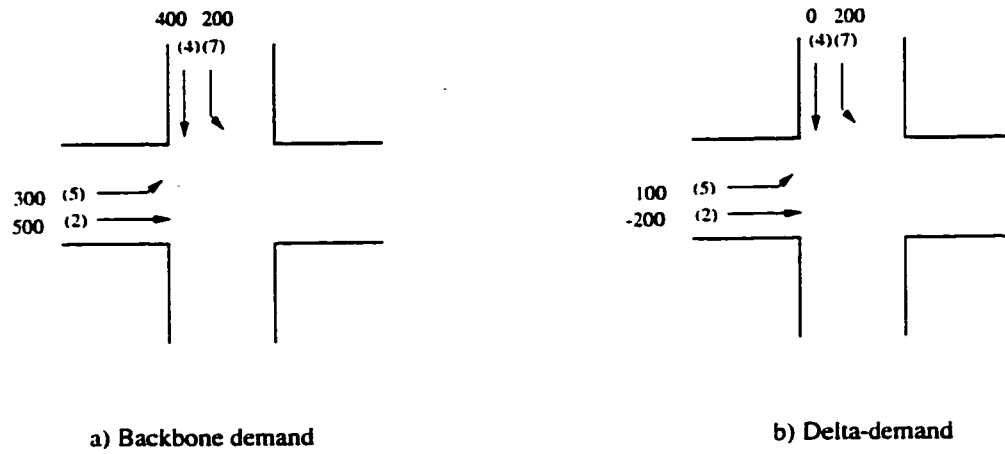


Figure 6.5: The backbone and Δ -demand at an intersection

For the estimation of demand on metered ramps resulting from diversion control, the procedure is analogous, but simpler since there is only one movement, i.e., the link demand is not split among two movements.

As an example, consider the situation illustrated in Figures 6.5 and 6.6. The backbone demand, the Δ -demand, and the expected current demand, for lane movements 2, 4, 5, and 7, expressed in vehicles per hour, are shown in Table 6.3.

Table 6.3: The backbone, Δ -demand, and expected current demand (veh/h)

movement	backbone demand	Δ -demand	expected current demand
2	500	-200	700
5	300	100	200
4	400	0	400
7	200	200	0

In the example, the diversion control is assumed to determine a shift of 200 veh/h from lane movement 2, of 100 veh/h to lane movement 5, and of 200 veh/h to lane movement 7. Note that the algebraic sum of the volumes diverted is not necessarily null for an intersection, since traffic may be diverted through other intersections as well. The expected current demand, aggregated across movements on the same approach, is computed using equation 6.8, which yields:

$$\begin{aligned} cur_{25} &= 500 - (-200) + 300 - 100 = 900 \\ cur_{47} &= 400 - 0 + 200 - 200 = 400 \end{aligned}$$

The expected current demand represents the volume that should be measured by the detectors if the traffic load assignment were exact. In reality, due to the approximate nature of the assignment and random fluctuations in the measured volumes, a difference is to be expected. The measured volumes at the two approaches under analysis are 600 veh/h and 400 veh/h, as shown in Figure 6.6. Comparing these values respectively with cur_{25} and cur_{47} , it is possible to note that the backbone demand (and therefore the currently expected) overestimates the volumes at approach (2,5) (of a 3/2 factor) and is correct at approach (4,7).

The adjusted volumes, as computed using equation 6.9, are:

$$\begin{aligned} vol_2 &= 600 \frac{500+200}{900} - 200 = 266.7 \\ vol_5 &= 600 \frac{300-100}{900} + 100 = 233.3 \\ vol_4 &= 400 \frac{400-0}{400} + 0 = 400 \\ vol_7 &= 400 \frac{200-200}{400} + 200 = 200 \end{aligned}$$

This example highlights the approximate nature of the approach. No adjustment is made on the diverted volumes. In other words, it is assumed that the information about the diverted volumes (the Δ -demand) is accurate. A more exact procedure would require to process data in such a way to obtain an adjusted, more accurate estimation of the diverted demand as well. This issue becomes more obvious if it is assumed, for instance, that the demand at movement 2 belongs to only one path. In this case it would be possible and more correct to adjust the Δ -demand at lane movement 2 Δ_2 , and conclude that

$$\begin{aligned} \Delta_2 &= 200 \frac{2}{3} = 133.3 \\ vol_2 &= 500 \frac{2}{3} + 200 \frac{2}{3} = 333.3 \end{aligned}$$

The problem with such a procedure is twofold. First, the computation of cur_{25} , which is used to adjust Δ_2 , makes use of Δ_2 itself, in a mutually recursive fashion. One way

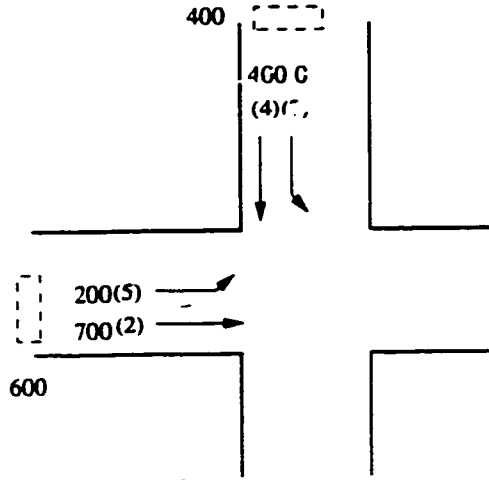


Figure 6.6: The currently expected demand and measured volumes at an intersection

to address this problem is to employ an iterative method for the search of an equilibrium solution, not unlike the traditional procedures for network traffic assignment. The second and more important issue is that, in general the demand at an intersection approach is the sum of the demand of all paths that cross that approach, and the adjustment of the diverted demand should take into account adjustments made to the demand of *all* paths across the network.

The adjustment of path-related demand using measured link volumes is a much more complex equilibrium process, that requires matching, the measured volumes with the sum of the demand of all possible paths that pass through that link. An approximate way of tackling the problem would be the following. Let n be the number of paths across the network, and m the number of links for which measured data is available. Let p_i ($i = 1 \dots n$) and d_j ($j = 1 \dots m$) be respectively the expected demand on path i and the measured demand on link j . Defining α_i as a multiplying factor that increases or decreases the demand on path i in order to match the observed volumes ($\alpha_i \geq 0$), and $[\lambda_{ij}]$ as a path-detector incident matrix, with $\lambda_{ij} = 1$ if path i passes through detector j and $\lambda_{ij} = 0$ otherwise, an adjusted path-demand distribution can be obtained by solving the following problem, which can be easily transformed into a linear programming problem:

Problem: Minimize

$$\sum_{j=1}^m \left\| \sum_{i=1}^n \alpha_i p_i \lambda_{ij} - d_j \right\|$$

subject to

$$\alpha_i \geq 0 \quad i = 1 \dots n$$

The main problem with this procedure is that it relies on the assumption that the list of paths is complete, i.e., that only paths 1 through n are responsible for the demand measured on the network. For the procedure to be "exact", *all* possible paths should be considered. Another drawback of this methodology is the fact that the procedure relies on the availability and correctness of data on a sufficiently high number of links, which, in a real-world scenario, is not always guaranteed. Given the highly approximate nature of this procedure, and the relative uncertainty over response to diversion, this feature is currently not implemented in the DTPS algorithm.

6.5 Treatment of Uncertainty

As noted earlier, the problem solving process is affected by the existence of several sources of uncertainty. Uncertainty is related to the lack of global knowledge, due to the distributed organization of data and problem-solving. Other sources of uncertainty are related to the absence of exact models for the interaction between vehicles and for driver response to traffic control and traveler information, and in general to all those real-time, stochastic phenomena, typical of human behavior, that make the analysis of traffic so complex and challenging.

Treatment of uncertainty in traffic control evolve along two different but related directions. On one hand, modeling techniques are continuously improved and more and more refined models of traffic behavior are developed. On the other, uncertainty is embodied in the reasoning process, through the definition of probability-based reasoning techniques. Even though the development of sophisticated models of traffic behavior is, in general, beyond the scope of this work, the system's adaptability can be greatly enhanced if the system is provided with a certain degree of flexibility for treating information.

In order to deal with the uncertainty associated with driver's compliance to traffic advisory, a probabilistic approach can be taken to verify the suitability of alternative routes. As previously noted, the selection of a CMS message may require the satisfaction of certain conditions, like the presence of adequate capacity on the paths proposed as an alternative. Satisfaction of these conditions is translated into the verification of the ability to provide a suitable LOS at the controlled junctions along the alternative paths - given the demand

distribution resulting from the diversion - and into the design of signal plans that achieve the best LOS. The stochasticity of such phenomena can be modeled by representing the volumes arriving at intersection as multivariate normally distributed random variables, rather than as a set of single numbers, and computing the probability that a suitable practical degree of saturation can be found. Simplifying assumptions on the correlation between arrival volumes on the different movements and at different intersections can be initially introduced but they may be released at a later moment, as a consequence of empirical analysis and model calibration.

As described in section 6.1, the basic step for the computation of the maximum green times and the practical cycle length, given a threshold degree of saturation, involves identifying the critical movements. If the flow ratios y_i (and therefore the green time ratios $u_i = y_i/x_{p_i}$) were assumed to have a known non-zero-variance probability distributions, it would be possible to calculate the probability distribution of the optimum and practical cycle times, or more simply, the probability that the sum of the critical movement flow ratios or green time ratios is lower than a given threshold.

Considering the four possible sequences of critical movements describes in section 6.1, the probability that the maximum among the four sums is less than a threshold y_0 can be computed. Let y_i be a random variable describing the probability of the flow ratio at movement i , and

$$V_1 = y_1 + y_2 + y_3 + y_4$$

$$V_2 = y_1 + y_2 + y_7 + y_8$$

$$V_3 = y_5 + y_6 + y_3 + y_4$$

$$V_4 = y_5 + y_6 + y_7 + y_8$$

the following probability must be computed:

$$P(\max_{i \leq 4} V_i \leq r_0)$$

The problem lies, as usual, in determining an appropriate distribution for the flow ratios. In first approximation it can be assumed that the variables are distributed according to a multivariate normal distribution. This assumption is justified by considering that the magnitude of the flow arriving at an intersection is a combination of many factors. By

applying the central limit theorem, the resulting value can be assumed to be normally distributed (even though, distributions of capacitated phenomena are sometimes skewed, and may assume a different form, due to the presence of the upper limit given by the capacity). The mean μ_i and variance σ_{ii} of each variable are assumed to be known. They can be estimated through empirical analyses, executing test simulations and aggregating the results. The covariance between the movement flow ratios can be initially assumed as follows: flow ratios of movements belonging to the same link group have a non-zero covariance σ_{ij} , while flow ratios of movements belonging to different link groups can be assumed to be independent, i.e., $\sigma_{ij} = 0$. In order to reduce the complexity of the problem, it is noted that in all the 4 movement combinations shown above, conflicting movements are always in pairs (and they are always independently distributed). This allows to define the following variables:

$$Z_1 = y_1 + y_2$$

$$Z_2 = y_3 + y_4$$

$$Z_3 = y_5 + y_6$$

$$Z_4 = y_7 + y_8$$

which have a multivariate normal distribution with a mean μ_Z and a variance Σ_Z . Then, the pairwise sum variables

$$V_1 = Z_1 + Z_2$$

$$V_2 = Z_1 + Z_4$$

$$V_3 = Z_3 + Z_2$$

$$V_4 = Z_3 + Z_4$$

are also multivariate normally distributed with a mean $\mu_V = [\mu_{V_i}]$ and a variance $\Sigma_V = [\sigma_{V_{ij}}]$.

