2006

Using causal knowledge to improve retrieval and adaptation in case-based reasoning systems for a dynamic industrial process

Christopher A. Tighe
University of Windsor

Using Causal Knowledge to Improve Retrieval and Adaptation in Case-based Reasoning Systems for a Dynamic Industrial Process

By

Christopher A. Tighe

A Thesis
Submitted to the Faculty of Graduate Studies and Research
Through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario Canada

2005

Your file Votre référence
ISBN: 978-0-494-17056-4
Our file Notre référence
ISBN: 978-0-494-17056-4

Abstract:

Case-based reasoning (CBR) is a reasoning paradigm that starts the reasoning process by examining past similar experiences. Human reasoning often parallels the CBR methodology when a human remembers past experiences and uses them to establish a basis for problem solving. It could be stated that the process of CBR is the computerization, automation and formalization of certain features of the human thought process. The parallelism of human reasoning and CBR suggests that in real-life environments, CBR can be a viable reasoning scheme. CBR is often called a methodology, thereby giving an open interpretation of how a CBR system is implemented and the technologies used in such implementation. This openness of the CBR methodology permits a wide range of mathematical, statistical, and artificial intelligence treatments that can be utilized in the realization of CBR systems. The motivation behind this thesis lies in the observation that causal knowledge can guide case-based reasoning in dealing with large and complex systems as it guides humans.

In this thesis, case-bases used for reasoning about processes where each case consists of a temporal sequence are considered. In general, these temporal sequences include persistent and transitory (non-persistent) attributes. As these sequences tend to be long, it is unlikely to find a single case in the case-base that closely matches the problem case. By utilizing causal knowledge in the form of a dynamic Bayesian network (DBN) and exploiting the independence implied by the structure of the network and known attributes, this system matches independent portions of the problem case to corresponding sub-cases from the case-base. However, the matching of sub-cases has to take into account the persistence properties of attributes. The approach is then applied to a real life temporal process situation involving an automotive curing oven, in which a vehicle moves through stages within the oven to satisfy some thermodynamic relationships and requirements that change from stage to stage. In addition, testing has been conducted using data randomly generated from known causal networks.

Acknowledgements:

First of all, I would like to thank the members of the thesis committee for giving me a chance to present my work and for their efforts in evaluating, commenting, and improving my thesis.

I would like to thank the Faculty at the School of Computer Science (namely, Dr. Tawfik, Dr. Goodwin, Dr. Ezeife, Dr. Rueda, Ms. Chaturvedi, Dr. Ahmad, and Dr. Yuan) who through the coursework made my graduate studies at the University of Windsor an enriching experience. I'm proud to say that I graduated from the School of Computer Science at the University of Windsor. I'd also like to thank Dr. Kent for his assistance in finding a new supervisor when my original supervisor left the University.

Dr. Tawfik took me on as a grad student with very little knowledge of my abilities and interests. Initially my research directions were scattered and weak, but with great effort and assistance from Dr. Tawfik, progress toward this thesis was slow and steady. His patience with my somewhat slow pace is admirable. He presented challenges and assisted in such hurdles. Our weekly conversations kept my interest elevated, even when my attention was wandering. His assistance and candor has made my second stint at the University of Windsor an enjoyable experience. Thank you Dr. Tawfik.

I would also like to thank my co-workers at DaimlerChrysler Canada's Automotive Coatings Research Facility who assisted in the collection of data, proof reading of some materials, and general assistance to my studies.

Finally, I would like to thank my wife Vickie, for her insistence that I complete my graduate studies. Also, my son Joel, whose academic excellence and inquisitive nature has shown me there is still some intelligence in our genes. My wife and I are very proud of our son Joel. The continued support of my parents, Len and Gail in all aspects of my life, whether academic or otherwise, is always appreciated.

Christopher A. Tighe

December, 2005

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

Bayesian network, belief network	BN
Dynamic Bayesian network, dynamic belief network	DBN
Case based reasoning	CBR
Multiply Sectioned Belief Network	MSBN
Probability of X given Y	P(X\Y)
Parents of X	pa[X]
Conditional Probability Table	CPT
Case-base	$C = \{c_1, c_2, \ldots c_{N_c}\}$
Number of cases in a case base	N_c
Case	$c_i = \{a_{i1}, a_{i2}, \ldots, a_{iN_a}\}$
Number of attributes in a case	N_a
Problem case	$\mathrm{P} = \{p_1, p_2, \cdots, p_{N_p}\}$
Number of attributes in a problem case	N_p
Solution Case	$S = \{s_1, s_2, \cdots, s_{N_c}\}$
Case w/ persistent and dynamic attributes	

$$c_i = \{\rho_1, \rho_2, \cdots, \rho_{N_\rho}, d_{11}, d_{12}, \cdots, d_{1l}, d_{21}, d_{22}, \cdots, d_{2l}, \cdots, d_{t1}, d_{t2}, \cdots, d_{N_d l}\}$$

Persistent attribute	ρ
Number of persistent attributes	N_ρ
Dynamic Attribute	d

Number of dynamic attributes	N_d
Number of time intervals in a case	l
Number of Subnets	N_s
Combination level	m
Error from evidence error metric	ë
Variance from average error metric	ê
Total Error metric	ε

Chapter 1: Introduction

The design, tuning, and operation of industrial processes has been the focus of several case-based reasoning (CBR) systems [BCSV04][BSS02][HT95]. Frequently, a case-based approach ignores or mutates into a static structure the temporal aspects of industrial processes. However processes in general are often dynamic in nature and contain spatio-temporal and temporal relations that should be captured and exploited by the case-based representation and reasoning process to improve results.

Temporal case-based reasoning has been applied to weather prediction [RH02], prediction of air pollution levels [LAV94], waste water treatment [MCMC05] and prognosis of medical conditions [SG02]. These application domains are characterized by the availability of a large amount of historical data and the lack of a clear detailed understanding of the underlying causal mechanisms. For the industrial processes under consideration here, data availability is limited while a qualitative causal understanding of the process is easily obtainable.

Due to the complexity of a complete numerical analysis of the host of factors affecting an industrial process and the need to interactively tune the process online, CBR offers an attractive option. The main advantage of utilizing case-based reasoning in this work is that once a problem has been solved, it would be more efficient to solve a similar problem starting from the old solution rather than repeating a reasoning exercise from the first principles within the causal theory [Kol93]. Due to the scarcity of the data, it is useful to use our causal understanding to divide the problem case into a collection of sections with minimal inter-dependence between sections. Matching cases are retrieved for each section separately. A technique for combining compatible sub-cases is then applied as an adaptation step. Thus, both case retrieval and case adaptation are guided by the causal understanding of the industrial process.

The causal model is captured by a dynamic belief network (DBN). Techniques for dividing a DBN into a group of independent sections [Pea88] or minimally inter-dependent sections [Xia96] are instrumental in guiding the division of the problem case into a collection of sub-cases [TT05]. Each sub-case could span one or more time-slices of the dynamic belief network structure.

Attributes of a dynamic process include some persisting attributes in addition to the dynamically changing attributes [TT05]. The values of persisting attributes have to be compatible throughout the set of sub-cases chosen for a particular problem case. In adapting multiple sub-cases to the problem case, a measure of compatibility of sub-cases is necessary to meet the persistence constraints. By utilizing the above mentioned procedures for dynamic and persistent attributes, a complete solution case is constructed [TT05].

In Chapter 2 some necessary background topics are covered, including case-based reasoning, as well as causal modeling using Bayesian networks and dynamic Bayesian networks. Related literature is also overviewed in Chapter 2. Chapter 3 presents a unique industrial process application domain used as the test bed for this thesis. The CBR/DBN Retrieval/Adaptation algorithm is developed in Chapter 4 along with a detailed example and complexity analysis. Chapter 5 provides an in-depth evaluation of the CBR/DBN Retrieval/Adaptation algorithm including additional complexity analysis. Finally, the last Chapter presents some conclusions and possible future directions.

Chapter 2: Introductory and Background Topics

The CBR/DBN Retrieval/Adaptation algorithm is based on several Artificial Intelligence concepts, namely case-based reasoning, Bayesian networks (BN) and several ideas relating to or providing mechanisms for the manipulation of Bayesian networks. An understanding of these concepts is crucial to the development of the ideas behind the CBR/DBN Retrieval/Adaptation algorithm. A case-based reasoning system is the reasoning methodology used and the Bayesian network ideas and mechanisms are used to implement and improve the methodology. The first two sections of this chapter present the key concepts (CBR and Bayesian networks) fundamental to the development of the ideas presented in this thesis. Section 2.3 presents a short literature review and sections 2.4, 2.5 and 2.6 illustrate several tools that can be used in the handling and manipulation of Bayesian networks and dynamic Bayesian networks that are crucial for the implementation of this work.

2.1 Case-based Reasoning Systems and Dynamic Environments:

A case-based Reasoning (CBR) system remembers previous experiences or episodes called cases and uses them to assist in obtaining a solution to a current problem. The premise of case-based reasoning is that once a problem has been solved, it is often more efficient to solve the next similar problem by starting from the known solution rather than by repeating all the reasoning that was necessary the first time [Kol93]. Unlike many reasoning systems that try to generalize knowledge into rules or models, CBR uses specific knowledge about previous episodes to reason. By using specific episodes for reasoning, it is often not necessary to start the reasoning process from first concepts, like various other reasoning methods.

2.1.1 The CBR Cycle:

In 1994, Aamodt and Plaza [AP94] developed and presented one of the first structured views of the CBR process commonly referred to as "The CBR Cycle". This CBR cycle is still viewed as the basic methodology today, although there is research into the modification and expansion of this four step CBR cycle [PR01][RI01]. Figure 2.1 illustrates the basic CBR cycle as presented in [AP94].

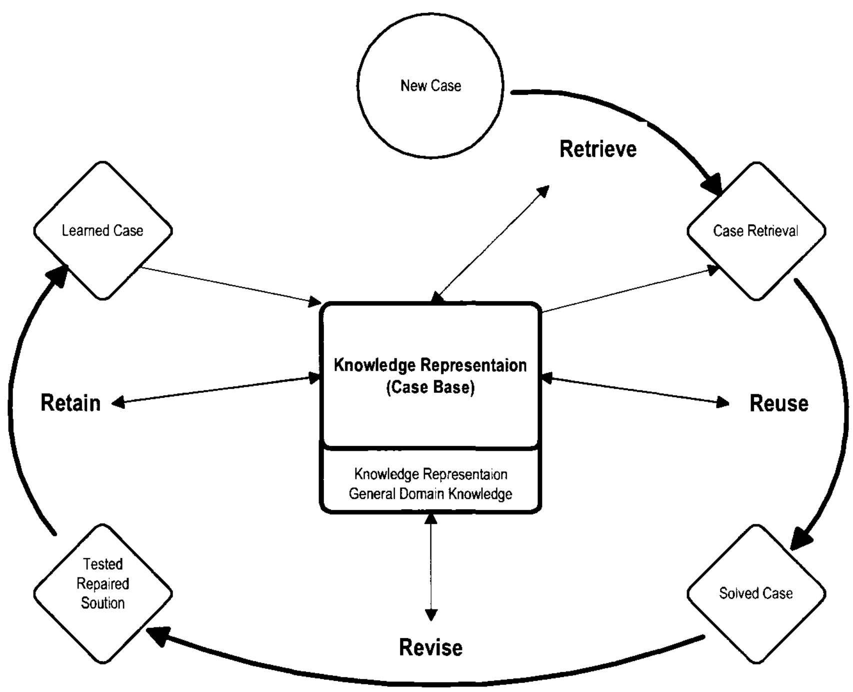


Figure 2.1: The CBR Cycle (adapted from [AP94])

As can be seen from the illustration, there are four main processes and a single central knowledge repository involved in the CBR process: Retrieve, Reuse, Revise and

Retain. The main knowledge is represented in the case-base. The four processes are often referred to as the four RE's.