$$\mu_V = \begin{pmatrix} \mu_1 + \mu_2 + \mu_3 + \mu_4 \\ \mu_1 + \mu_2 + \mu_7 + \mu_8 \\ \mu_3 + \mu_4 + \mu_5 + \mu_6 \\ \mu_5 + \mu_6 + \mu_7 + \mu_8 \end{pmatrix}$$

$$\Sigma_V = \begin{pmatrix} \sigma_{11} + \sigma_{22} + \sigma_{33} + \sigma_{44} & \sigma_{11} + \sigma_{22} + \sigma_{38} + \sigma_{47} & \sigma_{16} + \sigma_{25} + \sigma_{33} + \sigma_{44} & \sigma_{16} + \sigma_{25} + \sigma_{38} + \sigma_{47} \\ & \sigma_{11} + \sigma_{22} + \sigma_{77} + \sigma_{88} & \sigma_{16} + \sigma_{25} + \sigma_{38} + \sigma_{47} & \sigma_{16} + \sigma_{25} + \sigma_{77} + \sigma_{88} \\ & & \sigma_{33} + \sigma_{44} + \sigma_{55} + \sigma_{66} & \sigma_{55} + \sigma_{66} + \sigma_{38} + \sigma_{47} \\ & & & \sigma_{55} + \sigma_{66} + \sigma_{77} + \sigma_{88} \end{pmatrix}$$

The probability distribution of the maximum among the variables V_i , is computed by applying Clark's approximation (Clark 1961). Define:

$$\begin{aligned}\tilde{V}_2 &= \max(V_1, V_2) = \\ \tilde{V}_3 &= \max(\tilde{V}_2, V_3) = \max(V_1, V_2, V_3) \\ \tilde{V}_4 &= \max(\tilde{V}_3, V_4) = \max(V_1, V_2, V_3, V_4)\end{aligned}$$

and $\forall i = 1 \dots 4$, the moments of $\tilde{V}_i$ are iteratively computed as follows: the first moment (mean) of $\tilde{V}_i$ is

$$\tilde{m}_i = m_i + (\tilde{m}_{i-1} + m_i)\Phi(\alpha_i) + a_i\phi(\alpha_i)$$

The second moment of $\tilde{V}_i$ is

$$\tilde{m}_i^2 = m_i^2 + \sigma_{ii} + (\tilde{m}_{i-1} + \tilde{\sigma}_{i-1i-1} - m_i^2 - \sigma_{ii})\Phi(\alpha_i) + (\tilde{m}_{i-1} + m_i)a_i\phi(\alpha_i)$$

where

$$a_i = \sqrt{\sigma_{i-1i-1} + \sigma_{ii} + 2\tilde{\sigma}_{i-1j}\alpha_i} = \frac{m_{i-1} - m_i}{\alpha_i}$$

The variance of $\tilde{V}_i$ is

$$\tilde{\sigma}_{ii} = \tilde{m}_i^2 - \tilde{m}_i^2$$

and the covariance between $\tilde{V}_i$ and $\tilde{V}_j$ is

$$\tilde{\sigma}_{ij} = \sigma_{ij} + \tilde{\sigma}_{i-1j} - \sigma_{ij}\Phi(\alpha_i)$$

where Φ and ϕ indicate respectively the normal probability distribution and its density function.

The formulas above allow to compute the probability distribution of $\tilde{V}_4$, the maximum among the given variable, thus enabling to determine the probability that the maximum critical movement combination is lower than an acceptable threshold. Once that is known, for all signalized intersection along a certain path, the joint probability (i.e., the

product, if they are assumed to be independent), yields the probability that the path is available for rerouting.

An interesting extension of this problem would be to study the relationship between the flow ratios along a certain path. That requires to consider movements at different intersections, or the intersection LOS as mutually dependent variables.

Currently the system does not perform these computations, but it represents volumes as random variable, defined by a mean and a variance (assuming a normal distribution). At the moments all variables have 0 variance, but this capability can be exploited to reduce the uncertainty associated with the assessment of the compatibility among control action, using the described probabilistic approach.

Chapter 7

Performance Evaluation

The main objectives of this research involved developing a new, efficient and effective distributed problem-solving approach to the problem of network-wide non-recurring congestion management, based on functionally accurate cooperation among distinct decision-making entities. This task was accomplished through the development and evaluation of a distributed software system based on such approach.

This chapter describes the steps taken to evaluate the distributed system, and the results of the evaluation process, aimed at providing a quantitative and qualitative assessment of the system's applicability and effectiveness.

For analogous reasons to what described in Chapter 3, a simulation-based methodology was considered appropriate to obtain quantitative and qualitative measures of the effectiveness of the approach. The analysis of the simulated network performance under different traffic conditions, determined by the occurrence of several types of incidents, and the comparison between network-wide measures of effectiveness (MOEs), with and without the implementation of the control plans suggested by CARTESIUS, allowed to estimate the validity of this distributed problem solving approach. Of foremost concern was the assessment of the travel time reduction obtained by the implementation of the incident response plans suggested by the two agents. Another important measure of effectiveness was the system's response time. It was also interesting to assess the impact of the integration between the various control components, by comparing the effect of fully integrated control plans (freeway traffic diversion and arterial signal control) to incomplete control plans, that used exclusively traffic diversion schemes.

The remainder of this chapter first describes the site for which the distributed system was developed, with details on how real-world data was incorporated, whenever

possible, within the simulated environment. Then it reports on the quantitative and qualitative results of the validation process.

7.1 The Test Site

As part of the Testbed Research Implementation Control and Evaluation Prototype System (TRICEPS) project, the CARTESIUS system was developed for a highly-congested corridor network in the city of Irvine, in Orange County, that includes 4-mile sections of the Interstate 5 and 405 Freeways, the SR-133 Freeway, and the adjacent sub-network of surface streets, as shown in Figure 7.1. TRICEPS is a research project aimed at developing a software platform that facilitates the testing and evaluation of a wide range of algorithms for adaptive traffic control, advanced transportation management systems and advanced traveler information with simulated or real-world data.

The network under analysis is larger than the site of the FHWA City of Irvine FOT, described in Chapter 2. The inclusion of a section of the I-5 and the SR-133 Freeways, as well as the arterial connecting these freeways to the I-405 allowed to test the feasibility of diversion strategies that included freeway-to-freeway diversion.

The rapid urbanization occurring in Orange County has resulted in land use patterns which are characterized by a number of commercial activity centers surrounded by moderate density residential development. During the period from 1990 to 2020 the employment level for Orange County is projected to nearly double. The majority of these jobs will be added to existing activity centers in the central region in addition to significant growth occurring in the Anaheim Stadium area, South Coast Metro, Irvine Business Center and the Irvine Spectrum. Most of these centers are located within the central portion of the county and within one mile of a major freeway.

The City of Irvine Traffic Management Center (ITRAC) is responsible for traffic operations in the City of Irvine. ITRAC's computer-aided traffic system controls over 240 signalized intersections, 32 of which are within the TRICEPS network, and 5 arterial CMS, on Alton Parkway and Irvine Center Drive, the two major arterials in the TRICEPS network. Signal control is fully actuated, and signal control parameters (minimum and maximum green, phase recall, etc.) are set according to a time-of-day basis. ITRAC also has control over 30 CCTV cameras located at major intersections and connected to the TMC through a fiber-optic network.

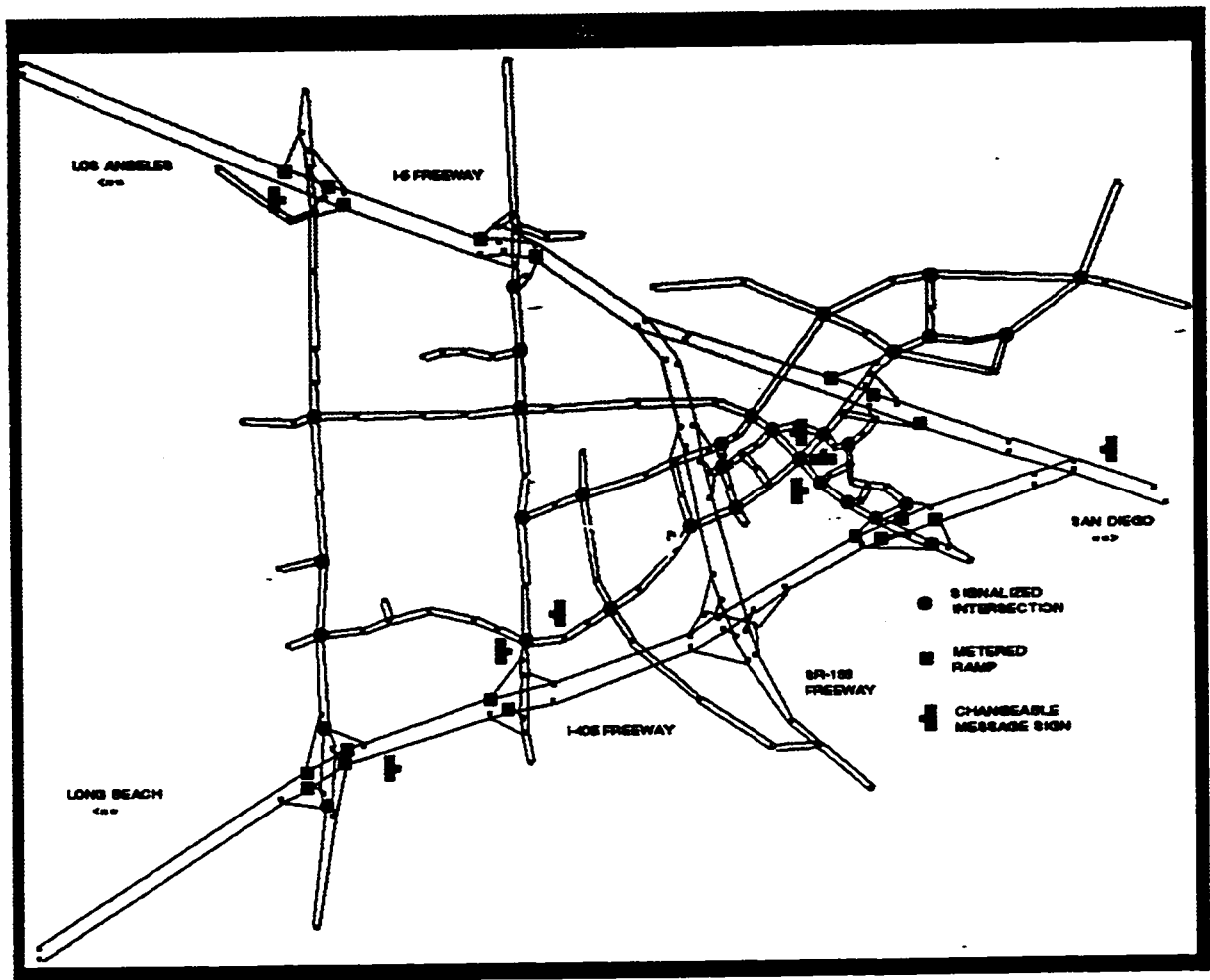


Figure 7.1: The test site in Irvine, California

Caltrans District 12's (Orange County) Advanced Traffic Management System (ATMS) uses state-of-the-art computer, software, and communication systems to manage the flow of traffic on the Orange County freeway network. Vital elements at the core of the system's operations include: 30 CCTV cameras, 34 CMS, the Highway Advisory Radio, 278 metered on-ramps, 1,098 incident call boxes and 258 directional miles of loop detectors. Within the subnetwork for which CARTESIUS was developed, Caltrans controls 3 CMS. Meter control is performed on all 18 freeway on-ramps within the network.

7.1.1 Real-Time Connection with Traffic Management Centers

The two agents within CARTESIUS are able to receive real-time traffic data from the Caltrans District 12 ATMS, through the California ATMS Testbed communication network, a wide-area communications network backbone linking the Cities of Anaheim and Irvine TMCs to the Caltrans's District 12 TMC and to the ATMS Research Laboratories at the Institute of Transportation Studies, University of California, Irvine.

A CORBA-based intertie between the Caltrans District 12 ATMS and the UCI ATMS Laboratories has been established, to provide real-time communication and messaging interface capabilities for the exchange of traffic management data and control requests (NET 1997).

The Common Object Request Broker Architecture (CORBA) (OMG 1995) is a standard for distributed object representation and communication, developed by the Object Management Group (OMG), a large consortium of software vendors, developers and users, to address the need for inter-operability among the rapidly proliferating number of hardware and software products available today. CORBA allows applications to communicate with one another in a heterogeneous distributed environment, independently from the physical location of their host systems and their design. CORBA's design is based on OMG Object Model, which defines common object semantics to specify the externally visible characteristics of objects in a standard and implementation-independent way. According to this model *clients* request services from *servers* through a well-defined interface. This interface is specified in OMG IDL (*Interface Definition Language*). A client thus accesses an object by issuing a *request* to the server. The request is an event, and it carries information that includes the specification of an operation, the object reference of the service provider, and actual parameters. The object reference is an object name that unequivocally defines an object.

Within the intertie architecture, CORBA clients have been developed to provide applications running at the UCI ATMS Laboratories with real-time traffic and control data. In particular, one client was developed for interaction with the freeway agent in CARTESIUS, to query the Caltrans ATMS data server for real time traffic data, and transfer such data to the agent, through an object-oriented, TCP/IP-based real-time connection. The communication with Caltrans District 12 ATMS provides the freeway agent with loop detector data (volume and occupancy), with CMS and ramp metering data, and with the

ability to transmit, subject to Caltrans's approval, ramp metering rate control. A real-time connection between the arterial agent and ITRAC is currently under development and could not be used in this research. It will allow the arterial agent to receive traffic and control status data from the arterial system, and to transmit, subject to ITRAC approval, alternative signal timing and CMS setting plans.

The two agents in CARTESIUS were developed using G2 (Gensym 1995) (version 4.1). G2 is a real-time knowledge-based system shell designed for the development and execution of complex applications that require intelligent monitoring and control in real-time. G2 is an object-oriented tool that allows one to define knowledge bases using several knowledge representation and inference techniques, including rule bases, procedures and frame matching. It also provide means to develop remote real-time communication between several knowledge bases.

7.2 The Simulated Environment

Prior to experimentation in the field, evaluation must be conducted in a simulated environment. The traffic simulator Dynasmart (Jayakrishnan et al. 1994), described in Chapter 3 has once again been used for simulation. For the communication with the agents, Dynasmart has been provided with an interface that simulates the functions of the Caltrans CORBA server. This allows to define the client interface of the two agents independently from the server they are communicating with. In other words, traffic and control device data (detector, CMS, signal, and ramp metering data) are exchanged between the agents and the server using exactly the same interface on the client side.

A time-varying OD matrix was estimated for input to the simulation (Kulkarni 1997), based on data from the Irvine Transportation Analysis Model (ITAM) (JHK & Associates 1993), using the TRANPLAN (UAG 1995) and CONTRAM (Leonard et al. 1989) software packages. ITAM is a data package that involves the traditional four-step process of trip generation, distribution, mode choice and assignment. ITAM is composed of a regional model, that encompasses the whole Southern California network (Los Angeles, Orange, Riverside, San Bernardino, and Ventura counties), and of a local model, which is a subsection of the regional component that covers the City of Irvine and adjacent areas, consisting of approximately 600 zones. The network under analysis within the TRICEPS project is a subset of the local component, thus only data related to the local component

was used for the generation of the OD matrix.

The trip generation process used 1991 person-trip rates for identified land uses and a data file corresponding to 1995 land use distribution, to produce updated person trip ends for the various zones. The trip distribution process was based on a calibrated gravity model, that used travel time as impedance, and resulted in person trip tables for the morning (AM), evening (PM), and off-peak hours. Based on the ITAM model, trips were then distributed among various modes (drive-alone, high occupancy vehicle (HOV), and transit, with a vast majority assigned to the drive-alone mode). Trips were then incrementally assigned to the network. The generated volumes, both on freeways and surface streets, were then checked against ITAM screenlines and Caltrans volume counts from 1991, and were found to be within tolerable limits (Kulkarni 1997). The subarea for the network under analysis, containing 320 nodes (including 106 zones, 85 internal and 21 external), and 679 links (including centroid connectors) was then extracted from the ITAM local network, and a new traffic assignment process was performed to load the subarea network. Volumes on arterials and freeways were found to closely match those obtained from the assignment on the local ITAM network and thus were considered appropriate for further use in the project.

The estimation of the dynamic OD matrix for a 3-hour PM-peak, specifying a path-based demand distribution aggregated in 15-minute intervals, was performed using a standard iterative process that involves, at each step, distributing the OD demand obtained from the previous step across the network, using observed vehicle counts previously collected for a subset of the network links, along major arterial and freeway sections (75 one-way links, in this case). This process uses the transportation software packages CONTRAM and COMEST. CONTRAM assigns the OD demand to paths in the network in packets of vehicles. COMEST uses those data and observed volumes (from detector counts) to provide updated 15-minute assignments that attempt to match the observed data. The updated assignment is then input again into CONTRAM and the process is repeated until convergence (small changes in the output produced by COMEST in consecutive iterations) is reached. The average difference between the estimated and the observed link counts, obtained by the assignment was 16%, that was considered an acceptable tolerance.

The time-varying path distribution resulting from this process was provided as input to the two agents, for the creation of the path-based, backbone network demand, described in Chapter 5.

The default PM-peak settings for signal control were obtained from ITRAC, and

encoded in the arterial agent. When knowledge was elicited from ITRAC, signal control was based on time-of-day, traffic-actuated control, according to which signal control parameters (minimum and maximum green, unit extension, etc.) vary according to the time of the day.

Analogously, default PM peak ramp metering rates were elicited from the Caltrans District 12 TMC, and embedded in the knowledge of the freeway agent.

7.3 Validation Results

At the core of the validation process was the assessment of the system's ability to provide opportune traffic control plans in response to the occurrence of incidents. Total and average travel time and traveled distance, were considered suitable MOEs, both because they provide an indication of the network level of service and because they are easily measurable using a simulator.

7.3.1 Network Performance

A set of 18 test scenarios was created, by running simulations of 90-minute peak periods and artificially injecting incidents (temporary reductions in the capacity of a link), with varying characteristics, such as the incident location, the associated loss of capacity and the duration of the capacity reduction. For each scenario, the MOEs provided by the simulator (average and total travel time and traveled distance), were collected. For each test case, two simulations were executed: one, the *before* case, using the default control (no CMS message and the default, time-of-day signal and ramp meter timing plan), and one, the *after* case, implementing the control suggested by the agents, in response to the notification of the occurrence of congestion. The comparison of the network performance, through the implementation of the two forms of control, provided a measure of the performance increase that can be expected when the default control is substituted with the control plans suggested by CARTESIUS.

A list of the major characteristics of the test scenarios is described in Table 7.1. In scenario 0 no incident was injected, in order to obtain estimations of the *basic* network average and total travel time and distance. Such estimations are used to assess the performance deterioration caused by the occurrence of incidents.

Given the disparity between traffic volumes on the freeway and the arterial sub-systems, CARTESIUS is primarily concerned with management of incidents occurring on

freeways, thus test scenarios involved mainly incidents occurring on both directions of the two major freeways, the Interstate 5 and 405.

Scenarios 1, 2, and 3 correspond to the occurrence of an incident on the northbound 405 freeway, with different durations and capacity reductions, thus having different impacts on traffic. Scenarios 4, 5, and 6 correspond to the occurrence of an incident on the southbound 405 freeway. Scenarios 7, 8, and 9 describe an incident occurring on the northbound 5 freeway, while scenarios 10, 11, and 12 are related to an incident on the southbound 5 freeway. Scenarios 13, 14, and 15 describe an incident occurring on respectively arterials (Alton Parkway) and on the northbound 405 on-ramp at Irvine Center Drive.

Scenarios 16 and 17 describe cases for which two incidents were injected at the same time. In both scenarios, the first incident was simulated on the freeway (405 north and southbound), while the second was injected along one of the major paths that had been chosen by the system as a bypass for the first incident. The purpose of these tests was to analyze the system's response to multiple incidents with interacting effect on each other.

Table 7.1: Distributed system: incident characteristics in the test scenarios

#	Location	Lanes	Length
0	no incident		
1	405 NB	1(4)	20
2		2(4)	20
3		2(4)	25
4	405 SB	1(4)	20
5		1(3)	20
6		2(4)	20
7	5 NB	2(5)	15
8		3(5)	15
9		3(5)	20
10	5 SB	1(5)	15
11		2(5)	20
12		3(5)	30
13	Alton EB	2(3)	20
14	Alton WB	2(3)	20
15	on-ramp	1(2)	15
16	405 NB	2(4)	20
	Alton WB	1(2)	15
17	405 SB	2(4)	20
	on-ramp	1(2)	10

Table 7.2 reports on the measured total and average travel time in each scenario, for the before and after case.