2.1.2 Knowledge Representation (The Case-base):

The case-base is the back bone of the entire CBR process as it contains all the past experiences or episodes that can be used in the reasoning process. A case is a contextual piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoning process [Kol93]. A case records experiences that have the potential to help a reasoner achieve a goal or set of goals more easily in the future, that warn about the possibility of a failure or point out an unforeseen problem. There are three major parts of a case: the problem situation and its assessment, the solution to the problem, and the outcome. There can often be other parts to a case represented in the knowledge base such as reasoning statistics, derivational replays and other information that is specific to each CBR system.

There are many issues involved in the representation of cases, primarily what exactly to store in a case and determining and maintaining an indexing structure that facilitates the CBR process. A common problem in the CBR community is called the "indexing problem" or the problem of ensuring that a case is accessible whenever appropriate [LEA03]. Establishing a good indexing vocabulary is a highly active area of research.

In this work, a simple structural case representation is adequate, as a case is simply a vector with all attributes declared and indexed by the case number. An important aspect of the case representation is the inclusion of both persistent and temporal knowledge in the case base. This in itself is not unusual, but the treatment of the different data types for case retrieval and adaptation has innovative merit. Although not conventionally part of a CBR system, another representation used in this project is that of a dynamic Bayesian network which is presented in Section 2.2.1.

Formally a case-base can be represented by:

$C = \{c_1, c_2, \ldots c_{N_c}\}$ where c_i is an individual case and

N_c is the number of cases in the case-base

Each individual case can be represented by:

$c_i = \{a_{i1}, a_{i2}, \ldots, a_{iN_a}\}$ where a_{ij} is the j^{th} attribute in case c_i and

N_a is the number of attributes in a case.

Often the problem case or initial starting case is represented by:

$\mathrm{P} = \{p_1, p_2, \cdots, p_{N_p}\}$

where $N_p \leq N_a$, i.e. the problem case does not necessarily contain all attributes of a case c_i and

N_p is the number of attributes in the problem case.

The solution case can be represented by:

$S = \{s_1, s_2, \cdots, s_{n_a}\}$

For the purposes of this thesis, the case c_i is represented in a slightly different manner to highlight the persistent and dynamic portions of a case c_i. Therefore

$$c_i = \{\rho_1, \rho_2, \cdots, \rho_{N_\rho}, d_{11}, d_{12}, \cdots, d_{1l}, d_{21}, d_{22}, \cdots, d_{2l}, \cdots, d_{t1}, d_{t2}, \cdots, d_{N_d l}\}$$

where ρ are persistent attributes, N_ρ is the total number of persistent attributes; $d_{N_d l}$ are dynamic attributes, l is the number of time intervals in the case and N_d is the number dynamic attributes in each time interval.

2.1.3 Case Retrieval:

Case retrieval is the process of identifying or retrieving previous cases that are similar to the problem case and can be adapted to provide a solution to the problem case, or in general assist the CBR system in achieving its goals. This process starts with the problem case or problem description. Before past cases can be retrieved, situation assessment must be completed on the current problem to determine the entire context of the problem in a vocabulary that the CBR system can understand. In this process the current problem case must be "flushed out", or in other words all available information must be extracted in order to totally quantify the problem. The problem case is the starting point for situation assessment, but other knowledge acquisition techniques such as the use of assumptions, interpolation and guided discovery can assist in complete situation assessment. For example, assuming that the ambient temperature for a process is similar to the ambient temperature recorded at a local weather station could be a good assumption. Depending on the nature of the case-base, situation assessment could introduce additional complexity to the CBR process. For example, a conversational CBR system could require additional initial development in the form of a natural language interpreter to convert dialogue to text and then to a format meaningful to the CBR system. In a structural CBR approach the attributes are represented in a simple feature value matrix that is generally easy to use, manipulate and maintain. [SG02][Ber01].

Once the situation assessment process has been accomplished the CBR system is ready to retrieve the most similar case or cases using some predefined similarity metric. By using similarity assessment procedures, cases that can be useful in the reasoning process are retrieved from the case-base. These similarity assessment procedures can range from a simple nearest neighbor similarity algorithm to various introspective algorithms that use the data from the case-base to assist in the retrieval problem [LEA03]. Matching and ranking procedures are used to further identify cases that could be the most useful in the reasoning process. As in most computational systems, the speed and efficiency of the retrieval processes are important metrics and consequently research in case retrieval is an active area and is a main topic of this research.

This thesis addresses retrieval from two distinct directions, the first being the retrieval of best case(s) for the dynamic element of the environment and the second being the determination of the best case for the persistent attributes of the domain. The causal structure of the process in the case-base is utilized in the case retrieval phase to reduce the size of the retrieval process in terms of the number of attributes that need to be considered at one time.

Formally case retrieval can be represented by obtaining the best case(s) c_{best} defined as:

$$c_{best} = c_i \quad \text{s.t.} \quad \min\left(\underset{i=1,n}{\forall} \text{distance}(c_i, \mathrm{P}_i) \right)$$

where distance is some similarity measure such as the Euclidean or Manhattan distance. The Euclidean distance is computed as the square root of the sum of the squared differences between the two cases over each attribute. The Manhattan distance between two cases is the function of the sum of the distances on all attributes involved.

At an attribute level, case retrieval can be represented by obtaining the best case(s) c_{best} defined as:

$$c_{best} = c_i \quad \text{s.t.} \quad \min\left(\underset{j=1,N_a}{\forall} \left(\sum_{i=1,N_p} \text{distance}(a_{ji}, p_i) \right) \right)$$

It should be noted that the problem case P $= \{p_1, p_2, \cdots, p_{N_p}\}$ does not necessarily contain all attributes of the case $c_i = \{a_{i1}, a_{i2}, \ldots, a_{iN_a}\}$ and hence retrieval and adaptation steps must consider and handle this predicament.

2.1.4 Case Reuse:

This phase is more commonly referred to as case adaptation, where the retrieved case is adapted by evaluating the differences between the retrieved case or cases and the problem case and modifying the retrieved case so the solution to this modified case can be used as a solution to the problem case. Case adaptation is perhaps the least developed phase of the CBR process, and it is an active and dynamic research focus. Adaptation is a complex and crucial step of case-based reasoning which is generally studied in the restricted framework of a particular application domain [FLMN99]. In fact, research is often hindered by this domain specific nature and hence some researchers do not consider case adaptation phase as necessary for an operational CBR system [HB02]. Domain independent case adaptation or automated case adaptation is perhaps the ultimate goal of research on adaptation in CBR. The effort and computational power required to perform case adaptation is often complex and could possibly outweigh the benefits. For this reason, retrieval only fielded CBR systems are popular and abundant [Whi05].

There are several basic adaptation techniques, such as substitution, where a component from the problem case is replaced with a similar component from the retrieved case [Kol93]. Interpolation could be used to determine a substitute value for the problem case. A concrete example of this type of adaptation could be that a retrieved solution's speed attribute has a maximum speed of 50 km/h and hence a 75 km journey would take about 1.5 hours. However the problem case's speed attribute indicates a maximum speed of 75 km/h, by interpolation a 75 km journey would take one hour for the retrieved case, so perhaps one hour could be substituted for 1.5 hours in the solution case. Transformational adaptation is the process of transforming a retrieved solution to fit a problem situation's constraints by making deletions, insertions or transforming some element of the retrieved solution [Kol93][HB02]. For example, the problem case may indicate that one needs to travel from Windsor to Toronto (a distance of approximately 400km) in two hours, however the retrieved solution has attributes that indicate that the automobile in possession can travel at a maximum of 100 km/hr. A transformation type adaptation would be to take an airline from Windsor to Toronto, thereby meeting the two hour constraint. Another common adaptation method is that of derivational replay, where the methods or means used to solve the retrieved case are also used to solve the target case. For example, the problem case may indicate that one needs to travel from Windsor to a southern destination such as Fort Myers FL. The retrieved case indicates that we traveled from Detroit to Miami previously using a particular airline. Perhaps by using the same reasoning (affordability, convenience, experience etc.) conduit for choosing the airline to Fort Myers FL, one can choose the same or perhaps a different airline that took the person to Miami, FL. In order to use this method, the retrieved case must record the inferences or computations that were used to solve the retrieved case.

An interesting idea stemming from a derivational replay type adaptation strategy is presented in [CJR01] in which the information used to perform adaptation is stored in a separate case-base. In essence there is an adaptation case-base in which previous adaptation derivations are stored and reused when appropriate. This work also recognizes the difficulties in adaptation and provides several secondary methods for adaptation; mainly a rule-based system is used when the adaptation case base method fails. If the rule-base adaptation system fails, then the system defaults to a manual adaptation process. A system called DIAL uses this case adaptation approach in the domain of disaster planning.

Other than the three adaptation strategies described above, most adaptation is usually highly domain dependent (e.g. domain model driven or domain rule based driven). The predominant strategy is to use a rule based system where the rules are determined by domain experts in an if-then format.

Formally case adaptation could be considered as:

$$c_{adapted} = f_{\text{Adaptation Method}}\left(c_{retrieved}\right)$$

$$\text{where similarity}\left(\mathrm{c}_{\text{adapted}}\right) > \text{similarity}\left(\mathrm{c}_{\text{retrieved}}\right)$$

Although the adaptation phase is somewhat oblique in this work, it is considered as a collective step that uses causal knowledge to assist in the unified retrieval and adaptation phase. The combination of the sub-cases from the dynamic aspects of the problem with the case that best represents the persistent portion of the given problem is what is considered the adaptation problem for this thesis.

2.1.5 Case Revision:

Case revision occurs when a derived solution from the case reuse phase is determined to be incorrect [AP94]. This phase involves the evaluation of the solution to the problem, often by systems outside the CBR system. The fault with the solution is then repaired and evaluated again or the CBR system reverts back to the reuse phase. Case revision is not an active aspect of this thesis.

2.1.6 Case Retainment:

This is the learning phase of the CBR process. The case retainment phase ensures that the CBR reasoner becomes more efficient over time. The breadth and depth of the CBR system can be improved over time by properly utilizing case retainment. The case-base coverage expands as an increasingly diverse set of cases are solved and retained. CBR systems can learn not only from correct cases being added to the case base, but from other information such as how the reasoning process was completed (derivational replay), even faulty case reasoning processes and possibly the use of statistics about the reasoning process. Case retainment will not be dealt with in this work.

2.1.7 Extensions to the Four Step CBR Process:

There have been several extensions to Aamodt and Plaza's original four step CBR process. Most extensions deal with the maintenance to CBR systems [PR01][RI01] which is often implied in the original four step process. There are several types of maintenance that need to be performed to a CBR system. The first type is that of updating the indexing structure for the case-base to ensure that it properly represents the case-base and that access to all meaningful cases is still relevant. Another maintenance task is to monitor and update the cases in the case-base, perhaps by removing unnecessary or misguided cases from the case-base. Another addition to the four step CBR process is that of recording statistics about the reasoning process that could be used at a later date to assist in the reasoning process. In [Bri05] it was amusingly suggested that perhaps the four step process could be expanded to an eleven step process.

Another popular extension to the four step CBR process is the idea of having hybrid CBR reasoning systems in which other reasoning techniques are used in conjunction with CBR to create a reasoning system [AD98] [Aha98] [Fox00] [HS04] [LS03] [Man01]. In [Man01] fuzzy logic is used to characterize imprecise and uncertain information in case representation. Case retrieval uses fuzzy matching techniques and degrees of similarity present in attributes of retrieved cases. This research also outlines several CBR systems that use fuzzy logic concepts including; the ARC System, the BOLERO system, the CAREFUL system, the CARS System and the FLORAN System. CBR is integrated into Decision support systems in [AD98] and [LS03] presents a CBR as a hybrid classifier that combines Bayesian networks with distance based algorithms.

2.1.8 CBR, DBN and a Dynamic Industrial Domain:

Case-based reasoning is an appropriate reasoning method for the parameterization of an industrial process mainly because of the complexity of the interactions within such processes. Although many elements of the domain are typically available, the interactions involved in the entire model can become complicated. To create a detailed model or a rule based reasoning system for the entire process would entail a great amount of research, study and measurements to quantify the interactions of the subsystems involved. The computational power required to model just a single element in a process step could easily exceed the computational power available, and the results are difficult to validate.

To-date there have been several industrial process related CBR systems [BSS02][HT95][BCSV04][Wat99] developed and presented as fielded applications. In [HT95], perhaps the first published CBR application, a CBR system was developed to determine the optimum loading of an autoclave (cure oven) that satisfies the cure requirements of all parts. In [BSS02][BCSV04], a tire production process was optimized using a CBR methodology. Most of these applications present the basic knowledge representation unit of the case-base as a static case, where the attributes in a single case remain constant throughout the case. Generally there is no aspect of time or time slices in a case and if time is required to be represented it is done statically and finitely with additional attributes. However, processes in general are often dynamic in nature and contain spatial and spatial-temporal relations that require unique representation. By utilizing dynamic representations of attributes, the temporal qualities of processes can be developed in a robust approach.

The use of dynamic Bayesian networks to support the retrieval and adaptation phases of CBR is a research area that seems to be appropriate for a sequential process type environment. The dynamic nature of a sequential process matches the reasoning assumptions of a DBN.

2.2 Bayesian Networks:

A Bayesian network (BN), also called belief network, Bayesian belief network and causal probabilistic networks are used to model a domain containing uncertainty due to imperfect understanding of the domain, incomplete knowledge of the state of the domain at the time when a given task is to be performed, randomness in the mechanisms governing the behavior of the domain, or a combination of these [Kja95]. Bayesian networks are compact probabilistic graphical models used to represent probabilistic, uncertain or causal relationships between variables [SA00]. Bayesian networks have been a topic of intense research for many years and seem to be integrated in many modeling and reasoning mechanisms. Hence the mechanics of BN are for the most part well understood and documented, but computationally complex [Coo90]. Generally, inference in BN is intractable [Coo90]. However, some practical algorithms exploit the properties of a network to provide exact or approximate results in a reasonable time frame [Jen02].

A Bayesian Network is a directed acyclic graph in which each node represents dependencies between variables and their associated probabilities [Gom04][RN03]. Each node in a Bayesian network is associated with either a continuous or discrete random variable that has a conditional probability table (CPT) assigned to it. There is a set of arcs or directed links that connect pairs of nodes representing probabilistic or causal dependencies between the variables. More concisely, if there is a link from X to Y, then there is a direct influence on Y from X, or X is a parent of Y. The conditional probability table (CPT) for a given node X defines the probability of that node X given all possible combinations of values of its parent nodes. To specify the joint probability distribution in a Bayesian network, one must give the prior probabilities of all root nodes, and the conditional probabilities of all non-root nodes given all possible combinations of their direct parents. Figure 2.2 illustrates a BN from the domain of paint applications. In this BN model, there are four variables or nodes of some significance in the paint application domain. For example; fluid rate (fr) is the rate at which the paint leaves the applicator, filmbuild (fb) is the thickness of the paint after it is applied, target distance (td) is the distance from the applicator to the part, appearance (ap) is whether the coating meets appearance standards and durability (dr) is whether the coating meets durability standards. These nodes are connected by arcs (causal relationships); like the arc from fluid rate to filmbuild indicates that fluid rate may be a cause of filmbuild, or stated another way, filmbuild could possibly be dependent on fluid rate. As another example of the causality present in the BN model, consider the node appearance with an arc from filmbuild indicating that paint appearance is dependent on filmbuild. The conditional probability tables associated with each node are read as such; P(fb|~fr,td) from the filmbuild CPT indicates that the probability of filmbuild being acceptable given an inadequate flow rate and an appropriate target distance is equal to 0.79. In summary, the probabilistic meanings are determined using the directional arcs and the conditional probability tables. The causal relationships are determined by the directional arcs.

The attributes in a Bayesian network are often considered to be binary (that is either true or false) as in Figure 2.2. This affords simplicity in the handling of the network. Multi-valued variables can be handled in a similar manner as binary without much additional effort. The representation of continuous variables in a Bayesian network is handled by specifying a distribution over the variable. Often a linear Gaussian distribution is used, in which the child has this distribution whose mean μ varies linearly with the value of the parent and whose standard deviation σ is fixed [RN03]. Discretization, by dividing a continuous variable into fixed intervals can also be used to handle continuous variables. For example, a continuous variable such as appearance, (generally, measured on a continuous scale from 0 to 100) can be discretized into intervals such as the following: Poor (0 – 25), Fair (25 – 50), Good (50 – 75) and excellent (75 – 100) creating a multi-valued discrete variable.

Formally, a BN is defined as: a directed acyclic graph G whose nodes V are random variables and edges E represent probabilistic or causal relationships among the nodes V. In this graph, $G = \{V, E\}$, the direct influence on a node $v \in V$ are known as the parents $pa[v]$. The model comprises the probability distribution of each node using the configuration of its parent nodes i.e. $p(X_v \mid X_{pa(v)})$. The conditional independence assumptions represented by the graph correspond to a joint probability function i.e $p(X_v) = \prod_{v \in V} p(X_v \mid X_{pa[v]})$. Each node in a network is associated with a conditional probability table that specifies the probability of that node given its parents i.e. $p(v_i \mid pa(v_i))$. For nodes with no parents, the prior probabilities are used.

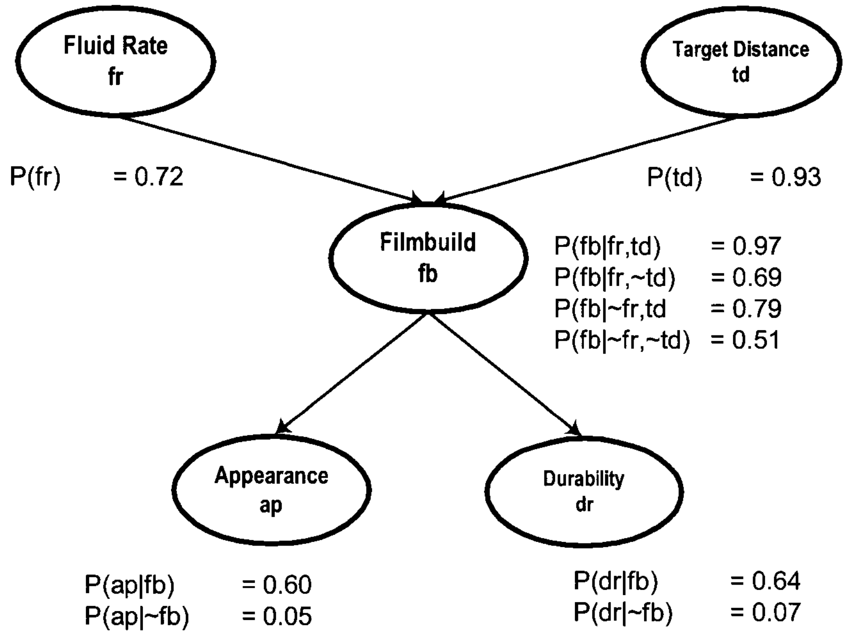


Figure: 2.2: A Bayesian Network Modeling a Paint Application Process.

Inference tasks in BN include assessing the posterior probability of a node given some evidence, finding the most probable values for all nodes in the network (or most probable explanation, MPE) and finding the most likely value for a node or a set of nodes given some evidence (also known as maximum a posteriori probability, MAP). Although inference in a Bayesian Network is known to be intractable [Coo90], by utilizing certain independence assumptions inherent in Bayesian networks and evidence acquired to date, efficient reasoning can be accomplished [Cha00]. Inference in a BN is the computation of the posterior probability distribution for the set of query variables, given a set of evidence variables for which the exact value is known. An inference algorithm is used to propagate these values through the BN, according to Bayes rule. There are several inference algorithms, but they basically fall into two categories: exact inference and approximate inference.

First, consider the case when we know the values of all variables associated with a BN, for example consider Figure 2.2. Suppose we want to find the probability that appearance is acceptable, durability is inadequate, filmbuild is ok, fluid rate and target distance is not acceptable.

$$P(ap, \sim dr, fb, fr, \sim td)$$

from $P(x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(X_i))$

$$\begin{aligned} & P(ap \wedge \sim dr \wedge fb \wedge fr \wedge \sim td) \\ & = P(ap \mid fb) * P(\sim dr \mid fb) * P(fb \mid fr, \sim td) * P(\sim td) \\ & = 0.60 * 0.07 * 0.69 * 0.07 \\ & = 0.0020286 \end{aligned}$$

It can be stated that the probability of appearance, filmbuild and fluid rate being true and the probability of durability and target distance being false is equal to 0.0020286.

In general, one does not always know the values of all variables in a BN network. There are several methods of exact inference that can be used to determine the probability given some evidence such as the variable elimination algorithm, the variable elimination algorithm with eliminating repeated calculation and clustering algorithms [RN03]. To provide some understanding of this category of algorithms an example is provided to detail the operation of the variable elimination algorithm.

Example (Variable elimination algorithm): Using Figure 2.2, and the known values of appearance = true and durability = true, the probability of filmbuild will be determined.

In general one wants to find $P(X \mid e, Y)$

$$\begin{aligned} & X = \text{the query variable (fb)} \\ \text{where } & \text{e} = \text{the evidence variable (ap \& dr = true)} \\ & \text{Y} = \text{hidden variables (fr \& td)} \end{aligned}$$

$$P(X \mid e, Y) = \alpha \sum_{Y} P(X, e, Y) \qquad \text{where } \alpha = \text{a normalizing (to 1) constant}$$

$$P(fb \mid ap = true, dr = true) = \alpha \sum_{fr} \sum_{td} P(fb, fr, td, ap, dr)$$

$$= \alpha P(fr) \sum_{fr} P(td) \sum_{td} P(fb \mid fr \mid td) P(ap \mid fb) P(dr \mid fb)$$

$$= \alpha * P(fr) * \begin{pmatrix} (P(fb \mid fr, td) * P(ap \mid fb) * P(dr \mid fb) * P(td)) \\ + (P(\sim fb \mid fr, td) * P(ap \mid \sim fb) * P(td)) \\ + (P(fb \mid fr, \sim td) * P(ap \mid fb) * P(dr \mid fb) * P(\sim td)) \\ + (P(\sim fb \mid fr, \sim td) * P(ap \mid \sim fb) * P(dr \mid \sim fb) * P(\sim td)) \end{pmatrix}$$

$$= \alpha * 0.720 \begin{pmatrix} (0.970 * 0.600 * 0.640 * 0.930) \\ + (0.490 * 0.050 * 0.070 * 0.930) \\ + (0.690 * 0.600 * 0.640 * 0.070) \\ + (0.310 * 0.050 * 0.070 * 0.070) \end{pmatrix}$$

$$= \alpha * (0.720) * (0.34664064 + 0.00159595 + 0.0185472 + 0.00007596)$$

$$= \alpha * (0.720) * (0.366245)$$

$$= \alpha * (0.26396964)$$

Repeat the same calculation for ~fb.

$$= \alpha * (0.083058416)$$

Therefore one has $P(fb \mid ap = true, dr = true) = \alpha * \langle 0.26396964, 0.083058416 \rangle$

Now use α to normalize to one. $\alpha = 2.8816113934$

Consequently, $p(fb \mid ap = true, dr = true) = \langle 0.7606, 0.2394 \rangle$

This approach to Bayesian network inference has a complexity of $O(n2^n)$, where n is the number of vertices which quickly becomes intractable for any significant value of n [Cha00]. There are some improved algorithms for exact inference but these generally lead to exponential time.