Table 7.2: Distributed system: simulated average and total travel time

#	Incident Characteristics			Average (min/veh)		Total ($\times 10^4$ veh-hrs)	
	Location	Lanes	Length (min)	Before	After	Before	After
0	no incident			5.24	5.24	0.380	0.380
1	405 NB	1(4)	20	5.87	5.87*	0.429	0.429
2		2(4)	20	6.76	6.32	0.496	0.463
3		2(4)	25	7.20	6.70	0.528	0.486
4	405 SB	1(4)	20	7.47	6.54	0.546	0.479
5		1(3)	20	6.86	5.85	0.502	0.428
6		2(4)	20	7.98	7.21	0.584	0.528
7	5 NB	2(5)	15	5.30	5.30*	0.387	0.387
8		3(5)	15	5.55	5.26	0.400	0.383
9		3(5)	20	5.81	5.33	0.426	0.391
10	5 SB	1(5)	15	5.24	5.24*	0.385	0.385
11		2(5)	20	7.05	5.97	0.510	0.438
12		3(5)	30	8.69	8.38	0.634	0.615
13	Alton EB	2(3)	20	5.26	5.26	0.381	0.381
14	Alton WB	2(3)	20	5.26	5.26*	0.382	0.382
15	on-ramp	1(2)	15	5.26	5.24	0.383	0.381
16	405 NB	2(4)	20	6.78	6.35	0.496	0.459
	Alton WB	1(2)	15				
17	405 SB	2(4)	20	8.09	7.57	0.601	0.560
	on-ramp	1(2)	10				

Table 7.3 reports on the measured total and average traveled distance in each scenario, for the before and after case.

In some scenarios, the injected capacity reduction was not high enough to cause congestion, thus the agents did not receive any alarm, and no alternative plan was proposed. Data for those scenarios, such as scenarios 1, 7, 10, and 14 in Table 7.1, 7.1, and 7.1 are reported for completeness, but they are not used for the computation of average measures of performance. Test scenario 13, for which Table 7.2 and 7.3 also reports no variation between the two forms of control, differs from the other three, since in this case, the agents

*no incident notification to CARTESIUS

were notified of the occurrence of congestion, but they did not find any alternative solution that would cause reduction of total travel time, and indeed the effect of congestion is so mild, that the average travel time increase per vehicle, with respect to the scenario with no incident is only 0.38%.

Table 7.3: Distributed system: simulated average and total traveled distance

#	Incident Characteristics			Average (miles)		Total (*10 ⁶ veh-miles)	
	Location	Lanes	Length (min)	Before	After	Before	After
0	no incident			4.48	4.48	0.197	0.197
1	405 NB	1(4)	20	4.51	4.51*	0.198	0.198
2		2(4)	20	4.52	4.55	0.199	0.200
3		2(4)	25	4.53	4.54	0.199	0.199
4	405 SB	1(4)	20	4.51	4.53	0.198	0.199
5		1(3)	20	4.49	4.52	0.198	0.199
6		2(4)	20	4.50	4.52	0.198	0.198
7	5 NB	2(5)	15	4.48	4.48*	0.197	0.197
8		3(5)	15	4.48	4.49	0.197	0.197
9		3(5)	20	4.48	4.48	0.197	0.197
10	5 SB	1(5)	15	4.49	4.49*	0.198	0.198
11		2(5)	20	4.50	4.50	0.195	0.198
12		3(5)	30	4.50	4.52	0.196	0.199
13	Alton EB	2(3)	20	4.48	4.48	0.197	0.197
14	Alton WB	2(3)	20	4.48	4.48*	0.198	0.198
15	on-ramp	1(2)	15	4.48	4.48	0.198	0.198
16	405 NB Alton WB	2(4) 1(2)	20 15	4.52	4.52	0.199	0.199
17	405 SB on-ramp	2(4) 1(2)	20 10	4.51	4.53	0.201	0.201

Table 7.2 and 7.3 show a summary of the network performance for the 18 test scenarios. For each scenario, average (per vehicle) and total travel time and traveled distance are reported. Each scenario includes data pertaining to the use of default control (Before), and to the use of control directives suggested by CARTESIUS (After).

Data reported in Table 7.2 and 7.3 show that the implementation of the control plans suggested by CARTESIUS determines, in general, a reduction in the average (and to-

*no incident notification to CARTESIUS

tal) network-wide travel time. The average (and total) traveled distance is not significantly affected by the alternative control, even though the control plans, in all test scenarios, included the use of CMS messages. This result should be perhaps attributed to the limited size of the network.

Travel Time Reduction

The quality of the improvement, in general, varies according to the availability of alternative paths, the amount of spare capacity on those paths, and the demand/capacity ratio on paths affected by congestion. Thus, the analysis of the percentage travel time reduction for the various test scenarios, is partitioned according to the location of the incident.

Tables 7.4-7.9 report on the difference in average travel time between the two forms of control, expressed as a percentage of the average travel time in the default case. In order to obtain a normalized measure of the travel time reduction across the scenarios, the tables also report, for each case, on the average travel time increase with respect to scenario 0, which corresponds to a *no incident* situation, and can thus be considered as the base case. For each case, the difference in per-vehicle travel time corresponding to the the two forms of control, is reported. Also, the mean percentage travel time reduction, among all the scenarios for which CARTESIUS was notified of the occurrence of congestion is reported.

Table 7.4 reports on the scenarios related to incidents on the Interstate 405 north-bound (405 NB)

Table 7.4: Average travel time for the 405 NB scenarios

#	Incident Characteristics		Average Travel Time			Comparison with Scenario 0		
	Lanes	Length (min)	Before	After	Δ (%)	Before	After	Δ (%)
0	no incident		5.24	5.24	0.00	0.00	0.00	0.00
1	1(4)	20	5.87	**	**	12.02	**	**
2	2(4)	20	6.76	6.32	-6.51	29.00	20.61	-8.39
3	2(4)	25	7.20	6.70	-6.94	37.40	27.46	-9.94
Mean					-6.73	-9.17		

Table 7.5 reports on the scenarios related to incidents on the Interstate 405 south-bound (405 SB)

Table 7.5: Average travel time for the 405 SB scenarios

#	Incident Characteristics		Average Travel Time			Comparison with Scenario 0		
	Lanes	Length (min)	Before	After	Δ (%)	Before	After	Δ (%)
0	no incident		5.24	5.24	0.00	0.00	0.00	0.00
4	1(4)	20	7.47	6.54	-12.45	42.56	24.81	-17.75
5	1(3)	20	6.86	5.85	-14.72	30.92	11.64	-19.28
6	2(4)	20	7.98	7.21	-9.65	52.29	37.60	-14.69
Mean			-12.27			-17.24		

Table 7.6 reports on the scenarios related to incidents on the Interstate 5 northbound (5 NB)

Table 7.6: Average travel time for the 5 NB scenarios

#	Incident Characteristics		Average Travel Time			Comparison with Scenario 0		
	Lanes	Length (min)	Before	After	Δ (%)	Before	After	Δ (%)
0	no incident		5.24	5.24	0.00	0.00	0.00	0.00
7	2(5)	15	5.30	**	**	1.15	**	**
8	3(5)	15	5.55	5.26	-5.23	5.92	0.38	-5.54
9	3(5)	20	5.81	5.33	-8.26	10.88	1.72	-9.16
Mean			-6.74			-7.35		

Table 7.7 reports on the scenarios related to incidents on the Interstate 5 southbound (5 SB)

Table 7.7: Average travel time for the 5 SB scenarios

#	Incident Characteristics		Average Travel Time			Comparison with Scenario 0		
	Lanes	Length (min)	Before	After	Δ (%)	Before	After	Δ (%)
0	no incident		5.24	5.24	0.00	0.00	0.00	0.00
10	1(5)	15	5.24	**	**	0.00	**	**
11	2(5)	20	7.05	5.97	-15.32	34.54	13.93	-20.61
12	3(5)	30	8.69	8.38	-3.57	65.83	59.92	-5.91
Mean			-9.45			-13.26		

Table 7.8 reports on the scenarios related to incidents on arterials and at a freeway on-ramp. As mentioned above, only scenarios 13 and 15 are used for the computation of the mean travel time reduction.

Table 7.8: Average travel time for Alton (WB and EB) and on-ramp scenarios

#	Incident Characteristics		Average Travel Time			Comparison with Scenario 0		
	Lanes	Length (min)	Before	After	Δ (%)	Before	After	Δ (%)
0	no incident		5.24	5.24	0.00	0.00	0.00	0.00
13	2(3)	20	5.26	5.26	0.00	0.38	0.38	0.00
14	2(3)	20	5.26	**	**	0.38	**	**
15	1(2)	15	5.26	5.24	-0.38	0.38	0.00	-0.38
Mean					-0.19	-0.19		

The set of scenarios includes cases where the control plan established in response to the first incident must be re-evaluated and altered at the occurrence of the second incident (scenarios 16 and 17). Table 7.9 reports on the scenarios related to those cases.

Table 7.9: Average travel time for multiple incidents scenarios

	Incident Characteristics		Average Travel Time		
#	Lanes	Length (min)	Before	After	Δ (%)
0	no incident		5.24	5.24	0.00
16	2(4)	20	6.78	6.35	-6.34
	1(2)	15			
17	2(4)	20	8.09	7.57	-6.43
	1(2)	10			
Mean					-6.39

Comparison with Scenario 0		
Before	After	Δ (%)
0.00	0.00	0.00
29.38	21.18	-8.20
54.39	44.47	-9.92
		-9.06

It is not a simple task to draw general and absolute conclusions about the validity of the approach, beyond the simple observation that in all tested scenarios the implementation of the control plans suggested by CARTESIUS determines an improvement of network-wide traffic conditions (except scenario 13, where no alternative solution is determined; it must be noted, however, that in this case, the occurrence of an incident determines a very low (0.38%) deterioration of average travel time).

The mean percentage reduction of average (per vehicle) travel time across all scenarios (including scenario 13) is 7.4%. Compared to the *no incident* base case (scenario 0), the mean percentage increase of average travel time, caused by the occurrence of incidents, is reduced from 31.2% to 20.3%.

The impact of incident and of incident response measures varies extensively, depending on a variety of factors, that include the capacity reduction caused by the incident, its duration, the location of the incident, the availability of feasible alternative routes, and the demand/capacity ratio before, during, and after the occurrence of the incident, both on the routes directly affected by the incidents and on the remaining sections of the network that are directly or indirectly affected by incident response strategies.

As noted, the scenarios can be classified according to the location of the incident. Looking at the results presented in Tables 7.4-7.9, it can be observed that on average, the highest performance improvement is experienced in the scenarios related to incident along the I-405 freeway, southbound, with a mean reduction in average travel time of 12.27%, and a mean difference in the travel time increase caused by the incident, of 17.24% (compared to scenario 0, see Table 7.5). The lowest improvements are observed for incidents occurring on surface streets. In those cases, as shown in Table 7.8, the effect of incidents is marginal, thus the improvement of traffic conditions is also very small.

Performance does not deteriorate for the cases of multiple incidents (Table 7.9), which seems to demonstrate the effectiveness of CARTESIUS in dealing with multiple, concurrent sources of congestion.

7.3.2 Response Time

Given the real-time nature of the problem that the system is intended to address, it was important to provide a measure of the system response time, i.e., the time required by the agents to determine a list of control plans, once they have been provided by the operator with all the necessary input. Thus, for each simulated scenario, the system's maximum response time was collected and reported. A comparative analysis of the response time with that of other systems could not be performed, since no comparable data, except that related to the centralized agent TCM, reported in Table 3.1 for a different network, is available in the literature. Nonetheless an absolute report on the response time provides a measure of the quality of the solution process, and can be used in the future as a benchmark

for similar applications.