There are also several methods of approximate inference that can be used to determine the probabilities of nodes within a Bayesian network. Direct sampling, rejection sampling, likelihood weighting and Markov chain Monte Carlo are some common algorithms for approximate inference in a BN [RN03].

In this work we use an extension of the BN called a dynamic Bayesian network that allows for the representation of temporal aspects of a given process while still permitting the exploitation of the mechanisms for the manipulation of a BN

2.2.1 Dynamic Bayesian Networks:

Dynamic Bayesian networks (DBN) represent a probability distribution over time. They consist of a BN for each time slice where temporal variables link each individual time slice. These edges represent persistence and causation [DK98]. The process of expanding from a single time slice to a semi-infinite DBN is called the unwrapping of the network structure. Basically, the BN is copied once for each time slice and temporal variables are used to link each of the time slices.

Formally, in a DBN the state at time t is represented by a set of random variables $Z_t = \{Z_{1,t}, Z_{2,t}, \ldots, Z_{d,t}\}$. Typically it is assumed that a state at time t depends only on the previous state (a first-order Markovian assumption, see section 2.2.3) i.e. $P(Z_t) = P(Z_t \mid Z_{t-1})$. Another typical assumption is that the process is stationary, that is the transition process is the same for all time slices. The set Z_t is typically divided into two sets of variables: the unobservable, transient or state variables X_t and the observed or evidence variables E_t. The state variables depend on previous states i.e. $P(X_t) = P(X_t \mid X_{t-1})$ and the evidence variables at time t depend only on the current state i.e. $P(E_t) = P(E_t \mid X_{0:t,} E_{0:t-1}) = P(E_t \mid X_t)$ sometimes referred to as the sensor model or observation model [RN03]. The joint distribution for the DBN can then be formulated as:

$$\mathbf{P}(\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_t, \mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_t) = \mathbf{P}(\mathbf{X}_0) \prod_{i=1}^{t} \mathbf{P}(\mathbf{X}_i \mid \mathbf{X}_{i-1}) \mathbf{P}(\mathbf{E}_i \mid \mathbf{X}_i).$$

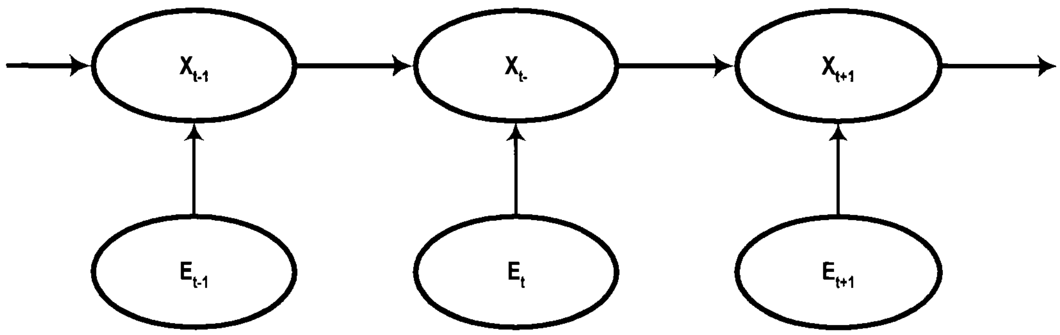


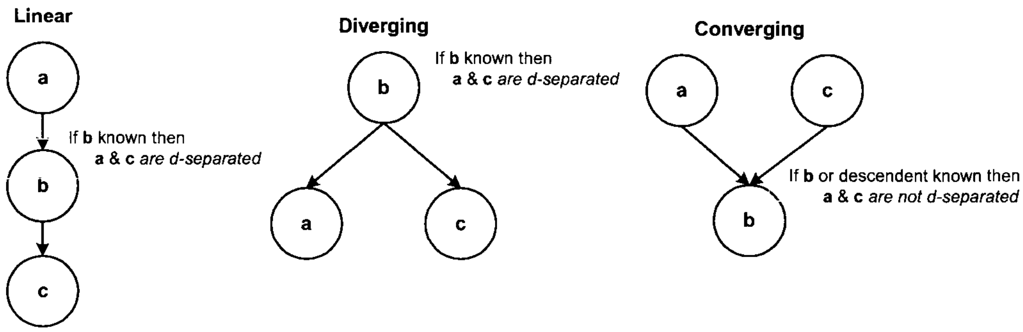
Figure 2.3: A Dynamic Bayesian Network

A dynamic model can be constructed from a set of building blocks that capture the instantaneous relationship between domain variables, together with a set of temporal dependencies that capture the dynamic behavior of the domain variables [BV98]. Accordingly, the building blocks for a dynamic Bayesian network are a set of static Bayesian networks. From a static Bayesian network, a time-varying dimension can be added by defining (a) significant events associated with each variable; and (b) the dynamic construction algorithm.

2.2.2 Independence Concepts:

An important aspect of Bayesian networks and dynamic Bayesian networks for the purposes of this research is the probabilistic independence properties that can be employed in BN and DBN to divide a large network into a set of smaller sub-networks. D-separation is used to determine whether two nodes in a network are independent given some evidence [Pea00]. Independence can be conditional independence or marginal (unconditional independence). Conditional independence of the two nodes x and y given z implies that known or specified attribute z causally separates x and y from influencing one another. Marginal or unconditional independence of two nodes x and y implies that $P(x \mid y) = P(x)$ and $\mathrm{P}(\mathrm{y} \mid \mathrm{x}) = P(y)$. Independence provides a powerful means for dividing a problem case into independent sub-cases. Thus, independence is instrumental in case retrieval as it helps in finding the closest and causally most relevant matching sub-case. Independence also plays a role in guiding the adaptation as these sub-cases are combined.

The concept of d-separation [Cha00] sometimes conversely referred to as d-connection [Pea00] is illustrated in Figure 2.4. Basically, if there is a linear configuration of nodes, such as in Figure 2.4a, and the value of random variable b is known (also referred to as has evidence) then nodes a and c are d-separated. When a and c are d-separated, it can be said that a is conditionally independent of c given b. A similar situation arises when there is a diverging configuration of nodes such as in Figure 2.4b. An inverse situation occurs if we have a converging configuration such as in Figure 2.4c. In this situation, if b or a descendent of b is known then a and c are not d-separated and hence are dependent on one another. These key ideas from the Bayesian network research play a key role in the CBR system developed in this thesis.



a) Linear **b) Diverging** **c) Converging**

Figure 2.4: **d**-Separation

2.2.3 Markov Assumption:

A Markov assumption, named after the Russian statistician, is that the current state depends on a finite history of previous states [RN03]. For example, in Figure 2.5a a first order Markov assumption dictates that the current state depends only on the previous state $P(X_t) = P(X_t \mid X_{t-1})$. Whereas in Figure 2.5b the second order Markov assumption specifies that the current state depends on the previous two states $P(X_t) = P(X_t \mid X_{t-1}X_{t-2})$. A Markov assumption is often a valid assumption. For example, in predicting the location of a robot, it is often dependent on where it was in the previous state only. However, is seems that the main undertaking of a Markov assumption is to reduce complexity of the problem state, perhaps at the risk of accuracy.

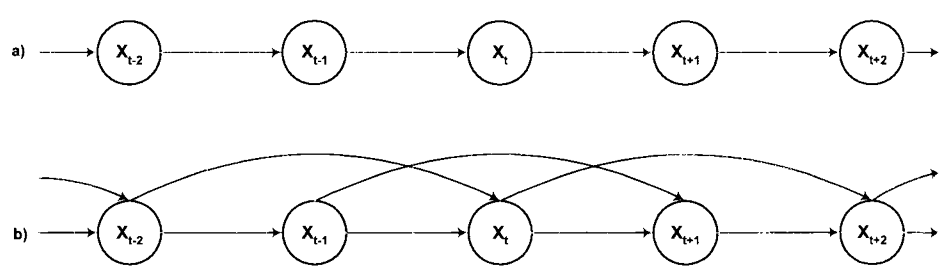


Figure 2.5: Markov Structure

a) First Order Markov DBN Structure b) Second Order Markov DBN Structure

2.2.4 Barren Nodes:

In [Sha86], Shachter defines a barren node as a sink node, that has no successors and hence no matter what value is assigned to the barren node variable, no other node is affected, so it may be removed from the diagram. For an example of a barren node, consider Figure 2.4a node c. By extending the notion of barren node to a barren sub network defined as a sub network that contains no evidence nodes that is connected to the rest of the network by a single incoming edge, it is possible to further divide a large dynamic Bayesian network into smaller more manageable sub networks. Once barren nodes are remove it is often the case that other barren nodes appear in the network structure which can also be removed.

2.2.5 Multiply Sectioned Bayesian Networks:

In general, the division of the dynamic belief network into independent sub-networks is not always possible. Fortunately, other techniques for dividing a belief network into sections can be adapted to the DBN used here. The multiply sectioned Bayesian networks (MSBN) process [Xia96] [Xia03] divides a belief network into a set of interrelated sub-networks by duplicating nodes at the interface between two sub-networks, provided that the set of nodes in each interface is a d-separation set and that the resulting collection of sub-networks forms a hyper tree. In the process of the MSBN decomposition of the DBN, two types of consistency constraints are imposed: global consistency constraints and local consistency constraints. Global consistency constraints are similar in behavior to persistent attributes while local consistency constraints ensure that nodes at the interface between two sections are assigned compatible values. Figure 2.6 illustrates the division of a dynamic belief network into two multiply sectioned sub-networks. The upper network is a segment of a DBN before the MSBN concept has been applied. The lower network is the same DBN segment with the MSBN concept applied. As can be seen by sectioning the network, the two temporal links $D_t -> A_{t+1}$ and $C_t \rightarrow C_{t+1}$ are sectioned with the variables at time t being repeated at time $t+1$ as $D_t^{'}$ and $C_t^{'}$. The posterior probabilities of the interface nodes C_t and D_t can be used as the prior probabilities of the repeated nodes $C_t^{'}$ and $D_t^{'}$. This technique allows us to separate conditionally independent sections of the DBN thereby creating subnets.

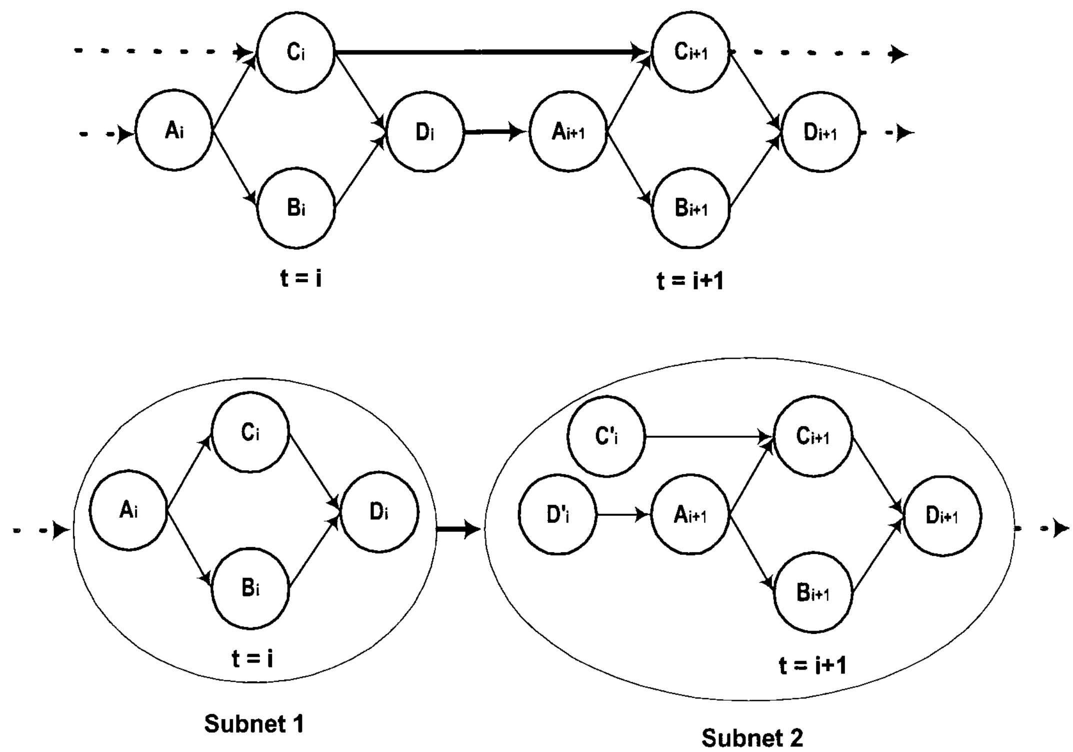


Figure 2.6: Multiply Sectioned Bayesian Network

2.3 Literature Review/Related Work:

Research in the field of Bayesian Networks for a variety of reasoning tasks is virtually limitless and there are several projects that integrate both BN and CBR in innovative ways. The combinations of static and dynamic indices in a CBR system together with the use of causal knowledge in the form of a DBN are new innovations introduced in this thesis. This section will review related research and compare it to the work presented here.

2.3.1 General CBR and BN Literature:

There is a great deal of literature on CBR systems with some general and milestone publications including [Kol93][LEA03],[AP94]. To date there are many fielded applications, some are illustrated in [Che01] [CVG01] [HT95] [BSS02] [BM01] [BCSV04] [Wat97] with a great deal of them having domains associated with a process situation similar to this project, however the domains represented are static in nature and adaptation is non-existent or does not relate to the causal structure of the domain. For example in [HT95], one of the first fielded CBR applications is presented and the domain is a cure oven similar to the one presented in this thesis. However, the temporal aspects of the domain are treated in a static manner using conventional CBR techniques.

For many years researchers have been looking to Bayesian networks to solve modeling situations and hence there are some excellent resources from basic text books [RN03], to lecture notes [Jen99], to research papers [Cha00] [Gom04] [AL98]. The topic of dynamic Bayesian networks is less developed, but there are many resources including [RN03] [Mur02] [DK98].

2.3.2 Literature Related to Ideas Presented in this Thesis:

The following research efforts have aspects that are similar to some of the innovations presented in this thesis, but in general are different in several distinct ways.

In [BH95], a diagnostic and troubleshooting application uses a stylized or layered BN set up as a cause, issue and symptom network using discrete variables and was developed by a domain expert. A single cause or symptom is used to select a sub section of the BN that is relevant to the cause or symptom. By using d-separation, variables that are relevant to current observations are identified. Using the constructed Bayesian network, the system generates recommendations for components to repair and makes suggestions regarding additional observations, by using the cost of observations. This work utilizes the independence assumptions of a causal network to reduce the BN to a more manageable Subnet similar to the work. This work does not include direct treatment of dynamic variables in a time slice nature and the handling of static variables is unique to the CBR/DBN Retrieval/Adaptation algorithm. The CBR/DBN Retrieval/Adaptation algorithm can use discrete or continuous variables.

In [BV98] a framework is described that combines Bayesian networks and case-based reasoning to create a knowledge representation scheme capable of dealing with time-varying processes. This scheme uses temporal Bayesian networks as individual cases in the case-base. This thesis uses temporal Bayesian networks and independence assumptions to improve retrieval in a CBR system. However, it does not represent an individual case as a temporal Bayesian network, but rather models the entire case-base as a dynamic belief network.

In [JAS02] a well known time interval theory from [All84] is used to develop a time based representation for cases in a CBR system. A retrieval algorithm based on Allen's thirteen possible time relationships is presented and demonstrated in an oil well drilling application. Although the work in [JAS02], utilizes time concepts in the retrieval of cases, more specifically interval based time concepts, this thesis uses a time instance or point based representation of time to represent temporal relationships. This work also uses dynamic Bayesian networks and the independence assumptions inherit in BN to improve the retrieval and adaptation.

In [SA00] the application domain is that of user profiling, where the idea is to build a repository of information (a case-base) regarding the users access and querying of a database. When a new query is made new attributes are added to the BN and the probability values associated to each node of the BN are updated. By utilizing the BN and its conditional probability tables, cases are matched and ranked as to possible or suggested next queries. Again there is no dynamic or time slice innovation in these ideas. The independence assumptions utilized in the CBR/DBN Retrieval/Adaptation algorithm are not eminent in their process and the complete probability distribution functions are required.

In [Gom04] Word Net [MBFCM90] and the case library are used to build a BN. Once the BN is constructed, a class diagram (query diagram) is used for the initial query. The evidence corresponding to the synsets of the query diagram objects are set to true. Once the evidence nodes are set the probability of the case nodes are calculated and such probabilities are used to retrieve and rank corresponding cases. The BN used in this project are static and the complete probability distribution is obtained and used to rank cases. Once again, in this project there is no dynamic aspect to the data and a complete probability distribution function is required. The interesting idea here is that the BN is learned from Word Net, the learning of the DBN is a future phase of the CBR/DBN Retrieval/Adaptation algorithm.

A dynamic Bayesian network containing the observable attributes of the domain model is created along with a user created static semantic model in [AL98]. Each case is indexed by the Bayesian network using binary features. In the first step of the retrieval process the BN is used for retrieving a set of relevant cases after the observed features are entered into the BN as evidence. This similarity metric is based on the calculation of the probability of the case being on, P(Case node is ON| Features of New Case). All cases having similarity metric greater than some threshold value are retrieved in this first step. In the second pass the number of cases is reduced using perhaps a BN approach like discriminating between influencing nodes. The CBR/DBN Retrieval/Adaptation algorithm utilizes all attributes of the domain and does not index the cases using the DBN. Independence assumptions utilized in the CBR/DBN Retrieval/Adaptation algorithm are used to partition the network.

In [MCMC05], temporal case based reasoning is presented from the perspective that a temporal episode consists of a collection of static cases. The reasoning in this work utilizes the episode as its main reasoning block. Much emphasis is placed on the utilization of the episode representation hierarchy or structure and maintaining such structure. The temporal episode is separated into static cases. In this thesis a case consists of the entire temporal sequence with both the dynamic and persistent features represented in a single case. Case representation is not the focal point of this thesis. The separation of the dynamic and persistent attributes of a case and the manipulation via the use of causal knowledge available in the case-base is the innovative approach taken in this thesis.

There are several other research efforts that utilize BN and CBR, but as seen above, are generally different in the ideas of the dynamic/time sliced BN, the lack of use of independence assumptions given evidence, often require the complete probability distribution and the handling of persistent variables in a dynamic realm. These papers include [LS03] [Din98] [Sha86] [WOB98].

2.4 Hugin Expert Systems:

The Hugin Development Environment provides a set of tools for constructing model-based decision support systems in domains characterized by inherent uncertainty [Kja95]. The models supported are Bayesian networks and their extension, influence diagrams. The Hugin Development Environment consists of the Hugin decision engine, Hugin APIs (Application Program Interfaces) and Hugin Graphical User Interface. Of interest is the Hugin API, in which a set of routines are available for use by other programs (C, C++ or Java). In particular, the d-separation function is interesting, in that one could input a BN and then determine which nodes are d-separated from other nodes. Figure 2.7 illustrates this function in action in the graphical user interface of Hugin. Basically, the crossed (red) nodes are d-separated from the source nodes which are the nodes without an accompanying ellipses. Due to cost restrictions, the Hugin API is not available for use in this work, but in theory could be integrated to determine the d-separation or conditional independence of attributes of a BN or DBN.

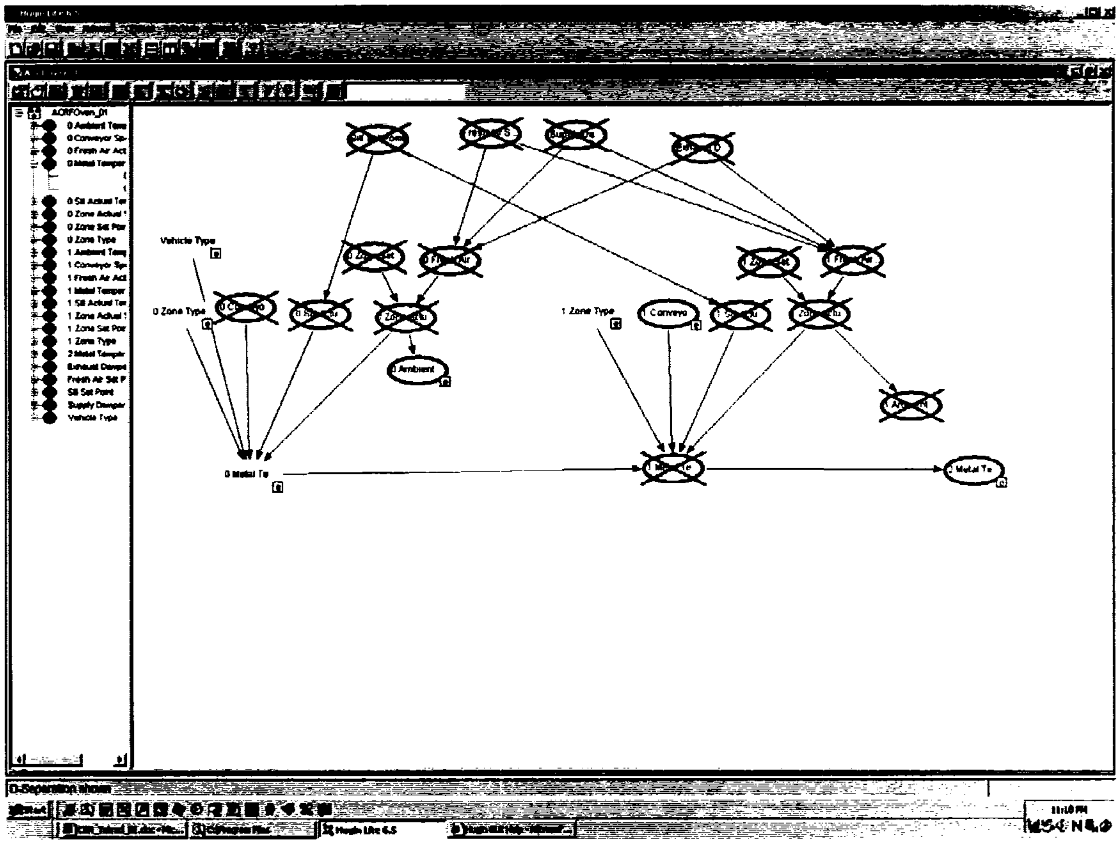


Figure 2.7: Hugin d-separation

2.5 Probabilistic Network Language:

Intel Corporation has developed Probabilistic Network Language (PNL) as a set of routines for handling Bayesian Networks. The library contains high-performance implementations of algorithms for working with Bayesian networks and Markov networks, such as belief propagation and junction tree inference, maximum likelihood and expectation maximization [Int03]. Although PNL is available for free, it is highly dependent on Microsoft Visual C++ and hence does not integrate with the application developed in this thesis (using Borland C++ Builder). PNL contains a function called d-connection that can be used to determine the independence of nodes in a BN. Again, it is possible in theory to utilize the routines available in PNL to resolve d-separation among attributes in a BN.