The response time is partially dependent on the number of alternative solutions that must be composed and evaluated, which, in turn, depends on the availability of alternative paths. Since, as part of the DTPS algorithm, each agent must compose pairs of local solutions into compatible global solutions, and assess the corresponding delay reduction, the response time also depends on the number of compatible global solutions found by the agents.

As described in Chapter 5, the problem-solving process requires, at various times, the agents to synchronize and exchange data. Even if, compatibly with the FA/C approach, synchronization requirements are limited, the agents need to synchronize and exchange data before the final solutions are presented to the operator. The last data exchange, in the DTPS algorithm involves transmission of the delay reduction associated with each global solutions, computed by each agent. After such data exchange, each agent performs the same operations on the same data set, thus, the agent have the same response time, and it is appropriate to speak of a *system* response time, rather than *agent* response time.

Table 7.10 shows the system response time. In all tested scenarios, the system response time is always below 22 seconds, with an average of 15.3 seconds. These measures were obtained using a SUN Ultra 30 Workstation, with an Ultra Sparc 2 processor.

The system response time, in all tested scenarios, is distributed fairly homogeneously around a mean value of 15.3 seconds, with a minimum of 11 seconds for scenarios 6 and 15, and maximum values of 21 and 22 seconds, for scenarios 2 and 3. This difference can be explained by the difference in the number of available alternative solutions.

The vast majority of traffic that takes advantage of diversion strategies is freeway traffic. Northbound traffic on the I-405 freeway can be diverted using several alternative routes, depending on the location of the incident, that always include the arterial Alton Parkway, a divided arterial, with three lanes per direction and two protected left-turn lanes at most intersections. Traffic can potentially be diverted through the I-5 freeway (leaving the freeway at the Alton off-ramp), or through the arterial Irvine Center Drive, that has relatively low cross traffic until the intersection with Alton Parkway (see Figure 7.1). Furthermore, the available CMS messages on Alton WB can be used to direct traffic back to the I-405 using either Irvine Center Drive, Sand Canyon Avenue, or Jeffrey Road. When looking for solutions, the agents must evaluate all feasible combinations of these alternatives.

Because of the location of CMS, southbound traffic on the I-405 and both north

Table 7.10: Distributed system: maximum response time

#	Incident Characteristics			Response Time (sec)
	Location	Lanes	Length (min)	
1	405 NB	1(4)	20	**
2		2(4)	20	22
3		2(4)	25	21
4	405 SB	1(4)	20	15
5		1(3)	20	15
6		2(4)	20	11
7	5 NB	2(5)	15	**
8		3(5)	15	15
9		3(5)	20	15
10	5 SB	1(5)	15	**
11		2(5)	20	16
12		3(5)	15	15
13	Alton EB	2(3)	20	15
14	Alton WB	2(3)	20	**
15	on-ramp	1(2)	15	11
16	405 NB	2(4)	20	15
	Alton WB	1(2)	15	
17	405 SB	2(4)	20	13
	on-ramp	1(2)	10	

and southbound traffic on the I-5 have slightly fewer diversion options. Thus in general, the agents must analyze fewer possible alternatives, and the system response time for these cases is lower. The location of the CMS on the I-405 southbound, downstream of Jeffrey Road, reduces the number of freeway off-ramps that can be effectively used for diversion to two, thus limiting the number of potential alternative routing solutions. An analogous situation occurs for incidents occurring on the I-5 freeway, where, because of the limited number of CMS, southbound traffic can be diverted by CARTESIUS only through the SR-133 freeway, to the I-405, and northbound traffic, depending on the incident location can only use either the I-405 or the off-ramp at Alton Parkway.

It must be observed that the system response time does not vary significantly for scenarios that involve the occurrence of multiple incidents (scenarios 16 and 17), compared to the cases of a single incident. This can probably be explained by considering that, in general, a higher number of problems on one hand increases the number of possible solutions

that must be analyzed, but on the other, reduces the number of possible alternatives, thus reducing the number of feasible solutions for which the corresponding delay must be computed. Even if scenarios with more than two concurrent incidents have not been simulated, this characteristic leads to believe that the system response time is not likely to vary significantly with the spatial density of incidents.

Finally, an approximative comparison can be drawn with data on the response time of the centralized system TCM, reported in Table 3.1. The Irvine network controlled by CARTESIUS is larger than the Anaheim subnetwork controlled by TCM, and it has a higher number of signalized intersections and metered freeway ramps. Nonetheless, even considering synchronization delay, CARTESIUS is significantly faster than the centralized TCM. Considering the two samples of scenarios through which TCM and CARTESIUS were evaluated (as reported respectively in Tables 3.1 and 7.10) the average response time for TCM was 53.2 seconds (with a maximum of 91.5 seconds), while that of CARTESIUS was 15.3 seconds (with a maximum of 22 seconds).

7.3.3 Effect of Signal Plans

As part of the analysis of the system's performance it was judged interesting to quantitatively assess the effect of signalization, within the integrated control plans proposed by CARTESIUS. For three of the scenarios for which diversion through the arterial system is recommended (scenarios 2, 5, and 9, with incidents respectively on the I-405 northbound, I-405 southbound, and I-5 southbound), an additional simulation was performed, in which the plans for signals and ramp meters suggested by CARTESIUS were not transmitted to the traffic simulator. Table 7.11 describes data for test scenario 2, with an incident on the Interstate 405 freeway, northbound.

Table 7.11: Travel time with and without signal control for an incident on 405 NB

#	Signal Control	Incident Characteristics		Average Travel Time (min)			Comparison with Scenario 0	
		Lanes	Length (min)	Before	After	Δ (%)	Before	After
0		no incident		5.24	5.24	0.00	0.00	0.00
2	Yes	2(4)	20	6.76	6.32	-6.51	29.00	20.61
2'	No	2(4)	20	6.76	6.87	1.63	29.00	31.11
Difference						-8.14		-10.05

The network performance, for the modified scenarios 2, 5 and 9 was compared with that of the corresponding basic scenarios, in which the complete control directives were transmitted to the simulator. This test was aimed at estimating the synergetic effect of integrated control plans, by estimating the reduction in the network performance caused by the lack of integration between traffic diversion control and signal and meter control.

Table 7.12 describes the comparison for test scenario 5, with an incident on the Interstate 405 freeway, southbound.

Table 7.12: Travel time with and without signal control for an incident on 405 SB

#	Signal Control	Incident Characteristics		Average Travel Time (min)			Comparison with Scenario 0	
		Lanes	Length (min)	Before	After	Δ (%)	Before	After
0		no incident		5.24	5.24	0.00	0.00	0.00
5	Yes	1(3)	20	6.86	5.85	-14.72	30.92	11.64
5'	No	1(3)	20	6.86	6.46	-5.83	30.92	23.28
Difference						-8.89		-11.64

Table 7.13 describes the comparison for test scenario 9, with an incident on the Interstate 5 freeway, northbound.

Table 7.13: Travel time with and without signal control for an incident on 5 NB

#	Signal Control	Incident Characteristics		Average Travel Time (min)			Comparison with Scenario 0	
		Lanes	Length (min)	Before	After	Δ (%)	Before	After
0		no incident		5.24	5.24	0.00	0.00	0.00
9	Yes	3(5)	20	5.81	5.33	-8.26	10.88	1.72
9'	No	3(5)	20	5.81	5.35	-7.92	10.88	2.10
Difference						-0.34		-0.38

The two scenarios that involve incidents on the I-405 freeway are characterized by a clear superiority of the integrated control compared to the partial one. The scenario that involves diversion from the I-5 freeway does not show such a noticeable performance gain. This difference can be perhaps explained considering the characteristics of the major alternative routes for the freeway traffic in the three cases. In the first case (scenario 2),

freeway traffic is diverted through Irvine Center Drive, Alton Parkway, and Sand Canyon Avenue, and it crosses nine signalized intersections and one metered freeway on-ramp. In the second scenario (scenario 5), traffic is directed through Sand Canyon Avenue, Alton Parkway, to the I-5 freeway, crossing six intersections and one metered ramp. Also, in both cases, as a consequence of congestion, arterial traffic directed to the freeway (coming from Alton Parkway) is advised to remain on the surface street, modifying the original traffic distribution even further. In the last case (scenario 9), the alternative route crosses only five signalized intersections, and traffic directed from Alton to the freeway is not affected by the diversion because no CMS is available on Alton Parkway westbound, upstream of the freeway on-ramp.

Chapter 8

Conclusions and Directions for Future Research

8.1 Conclusions

This dissertation describes a new distributed real-time system that provides automatic decision support to TMC operators for multi-jurisdictional management of traffic incidents on integrated freeway and arterial networks. The uniqueness of this approach lies in the development of cooperation mechanisms to support a truly multi-decision maker distributed problem solving process that takes into account requirements and real-time constraints from all cooperating parties and develops control solutions that satisfy them.

The necessity for seamless transportation management systems resulting from an effective cooperation effort among all agencies involved, has been widely advocated. Yet, to date, very few experiments have proven successful in providing efficient automatic operator decision support. Most attempts are directed towards the development of data sharing and communication networks, to provide all involved parties with shared data access to global information. Such approaches, spun from the recent advancements in telecommunication technology and distributed networks, such as the World Wide Web, if they do not cause operator cognitive overload facilitate better-informed decision-making, but they are not intended to explicitly facilitate cooperation, coordination and conflict resolution. With respect to multi-jurisdictional traffic incident management, some research and development projects, such as the Santa Monica Smart Corridor System, described in Chapter 2, stand out for their ability to provide data fusion, information storage, resource allocation, and semi-automated decision support. Yet, even such an advanced system like the Smart Corridor does not perform truly interactive multi-decision maker problem-solving. Expert system reasoning capabilities are used by the Smart Corridor system to assign, based on the

incident location, incident management responsibilities to one of the cooperating agencies, which takes control of the management process and selects suitable predefined plans and submits them to the other agencies for approval. In order for the incident management system to determine integrated response plans that satisfy real-time requirements from other agencies, it is necessary that the system have access to all relevant corridor-wide data, thus requiring the existence of a complex shared-access corridor database used by the system.

The approach used by CARTESIUS employs coordination mechanisms that support cooperation and conflict resolution among two distinct automatic problem solvers. The agents have access to separate databases and may use different control algorithms, thus reflecting the inherent administrative distribution of data and expertise among separate management agencies. The cornerstone of this cooperative approach is the assumption that satisficing control solutions can be efficiently obtained by relaxing the requirement that agents have shared access to *all* globally available information. In fact, synchronization delays and data communication requirements can be reduced if agents are able to perform partial processing of local data and to exchange only the processed results.

The contribution of this research lies in the demonstration of the validity of such assumptions and the application of theoretical principles of the FA/C distributed problem-solving paradigm to traffic control, through the development of a distributed software architecture. The simulation-based validation of the system performance permits assessment the effectiveness of this approach in producing real-time, integrated traffic control solutions that reduce the adverse impact of incidents on traffic circulation, network-wide.