2.6 Tetrad:

Tetrad is a program for creating, simulating data from, estimating, testing, predicting with, and searching for causal/statistical models [SSGM94]. Tetrad IV is the latest version of this software designed in Java and comes with a pleasant user interface. Tetrad III comes in a DOS or Solaris based versions and has an application program interface that consists of a library of routines that can be used with other software. Interest in Tetrad III was high due to the integratability and a function called "Build". The purpose of the build function is to take a set of data as input and output a set of causal models based on the input data. The idea behind the use of the build function is that a case-base could be taken as input to the build function and produce or learn a causal model or perhaps a Bayesian network that could be used for further reasoning. However after much experimentation with Tetrad, the build function and real life data, some problematic concerns such as the inability to determine the direction of causal arcs, difficulties in handling continuous variables and weaknesses in dealing with latent variables have surfaced. Further investigation of programs such as Tetrad with real life data sets is required to be able to utilize such tools in a real life application.

Tetrad can also be used to generate data from a known belief network structure. By starting with a network structure as illustrated in Appendix C.1, and using the MakeModel function, Tetrad can generate a set of fully parameterized recursive linear structural equations, refer to Appendix C.2 and C.3 for an example. This information can then be fed into the Monte function to generate a set of data coherent with the original belief network structure. This procedure is used to produce data for the evaluation of the methods presented in this thesis.

Chapter 3: The Domain for the CBR Application

During the process of painting a vehicle in an automotive assembly plant, the paint coatings are applied to a vehicle body, then the entire body is processed through a sequential oven to cure the coating. Proper cure is essential to produce a high quality painted surface; both paint appearance and durability are important qualities of a paint coating. Defects such as poor appearance, over-cure and surface defects are often associated with improper cure. Automotive paint curing ovens are typically made of several zones (often 4 to 10) that are controlled separately. These oven zones could use a variety of different types of heating processes, such as infra-red, black wall radiant, convection or some other proprietary heating methods and the zones can be of different lengths.

At the DaimlerChrysler Canada Inc./University of Windsor Automotive Research and Development Centre in Windsor, Ontario, there exists a highly automated paint laboratory called the Automotive Coating Research Facility (ACRF). This facility is dedicated to the research and development of the processes of applying paint coatings, which includes the curing of such coatings. The paint curing oven in the ACRF is a four zone, highly flexible oven used to cure numerous different vehicles and simulate many of the ovens located in the DaimlerChrysler assembly plants. The curing process is a dynamic process consisting of a vehicle on a conveyor that passes through the oven, where various parameters associated with the oven are monitored and adjusted to achieve the proper cure. These parameters include but are not limited to: conveyor speeds in each zone, temperature set points and actual values for each zone, and damper settings. The

ACRF oven has four distinct oven zones. The first two are radiant and the final two are convection with two additional heat sources feeding all four zones. Figure 3.1 illustrates a schematic and Figure 3.2 shows some photos of the ACRF curing oven. Although not typical of assembly plant ovens, the ACRF oven conveyor has a variable speed drive and hence the speed of the conveyor can be changed in a dynamic fashion. Table 3.1 details all the parameters that are used to control the oven.

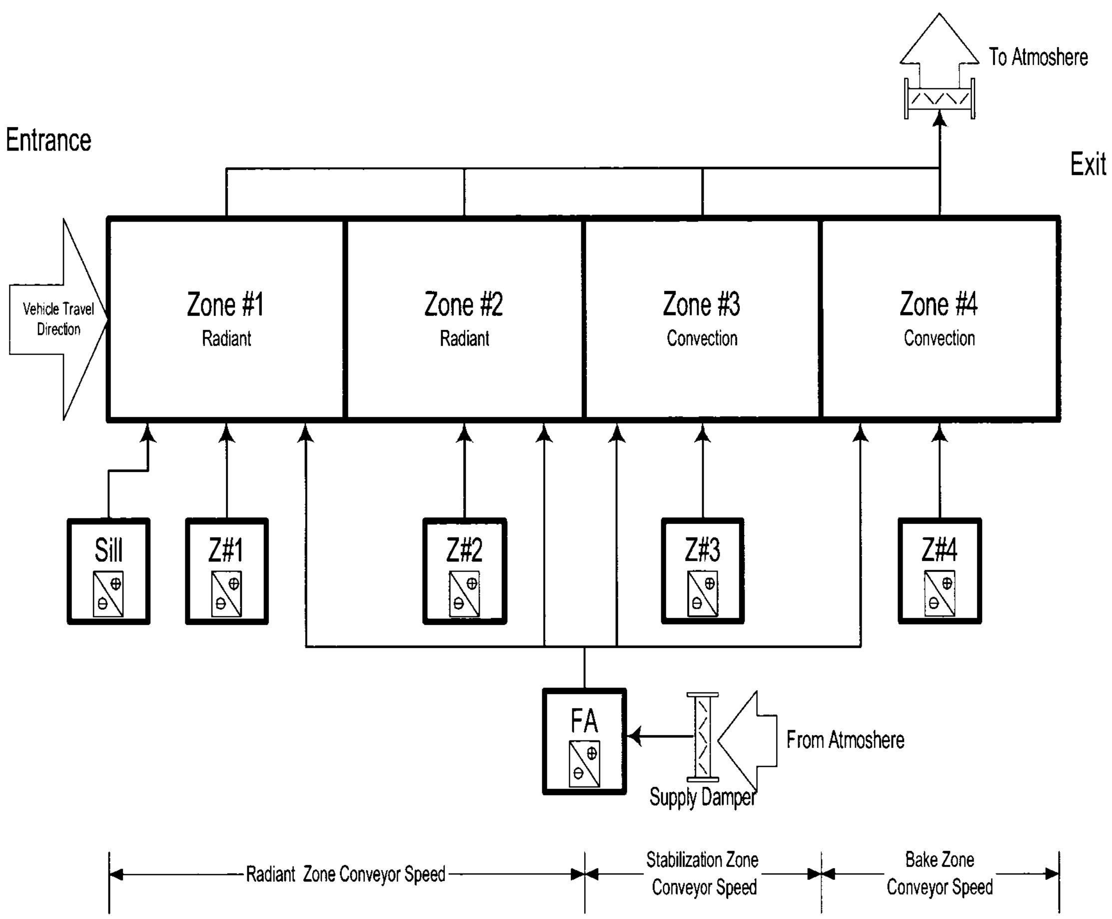


Figure 3.1: ACRF Oven Schematic

Figure 3.2: ACRF Oven Photos

		Units	Comments
Oven Parameters			
Zone #1	Set point	°F	The Set point for Zone #1
	Actual	°F	The Actual temperature for Zone #1
Zone #2	Set point	°F	The Set point for Zone #2
	Actual	°F	The Actual temperature for Zone #2
Zone #3	Set point	°F	The Set point for Zone #3
	Actual	°F	The Actual temperature for Zone #3
Zone #4	Set point	°F	The Set point for Zone #4
	Actual	°F	The Actual temperature for Zone #4
Sill Zone	Set point	°F	The Set point for the Sill Zone
	Actual	°F	The Actual temperature for the Sill zone
Fresh Air	Set point	°F	The Set point for the Fresh Air Zone
	Actual	°F	The Actual temperature for Fresh Air Zone
Supply Damper		%	Fresh Air Supply Damper opening
Exhaust Damper		%	Exhaust Air Damper opening
Conveyor Parameters			
	Radiant	fpm	Conveyor Speed in Zone #1 & #2
	Stabilization	fpm	Conveyor Speed in Zone #3
	Bake	fpm	Conveyor Speed in Zone #4

Table 3.1: ACRF Oven & Conveyor Parameters

The required cure parameters for any particular coating are outlined by the supplier of the coating based on the chemistry of the coating. The cure parameters are in general of the form of "**t** minutes at **x** metal temperature". Figure 3.3 illustrates a typical cure window for a coating. Although the paint supplier has deemed 20 minutes at 285°F as the optimal parameters, any point within the cure window provides an acceptable cure. Generally, each different coating chemistry has its own distinct cure window.

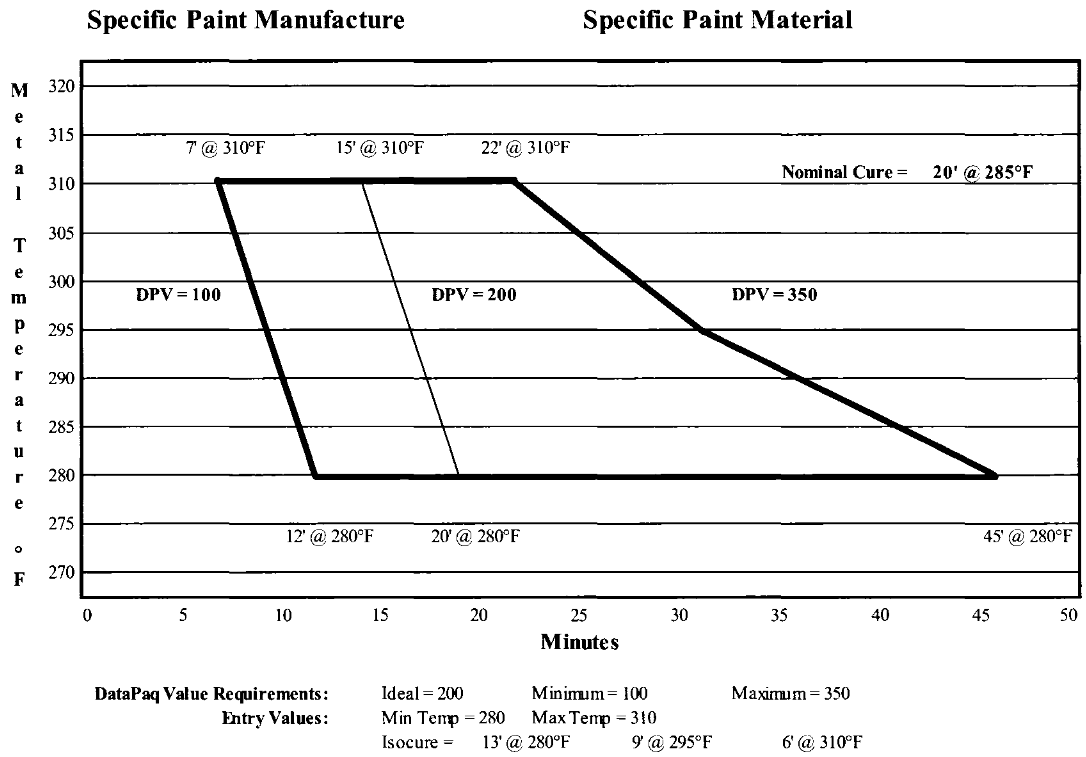


Figure 3.3: A Typical Paint Cure Window

The metal temperature of a vehicle, or effectively the cure of a coating, is measured using temperature thermocouples placed on the surface of a vehicle (see Figure 3.4). The thermocouples are connected to a transducer interface and the data is stored in a memory pack for downloading to a PC for analysis once the vehicle has exited the oven. The ACRF also uses a proprietary calculated value called the DataPaq Value (DPV) [Dat98] as a measure of cure.

$$DPV = f(MetalTemperature, Time)$$

Note that this thermocouple mechanism is not run on every vehicle. Generally it is setting up oven parameters or verification, and is run on several units per day. In an automotive assembly plant the same vehicles generally run through the same ovens continuously, every production day and the verification of the cure is required only minimally. At the ACRF, the oven conditions and vehicles are continuously changing due to the nature of the research and development activities being conducted. As can be seen from Table 3.1 there are a variety of parameters that must be adjusted, and the added parameter of different vehicle styles is also essential.

Figure 3.4: Thermocouple Probes

Currently, the ACRF process engineers utilize their past experience in an ad hoc manner, in combination with several test thermocouple runs to set the parameters in the oven to meet the cure specifications. Depending on many factors, including the geometry of the test specimen and complexity of the cure specifications, many trials could be required to meet and verify the cure requirements. The process engineers' past experience aspect of setting up the oven coincides with the CBR model of reasoning. However, the additional thermocouple runs that are often required are costly in terms of the resources required to accomplish this task. An automated technique with greater accuracy and repeatability than simple human intuition could prove valuable and cost effective. The paint curing oven domain discussed above is a dynamic environment with temporal entities that provides a good test bed for the CBR system that utilizes dynamic Bayesian networks in the retrieval and adaptation of cases. The problem case $\mathrm{P} = \{p_1, p_2, \cdots, p_{N_p}\}$ in CBR terminology would be an incomplete set of oven parameters (often referred to as evidence) from Table 3.1 and the solution case $S = \{s_1, s_2, \cdots, s_{n_a}\}$ would be a complete set of oven parameters that achieve the desired cure for the desired vehicle.

To model the persistent and dynamic attributes found in an industrial process, this thesis proposes using a dynamic Bayesian network representation. The domain expert is responsible for designing this network and incorporating both persistent and dynamic attributes. In an effort to simplify the representation, attributes can be grouped together to form a sort of super node. It should be noted that humans are typically poor at estimating probabilities [Pea00]. Therefore, the proposed approach relies mostly on the causal knowledge reflected by the structure of the network rather than on the actual distribution and probabilistic estimate. Causal dependencies seem to be much more intuitive than numeric probabilities in a DBN. Figure 3.5 illustrates a proposed DBN for the ACRF oven process as determined by the ACRF process engineers. The persistent attributes are duplicated in each time slice and enclosed at the top of the network within an ellipse and also have arcs (blue) from each time slice. In practice, since the persistent attributes

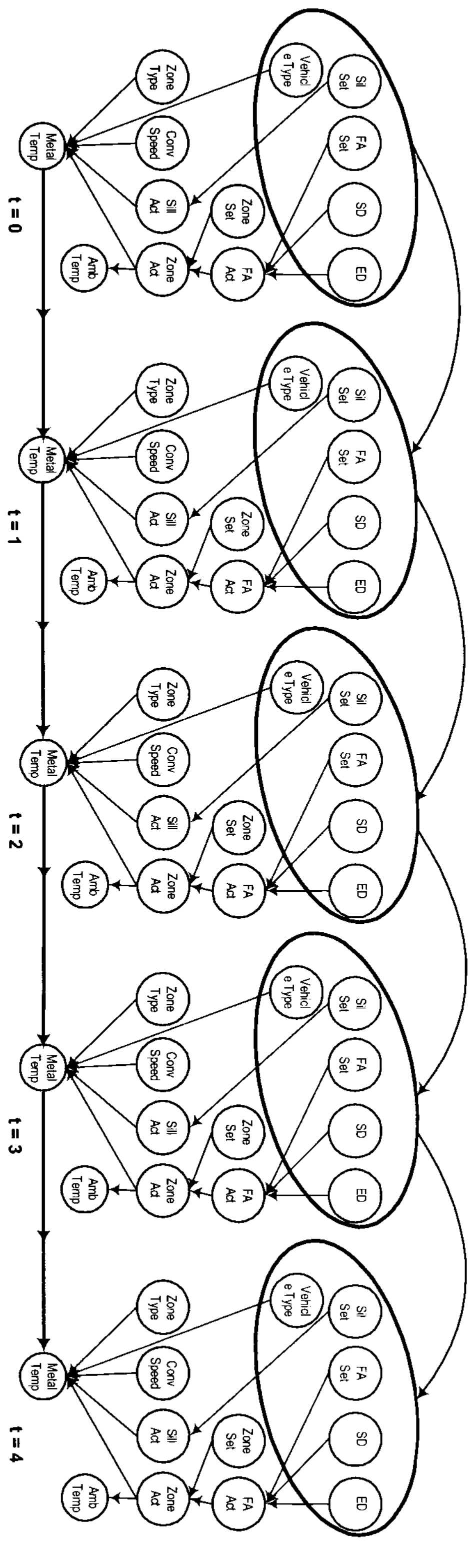


Figure 3.5: ACRF Oven Domain DBN

remain constant, there should be a single set of persistent attributes with arcs to each time slice. The representation used here is a compromise to provide clarity as arcs from each persistent attribute to every time slice would confuse the representation. All other nodes are dynamic attributes that could possibly change in each time slice. The arc from Metal Temp in each time slice represents the temporal link between time slices.

Chapter 4: CBR in an Industrial Process

A critical aspect of Bayesian network exact inference methods is the topology of the network, more precisely its connectivity [LD97]. By sectioning the network into sub networks and pruning irrelevant sections, it is hoped that a pseudo "divide and conquer" strategy can be used on the causal network for reasoning purposes. Although this strategy should prove to be more computationally efficient than working with the entire complex network, this work prefers to utilize the causal dependencies that these sectioning and pruning techniques afford. In [LD97] it is pointed out that, even by using schemes to section and prune causal networks, inference in the worst case remains NP-hard, however in some practical circumstances the problem becomes reasonable. In this work exact inference is not the preferred outcome, but a causally sectioned and pruned network is desired to improve retrieval and adaptation. A predicament that arises in real life circumstances (industrial processes) is the fact that there often exist both dynamic and persistent attributes in the same model and the treatment of these combinations requires different techniques. Persistent attributes tend to be of greater connectivity, as can be seen in the DBN in the examples provided in this work, and are handled separately in a statistical error calculation method.

4.1 CBR/DBN Retrieval/Adaptation Algorithm:

In this section, a CBR Retrieval/Adaptation algorithm utilizing a process DBN is presented and the main steps reviewed. Then a detailed example utilizing the domain outlined in Chapter 3.0 is worked out to provide concrete explanations of the ideas presented in this project. Most CBR systems separate the retrieval and the adaptation (if one exists) phases into two distinct and separate processes. In this project we combine these two steps into a single phase that utilizes causal knowledge in the form of a process DBN to produce a mechanism to retrieve portions of previous cases that are relevant to a particular section of the process DBN based on the causal independencies. The CBR/DBN Retrieval/Adaptation algorithm is presented in Figure 4.1 and explained below.

Step 1 of the algorithm takes a process description as input and produces a dynamic Bayesian network based on causal knowledge or relationships of the variables attributed to the process. For the purposes of this project the causal diagram is created by a domain expert. The attributes $c_i = \{\rho_1, \rho_2, \cdots, \rho_{N_p}, d_{11}, d_{12}, \cdots, d_{1l}, d_{21}, d_{22}, \cdots, d_{2l}, \cdots, d_{t1}, d_{t2}, \cdots, d_{N_d l}\}$ of the case-base are used to form the process DBN utilizing the causal relationships among the attributes of the case-base. Moreover, an expert's vision of the causal structure can be different from other experts' ideas and hence this step of the algorithm could be implemented using a learning algorithm for Bayesian networks [SSGM94]. However, these algorithms often have problematic concerns such as the inability to determine the direction of causal arcs, difficulties in handling continuous variables and weaknesses in dealing with latent variables. In summary, the output of step one is what we call a "process dynamic Bayesian network" which details the structure and causality of the process to be reasoned about over several time periods.

1. Causal Knowledge Acquisition: Obtain a dynamic Bayesian belief network utilizing expert knowledge of the process.

- At this time this network is created by the expert.

2. Situation Assessment: Obtain evidence or observations $P = \{p_1, p_2, \cdots, p_{N_p}\}$. This evidence is persistent and/or dynamic process parameters that would be known or required.

3. Applying Bayesian Network Concepts: Utilizing the evidence from step 2, analyze the DBN produced in step 1. Reduce and partition the DBN into sub-networks using the ideas outlined in Section 2.2:

a) Prune barren subnets.
b) First-order Markov assumption.
c) D-separation.

4. Case Retrieval – Dynamic Variables: For each sub-network found in step 3, utilizing the known evidence for that particular sub-network and the Multiply Sectioned Bayesian Network concept, query the case data for the nearest case obtaining unobserved attributes of the particular sub-network. Repeat step 4 for each sub-network obtained in step 3:

5. Case Retrieval – Persistent Variables: Calculate the sum of the average variation of combinations of persistent variables without evidence and the error of evidence variables from retrieved variables. Use the minimum variation plus error to determine the persistent variable case.

6. Case Adaptation: Combine the attributes for each sub-network found in step 4 and the values from step 5 to determine the completed retrieved/adapted solution $S = \{s_1, s_2, \cdots, s_{N_c}\}$.

Figure 4.1: CBR/DBN Retrieval/Adaptation Algorithm

The next step in the algorithm is common to most CBR systems. It is referred to as situation assessment and is simply the acquisition of some known data or observations. The data acquired is referred to as the problem case P $=\{p_1, p_2, \cdots, p_{N_p}\}$ from which a complete solution $S=\{s_1, s_2, \cdots, s_{N_c}\}$ is to be found. Techniques used in this step range from the utilization of known and given data attributes to making skillful assumptions and possibly pre-testing methods. On completion of the situation assessment step, a complete set of evidence or problem case P $=\{p_1, p_2, \cdots, p_{N_p}\}$ is established that will be used as the start of the reasoning process.

Thus far, in the previous two steps, a DBN that represents the process to be reasoned with and a set of evidence has been obtained. By placing the evidence found in step two against the DBN completed in step one and utilizing three well known causal and Bayesian network analysis techniques, the DBN can be reduced and partitioned into smaller causally independent sub-networks which can be further analyzed individually. Initially a reduction of the process DBN can be done utilizing the Barren node concept introduced in section 2.2.4. Barren nodes, often referred to as sink nodes, have no successors and hence no matter what value is assigned to a barren node, no other node can be affected. Hence, barren nodes can simply be eliminated from the DBN structure, thereby reducing the size of the entire causal structure. A Markov assumption can be used to partition the DBN by utilizing the time limited dependencies. By limiting the size of a subnet to a restricted number of previous time slices, the DBN can be partitioned. Finally, the concept of d-separation is used to partition the network into causally independent Subnets. By analyzing the evidence found in step two against the DBN from step one, the three d-separation conditions (linear, diverging or converging) can be resolved and hence independent portions of the DBN can be partitioned. This step may be performed by the domain expert, but automated determination of d-separation is possible. Hugin Expert Systems (Section 2.4) performed relatively well at d-separation and PNL (Section 2.4) showed some promise at automating d-separation inference. In summary the output from step three is a set of causally independent subnets.