The effectiveness of this approach for the problem of distributed traffic control seems to indicate the practical worth of an exploration of its applicability to other transportation problems with similar characteristics. These characteristics include the presence of stringent real-time constraints, the possibility of abstracting relevant properties of the problem space according to macro or mesoscopic principles, and the existence of physical or logical interaction effects between remote problems and solutions. Under such circumstances, the main features of the communication protocol among the agents, such as the specification of conditions and their translation into conditional subgoals that update the goal structure of the agents, and the exchange of highly relevant, goal-related information among them, may find application in other distributed domains. One such domain, especially under real-time conditions, could be distributed dynamic traffic assignment, where conditions may specify the viability of a set of paths, and goal-related information may in-

clude the current travel time on a subpath which is under the jurisdiction of a remote agent. Another potentially applicable domain for which no satisfactory solution exists, could be the distributed real-time estimation of driver reaction to traffic information and the construction of compliance rate databases, using appropriate techniques for vehicle detection and summarizing locally observed data into macroscopic measures, making the latter available to remote agents for further processing.

The software system developed as part of the TRICEPS project can also serve as a backbone platform for testing and validation of further traffic control and traveler information algorithms. If interfaced within the DTPS algorithm, alternative strategies for signal, ramp and CMS control can then be easily implemented and evaluated.

8.2 Future Research

The analysis of the problem of real-time, network-wide traffic management and the design, development and evaluation of CARTESIUS have indicated several directions for future research, highlighting design aspects that need to be refined, identifying issues that can enhance the system's effectiveness, and proposing the exploration of the applicability of the approach to other domains.

8.2.1 Refinements to Problem Solving

The quality of the problem-solving process can be enhanced, taking advantage of the modularity of the architecture, by applying more sophisticated and efficient algorithms for signal and ramp meter control. As pointed out in the previous chapters, the intent of this research was not to devise new traffic control algorithms but rather to efficiently integrate existing techniques into a modular architecture for incident management, with emphasis on coordinating the efforts of the agencies responsible for operations and control of urban freeway and arterial networks.

Refinements to the agents' problem-solving capabilities include a more sophisticated treatment of the uncertainty inherent in many aspects of the real-time traffic control domain, for instance adopting techniques such as the one described in Section 6.5.

Another important aspect that deserves attention is the estimation of the traffic demand. A correct estimation of the current and near future demand is the key to effective

control measures. At present, CARTESIUS adjusts a *backbone* demand distribution, resulting from a network assignment, using real-time measurements of traffic data. As pointed out in Chapter 6, such a demand computation module provides a fast but rather coarse assessment of the network demand. A more precise estimation of the traffic demand, based on real-time dynamic traffic assignment techniques is likely to improve the system's control effectiveness.

CARTESIUS' simulation-based validation has confirmed the importance of correctly monitoring the dynamics of congestion and the effect of the implemented control. In the present design, the Current Problem Monitor module notifies the system when sudden and unexpected events occur that are related to or likely to affect the current control, such as the onset of congestion on alternative paths, the early clearance of a queue, or the termination of the prespecified recovery time interval. While such implementation is effective in quickly determining a coarse measure of the suitability of the current control, a more refined control can be obtained using a more sophisticated module. This would not just react to the presence or absence of congestion, but rather analyze current traffic flows during the recovery time interval, comparing observed data with expected conditions, and performing slight adjustments to the current control. Such adjustments may include the update of delay information on CMS messages, or the slight modification of signal and ramp meter control parameters.

8.2.2 Extensions

The interaction of public transport with private traffic, both on freeway and arterial networks suggests an analysis of the applicability of the cooperative approach to the problem of integrated traffic control and transit operations.

Recent studies (Schweiger 1995, Hickman & Day 1996) have demonstrated the inadequacy of the current level of interaction between transit and traffic management centers, and have identified some of the potential advantages of a more effective cooperation. Transit agencies can dynamically adjust their operations based on traffic information received from traffic management centers, while traffic operators can be notified in real-time of incidents and traffic problems encountered by transit vehicles. Furthermore, travelers (both car drivers and bus riders) can receive up-to-date and more reliable information that may have an impact, both in the short and in the long run, on mode and route choice, and

consequently on traffic volumes.

An interesting extension of the multi-agent concept, that can take advantage of the modularity of the CARTESIUS architecture, involves the development of a third agent, responsible for the interface with a real-time management and operations system of a transit agency, in response to incidents. The control decision-making power of the transit agent is likely to be subordinated to that of the two traffic control agents. Nonetheless, requests could be issued for the implementation of bus-priority oriented control strategies, such as signal preemption. The most significant advantage to all parties, though, would lie in the exchange of updated information on current traffic conditions, for the dynamic adjustment of traffic control and transit operations.

A significant enhancement of CARTESIUS' reasoning power involves the development of *learning* capabilities, or more humbly, of semi-automatic knowledge acquisition, i.e. the ability to analyze past events, determine the effectiveness of the implemented control approach, and apply the newly acquired knowledge to the solution of future problems. Automatic learning is a very complex task, both at theoretical and practical levels, that may require abstracting all relevant information, recognizing the similarity between past and current events, and adapting current operations according to the difference between previous experiences and current observations. Yet, the development of a learning module within CARTESIUS could be performed as a step-wise process, in which sophistication is increased incrementally. A first step could involve the definition of a semi-automated module, in which interaction with a human operator facilitates the system's learning process by, for instance, requesting the operator to provide an assessment of a previously implemented solution.

8.2.3 Application of Cooperative Approach to Other Domains

Perhaps one of the most challenging tasks involves the exploration of the applicability of the cooperation protocol designed for CARTESIUS to other problem domains. Necessary steps of this effort involve the identification of suitable characteristics of such domains, and the tailoring of some of the developed concepts to such characteristics.

Some of the basic requirements, such as a certain degree of near-monotonicity or the presence of real-time constraints have been identified, both in this dissertation and by others. Some transportation-related problem domains that are likely to benefit from a

FA/C-based approach have been introduced in the previous section, including distributed dynamic traffic assignment. The application of the developed concepts to other domains may require the exploration of alternative cooperation strategies, or the refinement of current ones, in such ways as those introduced in Chapter 5.

BIBLIOGRAPHY

- Abdulhai, B. (1997). Development of a universally transferable freeway incident detection framework. Paper no. 971121 prepared for presentation at the 76th annual meeting of the Transportation Research Board, Washington, DC.
- Akcelik, R. (1981). Traffic signals: capacity and timing analysis, *Research Report ARR123*. Australian Road Research Board, Victoria, Australia.
- Akcelik, R. (1994). Estimation of green times and cycle times for vehicle-actuated signals. *Transportation Research Records* 1457: 63–72.
- Ambrosino, G., Boero, M. & Turrini, M. (1995). Integrating urban traffic management services: the THERMIE-JUPITER approach in Florence, *Congress on Applications of Transport Telematics and Intelligent Vehicle-Highway Systems. Towards an intelligent transport system*, Vol. II, Artech House, pp. 446–453.
- Anderson, B. D. O. & Moore, J. B. (1990). *Optimal control - Linear quadratic methods*. Prentice Hall, Englewood Cliffs, NJ.
- Andrews, C. M., Elahi, S. M. & Clark, J. E. (1997). Evaluation of New Jersey route 18 OPAC/MIST traffic control system, *Transportation Research Record* 1603: 150–155.
- Bender, N. A. (1996). *Mathematical methods in artificial intelligence*. IEEE Computer Society Press, Los Alamitos, CA, chapter 6.
- Bielli, M., Ambrosino, G. & Boero, M. (1994). *Artificial intelligence applications to traffic engineering*, VSP, Utrecht, The Netherlands.
- Birrel, A. D. & Nelson, B. J. (1984). Implementing remote procedure calls, *ACM Transactions on Computer Systems* 2: 39–59.

- Black, U. D. (1992). *TCP/IP and related protocols*, McGraw-Hill, New York, NY.
- Bond, A. H. & Gasser, L. (eds) (1988). *Readings in Distributed Artificial Intelligence*. Morgan Kauffman, San Mateo, CA.
- Brocken, M. G. & van der Vlist, M. J. M. (1991). Traffic control with variable message signs, in W. M. Spreitzer (ed.), *Vehicle Navigation and Information Systems Conference Proceedings*, Vol. 1, Society of Automotive Engineers, Warrendale, PA, pp. 27-46.
- Buchanan, B. G. & Shortliffe, E. H. (1984). *Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, Reading, MA.
- Carver, N. (1990). *Sophisticated Control for Interpretation: Planning to resolve uncertainty*, Ph.D Dissertation in Computer and Information Science, University of Massachusetts.
- Carver, N., Cvetanovic, Z. & Lesser, V. (1991). Sophisticated cooperation in FA/C distributed problem-solving systems, *Proceedings of the National Conference on Artificial Intelligence*, pp. 191-198.
- Carver, N. & Lesser, V. (1994). A first step toward the formal analysis of solution quality in FA/C distributed interpretation systems, *Technical Report 94-37*, Department of Computer Science, University of Massachusetts.
- Carver, N. & Lesser, V. (1996). A formal analysis of solution quality in FA/C distributed sensor interpretation, in V. Lesser (ed.), *Proceedings of the 2nd International Conference on Multi-Agent Systems*, AAAI Press, Menlo Park, Ca, pp. 11-17.
- Carver, N., Lesser, V. & Whitehair, R. (1996). Nearly monotonic problems: a key to effective FA/C distributed sensor interpretation?, *Proceedings of the 13th National Conference on Artificial Intelligence*, Vol. I, MIT Press, Cambridge, MA, pp. 88-95.
- Chandrasekaran, B. (1983a). On evaluating AI systems for medical diagnosis, *AI Magazine* 4: 34-37.
- Chandrasekaran, B. (1983b). Towards a taxonomy of problem solving types, *AI Magazine* 4: 9-17.

- Chang, E. C., Ho, K. K., Liu, K. W. & Hsu, C. J. (1995). Development of an integrated freeway-management system, *Proceedings of the Annual Meeting of the ITS America*. Vol. I, ITS America, pp. 589–596.
- Charniak, E. & McDermott, D. (1985). *Introduction to artificial intelligence*. Addison-Wesley, Reading, MA.
- Clark, C. E. (1961). The greatest of a finite set of random variables, *Operations Research* 9(2): 145–162.
- Clelland, A. (1993). The Los Angeles Smart Corridor Project. *Wescon/93 Conference Record*. IEEE. New York. NY.
- Cohn, L. F. & Harris, R. A. (1992). Knowledge-based expert systems in transportation. *NCHRP Synthesis 183*, Transportation Research Board, National Research Council. Washington. DC.
- Commission of the European Communities (1991). Advanced telematics in road transport. *Proceedings of the DRIVE Conference*. Elsevier.
- Conry, S. E., Kuwabara, K., Lesser, V. & Meyer, R. A. (1991). Multistage negotiation for distributed constraint satisfaction, *IEEE Transactions on Systems, Man, and Cybernetics* SMC-21(6): 1462–1477.
- Csallner, A. & Schlichter, H. G. (1995). Munich COMFORT: COoperative Management FOR urban and Regional Transport. *Traffic technology international* pp. 48–50.
- Cuena, J., Hernandez, J. & Molina, M. (1995). Knowledge-based models for adaptive traffic management systems, *Transportation Research* 3C(5): 311–337.
- D'Ans, G. C. & Gazis, D. C. (1976). Optimal control of oversaturated store-and-forward transportation networks, *Transportation Science* 10(1): 1–19.
- Davis, R. (1980). Report on the workshop on distributed artificial intelligence. *SIGART Newsletter* 73: 42–52.
- Davis, R. (1982). Report on the second annual workshop on distributed artificial intelligence, *SIGART Newsletter* 80: 13–23.