In step 4, the evidence from step 2 and the causally independent subnets from step three are used to determine the dynamic attributes for the solution case. Available evidence from a particular subnet and any evidence for persistent attributes are used to query the case-base for the best case(s). Additionally, by utilizing the MSBN concept introduced in Section 2.2.5, certain evidence from the previous adjacent subnet is also used in the query of the case-base. Each subnet will have in theory, its own closest case for those set of dynamic attributes in the subnet, which is justified by the independence of the subnet given the evidence. Currently, the Euclidean distance metric with a normalizing function for different attributes is used to find the closest case for each subnet. This procedure is repeated for each subnet, thereby determining the dynamic variables for the entire network. The best cases from each subnet will also be used in step 5 to determine the persistent variables.

The fact that most real life situations involve both persistent and dynamic variables is handled in step 5. As can be envisioned from the results in step 4 there could be a different set of values for persistent variables retrieved for each individual subnet. This introduces a dilemma as to which values for the persistent variables to use in the combined case. The proposed solution to the problem of selecting values for persistent variables is based on imposing an additional penalty in the Euclidean distance calculations on each combined case whose constituent sub-cases assign conflicting values

// $\ddot{e}_n$ is called the error from evidence error metric
// $\hat{e}_n$ is called variance from average error metric
// ε_n is called the total error metric for case ***n***

Let $c_i = \{\rho_1, \rho_2, \cdots, \rho_{N_\rho}, d_{11}, d_{12}, \cdots, d_{1l}, d_{21}, d_{22}, \cdots, d_{2l}, \cdots, d_{t1}, d_{t2}, \cdots, d_{hl}\}$ or $c_i = \{a_{i1}, a_{i2}, \ldots, a_{iN_a}\}$ be a case in the case base C

where N_c be the total number of cases.
N_a be the number of attributes in a case.
ρ_j are persistent attributes in a case.
N_ρ be the number of persistent attributes in a case.
d_{kl} are dynamic attributes in a case.
N_d be the number of dynamic attributes in a case.
l is the number of time slices in a case.
a_i are attributes in a case.

Let $P = \{p_1, p_2, \cdots, p_{N_p}\}$ be the problem case (evidence)

where N_p be the number of attributes in the problem case.

Let $R_i = \{r_1, r_{22}, \cdots, r_m\}$ be a best case set from sub net i, determined by taking the best m (Combination Level) cases from sub-net i.

Let $Q_i = \{q_1, q_2, \cdots, q_{N_s}\}$ be a combination set determined by taking, $q_1 \in R_1, q_2 \in R_2, \cdots, q_{N_s} \in R_{N_s}$

where N_s is the number of sub nets.

Let $S = \{Q_1, Q_2, \cdots, Q_{N_{comb}}\}$ be the set of all possible combinations of Q_1

where N_{comb} is the total number of possible combinations

// Determine the distance from evidence error metric
for all **Cases** c_i $(i = 1 \text{ to } N_c)$
  if $(c_i \in Q)$ then
    for all **Attributes** in C_i $(j = 1 \text{ to } N_a)$
      $\ddot{e}_i = \ddot{e}_i + \text{abs}(a_{ij} - p_j)$

// Determine the variance from average error metric
for all **Combinations** Q_i $(i = 1 \text{ to } N_{comb})$
  for all **Persistent Attributes** ρ_j in Q_i $(j = 1 \text{ to } N_\rho)$
    for all **Sub nets** $k = 1 \text{ to } N_s$
      $\text{Sum}(\rho_j) = \text{Sum}(\rho_j) + \rho_j$ from sub net k
    $\text{Average}(\rho_j) = \text{Sum}(\rho_j) / N_s$
    $\hat{e}_{ij} = \text{abs}(\rho_j - \text{Average}(\rho_j))$
  for all **Cases** c_j in Q_i $(j = 1 \text{ to } N_\rho)$
    $\varepsilon_{ij} = \hat{e}_{ij} + \ddot{e}_j$

// Select the best case in which the persistent values come from
Select **Case** c_j with $\text{Minimum}(\varepsilon_{ij})$

Figure 4.2: Determination of Persistent Variables

to persistent attributes. Basically, the best m (combination level) sub-cases are selected from each sub-network and then all possible combinations of these sub-cases are enumerated. In each combination, the mean and variance from the mean of each combination of variables are calculated and added to the error metric. The combined case with the smallest sum of variance and error is selected for the persistent variable representation. The algorithm is detailed in Figure 4.2.

In the final step, persistent values found in step five and the dynamic variables found in step four are combined to form the complete solution case $S = \{s_1, s_2, \cdots, s_{N_c}\}$. These variables can be use in the process to solve the original problem.

4.2 Detailed CBR/DBN Retrieval/Adaptation Example:

This example (referred to as the ACRF topcoat example 1) uses the oven domain reviewed in Chapter 3.0, with a real life set of data (the case-base), consisting of 137 separate transducer runs (individual cases) through the oven to produce a case-base.

Starting with step one of the CBR/DBN Retrieval/Adaptation algorithm, Figure 3.5 illustrates an experts view of the causal relationships involved in the oven in the form of a dynamic Bayesian network.

In the situation assessment step all available evidence from given information, observations and assumptions are flushed out as illustrated in Table 4.1. This table was generated from the initial problem situation; a vehicle type 0 (SUV) where a temperature ramp of 25°F per minute is required, the oven zone types are fixed and some additional realistic assumptions were made.

Evidence from Table 4.1 is entered into the DBN as illustrated by the shaded nodes in Figure 4.4. Next the Bayesian Network analysis tools from Section 2.2 can be used to reduce and partition the DBN. First the Ambient Temperature (AmbTemp) node at time t = 0 is eliminated because it is a barren node as illustrated by the cross-hatched node. By using the d-separation criteria and independence assumptions from [Pea00], the DBN can now be separated into four separate subnets (t = 0, t = 1 and t = 2, t = 3, and t = 4) because the metal temperature evidence is known for these subnets as illustrated by the vertical lines in Figure 4.3.

		Variable	Evidence	Reason
Static		Vehicle Type	0 (SUV)	Observations
Dynamic	t = 0	Zone Type	0 (not in oven)	Entering oven
		Ambient Temp	75°F	Assumption outside oven
		Metal Temp	75°F	Assumption outside oven
	t = 1	Zone Type	1 (Radiant)	Fixed Oven
		Conveyor Speed	10	Z1 = Z2 physical limitation, Ramp
	t = 2	Zone Type	1 (Radiant)	Fixed Oven
		Metal Temp	250°F	Ramp = 25°/minute
		Conveyor Speed	10	Z1 = Z2 physical limitation, Ramp
	t = 3	Zone Type	2 (Convection)	Fixed Oven
		Metal Temp	285°F	Cure Window
	t = 4	Zone Type	2 (Convection)	Fixed Oven
		Metal Temp	285°F	Cure Window
		Conveyor Speed	20	Cure Window

Table 4.1: Evidence for the ACRF Oven Domain

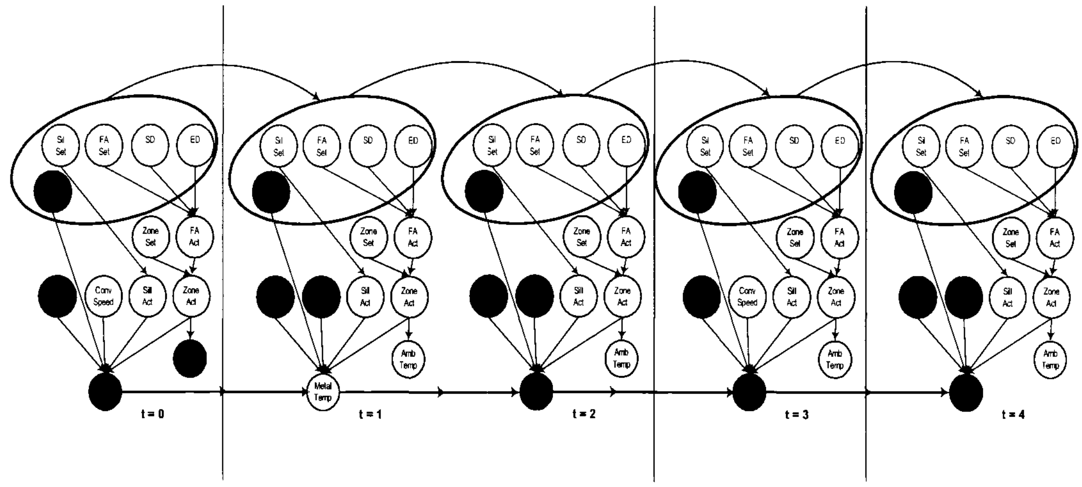


Figure 4.3: DBN Reduced and Partitioned into Subnets

The next step involves taking each subnet, the persistent evidence, the evidence known for that particular subnet and querying the case-base, to retrieve the best case for that particular subnet. For this example, utilizing the shaded nodes (evidence) in the fourth subnet (VehicleType, ZoneType @ t = 4, ConvSpeed @ t = 4, MetalTemp @ t = 4 and MetalTemp @ t = 3) a query is made of the case-base to determine the best case with respect to evidence from subnet 4. Note that MetalTemp @ t = 3 is included in the query by way of the MSBN concept. This query determines that the best case for Subnet 4 is case 30. Hence, the values from case 30 for MetalTemp @ t = 4 (284), ZoneType @ t = 4 (2), ConvSpeed @ t = 4 (18.9), SillAct @ t = 4 (360), ZoneAct @ t = 4 (300), ZoneSet @ t = 4 (300), FAAct @ t = 4 (301) and AmbTemp @ t = 4 (295) will be used for solution values for subnet 4. The essential values here would be the zone set point is 300 and the conveyor speed is 18.9. This procedure is repeated for each subnet, thereby obtaining a best case for each subnet and hence values for all dynamic variables in the dynamic Bayesian network model. Note that for this example, the best case for subnet 1 is case 98,

Test: ACRF Oven Domain **Comments:** Common DBN example m = 10

Attribute	Evidence	Time	Barren	CBR/DBN Method SubNet1 1	2	Case	Value	Δ Evidence	Convential CBR w/ Euclidean Distance Case	Value	Δ Evidence
						Persistent					
RunID	None	All	No	N/A	N/A	120	121	--	65	66	--
RunDate	None	All	No	N/A	N/A	120	11/18/2003	--	65	10/4/2002	--
VehicleType	0	All	No	N/A	N/A	120	0	[redacted]	65	7	7
SupplyDamper	None	All	No	N/A	N/A	120	30	--	65	30	--
ExhaustDamper	None	All	No	N/A	N/A	120	75	--	65	75	--
FASetPoint	None	All	No	N/A	N/A	120	290	--	65	300	--
SillSetPoint	None	All	No	N/A	N/A	120	300	--	65	380	--
						Zone 0					
ZoneType	0	t = 0	No	0	--	98	0	0	65	0	0
ConveyorSpeed	None	t = 0	No	0	--	98	12.6	--	65	11.6	--
FAActual	None	t = 0	No	0	--	98	180	--	65	290	--
SillActual	None	t = 0	No	0	--	98	330	--	65	450	--
ZoneSet	None	t = 0	No	0	--	98	75	--	65	75	--
ZoneActual	None	t = 0	No	0	