- Decker, K. S. (1987). Distributed problem-solving techniques: a survey, *IEEE Transactions on Systems, Man and Cybernetics* SMC-17: 729-740.
- Derr, D. R. (1987). Traffic management teams, management of freeway emergency and special events, *Transportation Research Board Circular* 326.
- Diakaki, C., Papageorgiou, M. & McLean, T. (1997). Simulated studies of integrated corridor control in Glasgow, *Transportation Research* 5C(3/4): 211-224.
- Diakaki, C., Papageorgiou, M. & McLean, T. (1998). Applying integrated corridor control in Glasgow, in C. Hendrickson & S. G. Ritchie (eds), *Proceedings of the Fifth International Conference on Applications of Advanced Technologies in Transportation*, American Society of Civil Engineers, Newport Beach, CA, pp. 9-16.
- Doran, J. E. & Michie, D. (1966). Experiments with the Graph Traverser program, *Proceedings of the Royal Society of London* 294A: 235-259.
- Dudek, C. L., Weaver, G. D., Hatcher, D. R. & Richards, S. H. (1978). Field evaluation of messages for real-time diversion of freeway traffic for special events, *Transportation Research Record* 682: 37-45.
- Durand-Rauchet, Y. (1995). Sirius and Carminat: Cornerstones of a well-balanced policy between traffic management and information in the Paris region, *Proceedings of the Annual Meeting of the ITS America*, Vol. I, ITS America, pp. 575-578.
- Engelbrecht, R. J., Fambro, D. B., Rouphail, N. M. & Barkawi, A. A. (1997). Validation of a generalized delay model for oversaturated conditions, *Proceedings of the 76th Annual Meeting of the Transportation Research Board*, Washington, DC.
- Ernst, G. W. & Newell, A. (1969). *GPS: a case study in generality and problem solving*, Academic Press, New York, NY.
- Euler, G. W. (1990). Intelligent Vehicle/Highway Systems: definitions and applications, *ITE Journal* 60(11): 17-22.
- Euler, G. W. & Lindley, J. A. (1989). Integrated traffic management concepts and considerations, *ITE Compendium of Technical Papers*, Vol. 59, Institute of Transportation Engineers, Washington, DC, pp. 255-260.

- Fambro, D. B. & Rouphail, N. M. (1997). Generalized delay model for signalized intersections and arterial streets, *Proceedings of the 76th Annual Meeting of the Transportation Research Board*, Washington, DC.
- Farradyne Systems Inc. (1996). Integrated ramp meter/adaptive signal control. City of Irvine Field Operational Test, Final OPAC Software Test Plan Document.
- Fehling, M. & Erman, L. (1983). Report on the third annual workshop on distributed artificial intelligence, *SIGART Newsletter* 84: 3-12.
- FHWA (1987). *Enhanced UTCS software data base specification*. Federal Highway Administration, McLean, VA.
- Fikes, R. E. & Nilsson, N. J. (1971). STRIPS: a new approach to the application of theorem proving to problem solving, *Artificial Intelligence* 2(3-4): 189-208.
- Gartner, N. H. (1983). OPAC: a demand-responsive strategy for traffic signal control. *Transportation Research Record* 906: 75-81.
- Gartner, N. H. (1991). Evaluation of optimized policies for adaptive control strategy, *Transportation Research Record* 1324: 105-114.
- Gazis, D. C. (1974). *Traffic control - theory and application*. John Wiley and Sons, New York, chapter 3.
- Gensym (1995). *G2: Reference Manual. Version 4.0*, Gensym Corporation, Cambridge, MA.
- Grasso, B., Ward, M., Hall, G., Perez, C. E. & Eiger, A. (1995). ATMS and wide-area traffic management, *Proceedings of the Annual Meeting of the ITS America*. Vol. I. ITS America, pp. 543-553.
- Gupta, U. (1990). *Validating and verifying knowledge-based systems*, IEEE Computer Society Press, Los Alamitos, CA.
- Hall, D. L. & Heinze, D. T. (1989). The use of simulation techniques for expert system test and evaluation, *ISA Transactions* 28(1): 19-22.

- Hangleiter, S. (1995). Urban traffic control: AILA - Automatic Incident Detection Algorithm, *Proceedings of the LLAMD European Conference on Advanced Transport Telematics*.
- HCM (1985). Highway Capacity Manual, Transportation Research Board, Washington, DC.
- HCM (1994). Highway Capacity Manual, Transportation Research Board, Washington, DC.
- Hickman, M. & Day, T. (1996). Assessment of information systems and technologies at California transit agencies, *Transportation Research Record* 1521: 49–57.
- Hoops, M. & Tsavachidis, M. (1997). An integrated approach to collective route guidance, in: M. Papageorgiou (ed.), *8th IFAC Symposium on Transportation Systems*. Vol. III, International Federation of Automatic Control, pp. 1175–1180.
- Hurdle, V. F. (1984). Signalized intrersection delay models: A primer for the uninitiated. *Transportation Research Record* 971: 96–104.
- Jämsä, H. (ed.) (1990). *OECD workshop on knowledge-based expert systems in transportation*. Vol. I, Organisation for Economic Co-operation and Development, Espoo, Finland.
- Jayakrishnan, R. (1992). *In-vehicle information systems for network traffic control*. Ph.D Dissertation in Civil Engineering, University of Texas at Austin.
- Jayakrishnan, R., Mahmassani, H. S. & Hu, T. (1994). An evaluation tool for advanced traffic information and management systems in urban networks, *Transportation Research* 2C(3): 129–148.
- JHK & Associates (1993). *Irvine Transportation Analysis Model*. City of Irvine, Irvine, CA.
- Karimi, A., Gupta, A., Wheeler, G. & Massarano, G. (1992). Application of knowledge-based expert systems to incident management in the Santa Monica Smart Corridor, in S. G. Ritchie & C. Hedrickson (eds), *Proceedings of the International Conference on Artificial Intelligence Applications in Transportation Engineering*, Engineering Foundation, New York, NY, pp. 59–76.

- Khan, S. I. (1995). *Modular neural network architecture for detection of operational problems on urban arterials*, Ph.D Dissertation in Civil Engineering, University of California, Irvine.
- Kimber, R. M. & Hollis, E. M. (1979). Traffic queues and delays at road junctions, *TRRL Report 909*. Transport and Road Research Laboratory, Crowthorne, England.
- Kirby, H. R. & Parker, B. G. (1994). Artificial intelligence in transport: motivations, current state and perspectives, in M. Bielli, G. Ambrosino & M. Boero (eds), *Artificial intelligence applications to traffic engineering*, VSP, Utrecht, The Netherlands, pp. 3-28.
- Koble, H. M., Adams, T. A. & Samant, V. S. (1980). Control strategies in response to freeway incidents, *Technical Report FHWA/RD-80/005*, Federal Highway Administration, Washington, DC.
- Kulkarni, A. (1997). Testbed2 demand summary report. Unpublished internal report. Institute of Transportation Studies, University of California, Irvine.
- Lee, L. K. & McCourt, R. (1995). Institutional issues and jurisdictional cooperation model: the Portland ITS early deployment plan, *Proceedings of the Annual Meeting of the ITS America*, Vol. I, ITS America, pp. 471-480.
- Lee, S. & O'Keefe, R. M. (1994). Developing a strategy for expert system verification and validation, *IEEE Transactions on Systems, Man and Cybernetics* SMC-24(4): 643-655.
- Leonard, D. R., Gower, P. & Taylor, N. B. (1989). CONTRAM, structure of the model, *TRRL research report*, Transport and Road Research Laboratory, Crowthorne, England.
- Lesser, V. & Corkill, D. D. (1987). Distributed problem solving, in S. C. Shapiro (ed.), *Encyclopedia of Artificial Intelligence*, Vol. I, John Wiley and Sons, New York, NY, pp. 245-250.
- Lesser, V. & Ertman, L. D. (1980). Distributed interpretation: a model and an experiment, *IEEE Transactions on Computers* C-29(12): 1144-1163. Special issue on distributed processing.

- Lesser, V., Fennel, R. D., Erman, L. D. & Reddy, D. R. (1975). Organization of the Hearsay-II speech understanding system, *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-23*: 11-23.
- Lesser, V. R. (1991). A retrospective view of FA/C distributed problem solving, *IEEE Transactions on Systems, Man, and Cybernetics SMC-21*(6): 1347-1362.
- Lesser, V. R. & Corkill, D. D. (1981). Functionally Accurate, Cooperative distributed systems, *IEEE Transactions on Systems, Man, and Cybernetics SMC-11*(1): 81-96.
- Lesser, V. R. & Corkill, D. D. (1983). The Distributed Vehicle Monitoring Testbed: a tool for investigating distributed problem solving networks, *AI Magazine* 4(3): 15-33.
- Logi, F. (1995). A software architecture for the integration of Advanced Transportation Management and Information Systems, Master of Science in Civil Engineering, University of California, Irvine.
- Logi, F. & Ritchie, S. G. (1997). Verification, validation, and evaluation of TCM, a knowledge-based system for traffic congestion management, *Technical Report UCI-ITS-RR-97-1*, Institute of Transportation Studies, Irvine, CA.
- Longley, D. A. (1968). A control strategy for congested computer-controlled traffic networks. *Transportation Research* 2(4): 391-408.
- Mattingly, S. P. & McNally, M. G. (1998). Institutional assessment of the Irvine integrated corridor FOT, *Draft research report*, Institute of Transportation Studies, University of California, Irvine, Irvine, CA.
- McShane, W. R. & Roess, R. P. (1990). *Traffic Engineering*, 1 edn, Prentice Hall, Englewood, NJ.
- Messner, A. & Papageorgiou, M. (1990). METANET: a macroscopic simulation program for motorway networks, *Traffic Engineering and Control* 31(8/9): 466-470.
- Minsky, M. (1974). A framework for representing knowledge, *Artificial Intelligence Memo 306*, MIT AI Lab, Massachusetts Institute of Technology.

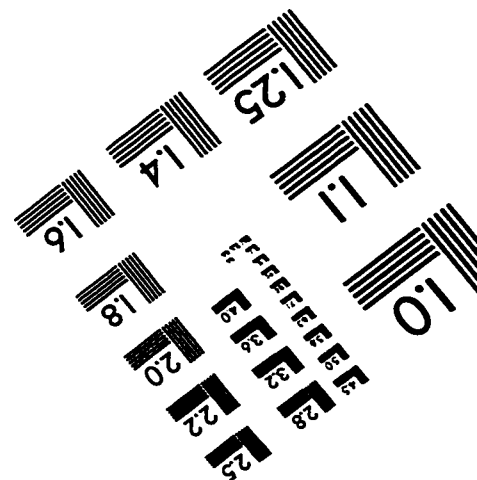
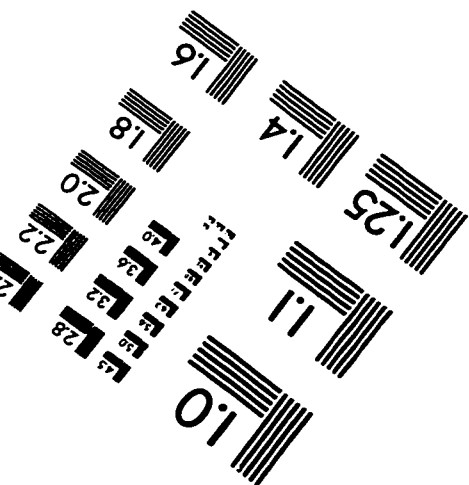
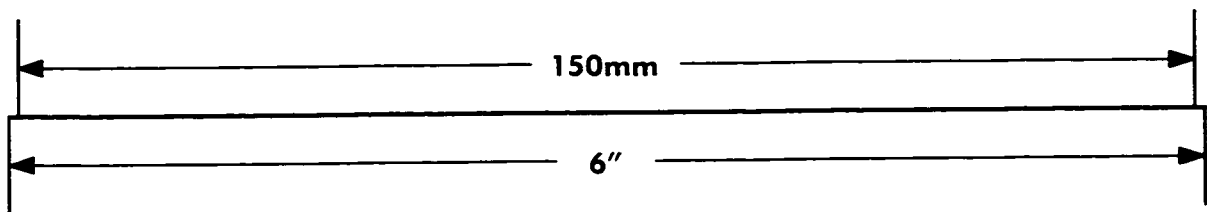
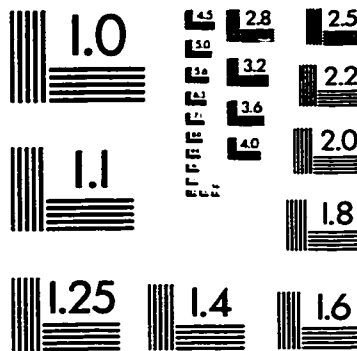
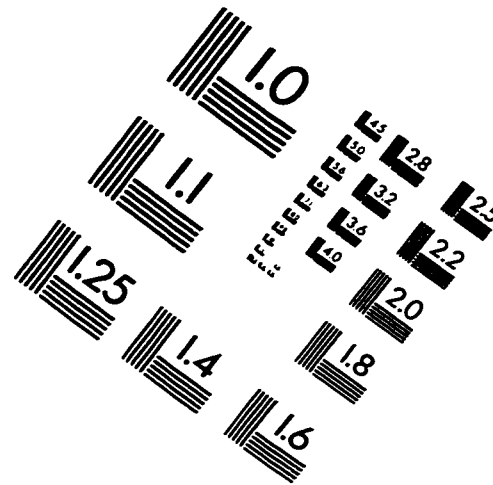
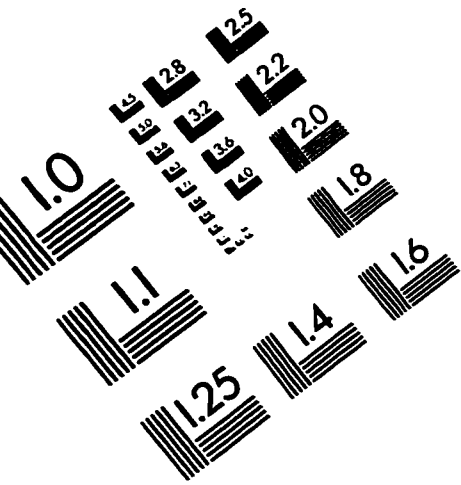
- Molina, M., Logi, F., Ritchie, S. G. & Cuenca, J. (1995). An architecture integrating symbolic and connectionist models for traffic management center decision support, *Proceedings of The 4th International Conference on Applications of Advanced Technologies in Transportation Engineering*, ASCE, New York, NY.
- National Engineering Technology Corporation (1996). System wide adaptive ramp metering high level design, *Draft report Contract 65-005*, FHWA FOT - City of Irvine/Caltrans D12 - National Engineering Technology Corporation.
- NET (1997). *Intertie system manual: intertie between Caltrans D12 ATMS and UCI*. National Engineering Technology Corporation.
- Newell, A. & Simon, H. A. (1963). GPS, a program that simulates human thought, in E. A. Feigenbaum & J. Feldman (eds), *Computers and Thought*. McGraw-Hill, New York, NY.
- North Seattle ATMS (1997). Washington State DOT Web Page <http://www.wsdot.wa.gov/regions/northwest/nsatms/atmsmain.htm>.
- NTCIP Steering Group (1995). The National Transportation Communications/ITS Protocol, *ITE Journal* 65(12): 46-40.
- O'Keefe, R. M., Balci, O. & Smith, E. P. (1987). Validating expert system performance. *IEEE Expert* 2(4): 81-89.
- O'Keefe, R. M. & O'Leary, D. E. (1993). Expert system verification and validation: a survey and tutorial, *Artificial Intelligence Review* 7(1): 3-42.
- OMG (1995). *The Common Object Request Broker: architecture and specification*, 2 edn.
- Papageorgiou, M. (1995). An integrated control approach for traffic corridors. *Transportation Research* 3C(1): 19-30.
- Papageorgiou, M., Haj-Salem, H. & Blosseville, J. M. (1991). ALINEA: a local feedback control law for on-ramp metering, *Transportation Research Record* 1320: 58-64.
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann, San Mateo, CA.

- Pignataro, L. J., McShane, W. R., Crowley, K. W., Lee, B. & Casey, T. W. (1978). Traffic control in oversaturated street networks, *NCHRP report 194*, Transportation Research Board, Washington, DC.
- Planning Consultants Research (1996). Addendum to the Disneyland resort final EIR no. 311, City of Anaheim, CA.
- Pooran, F. J. & Lieu, H. C. (1994a). Evaluation of system operating strategies for ramp metering and traffic signal coordination, *Proceedings of the annual meeting of the IVHS America*, Vol. II, IVHS America, Washington, DC, pp. 528-534.
- Pooran, F. J. & Lieu, H. C. (1994b). Implementation of operating strategies for integrated freeway/arterial control systems, *ITE Compendium of Technical Papers*, Institute of Transportation Engineers, Washington, DC.
- Prerau, D. S., Papp, W. L., Bhatnagar, R. & Weintraub, M. (1993). Verification and validation of expert systems: experience with four diverse systems, *International Journal of Expert Systems* 6(2): 251-269.
- Quinn, D. J. (1992). A review of queue management strategies, *Traffic Engineering and Control* 33(11): 600-605.
- Rich, E. & Knight, K. (1991). *Artificial Intelligence*, 2nd edn. McGraw Hill. New York. NY.
- Ritchie, S. G. (1990). A knowledge-based decision support architecture for advanced traffic management. *Transportation Research* 24A(1): 27-37.
- Robertson, D. I. (1979). Traffic models and optimum strategies of control - a review, in W. S. Homburger & L. Steinman (eds), *Proceedings of International Symposium on Traffic Control Systems*, Vol. 1, Berkeley, CA, pp. 262-288.
- Roseman, D. (1994). Incident management control concept: multi-agency coordinated traffic management: Santa Monica Freeway Smart Corridor, *ITE Compendium of Technical Papers*, Vol. 64, Institute of Transportation Engineers, Washington, DC, pp. 528-532.
- Roughail, N. M., Anwar, M., Fambro, D. B., Sloup, P. & Perez, C. E. (1997). Validation of a generalized delay model for vehicle-actuated traffic signals, *Proceedings of the 76th Annual Meeting of the Transportation Research Board*, Washington, DC.

- Rowe, E. (1989). Integrated traffic control: the Los Angeles Smart Corridor, *Proceedings of the 75th AASHTO Annual Meeting*, American Association of State Highway and Transportation Officials, pp. 159–167.
- Sachse, T. (1995). VARIA - variable message sign control based on OD estimation in a motorway network, *Proceedings of the LLAMD European Conference on Advanced Transport Telematics*.
- Schweiger, C. L. (1995). Integrated Transportation Management Centers, *Proceedings of the Annual Meeting of the ITS America*, Vol. I, ITS America, pp. 517–529.
- Shapiro, S. C. (ed.) (1987). *Encyclopedia of Artificial Intelligence*, Vol. I,II. John Wiley and Sons, New York, NY.
- Spring, G. S. (1993). Validating expert system prototypes using the Turing Test. *Transportation Research* 1C(4): 293–301.
- Spring, G. S. (1997). Critical review of expert system validation in transportation. *Transportation Research Record* 1588: 104–109.
- Sumner, R. L., Jacobson, L. & Berg, D. B. (1995). Seattle ATMS: A multi-jurisdictional information system for traffic. *Proceedings of the Annual Meeting of the ITS America*, Vol. II, ITS America, pp. 799–806.
- TAB (1997). Telematics Applications in BAvaria, SCotland and Others. TR1054. Urban integrated traffic control, Validated models report, deliverable 08.1.
- TRB (1997). Advanced transportation management systems and transportation system management, Series: Transportation Research Record: no. 1603 0361-1981. National Academy Press. Washington, DC. A peer-reviewed publication of the Transportation Research Board.
- Tsavachidis, M., Hoops, M. & Keller, H. (1998). Coordinated traffic management in the greater area of Munich, in C. Hendrickson & S. G. Ritchie (eds), *Proceedings of the Fifth International Conference on Applications of Advanced Technologies in Transportation*. American Society of Civil Engineers, Newport Beach, CA. pp. 25–32.
- Turing, A. M. (1950). Computing machinery and intelligence, *Mind* 59: 433–460.

- UAG (1995). *Urban/Sys User Manual*, version 8.0 edn.
- Van Aerde, M. & Yagar, S. (1988). Dynamic integrated freeway/traffic signal networks: Problems and proposed solutions, *Transportation Research* **22A**.
- Wardman, M., Bonsall, P. W. & Shires, J. D. (1997). Driver response to variable message signs: a stated preference investigation, *Transportation Research* **5C**(6): 389–405.
- Webster, F. V. (1961). Traffic signal settings, *Technical Report 39*, Road Research Laboratory, HMSO, London.
- Webster, F. V. & Cobbe, B. M. (1966). Traffic signals, *Technical Report 56*, Road Research Laboratory, HMSO, London.
- Wentworth, J. A., Knaus, R. & Aougab, H. (1995). Verification, validation, and evaluation of expert systems. *FHWA Handbook Report FHWA-RD-95-196*, Federal Highway Administration, Office of Safety and Traffic Operations R&D, McLean, VA.
- Winston, P. (1975). *The psychology of computer vision*, McGraw Hill, New York, NY.
- Wittig, T. (1992). *ARCHON: an architecture for Multi-Agent Systems*, Ellis Horwood, Chichester, England.

IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc.
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved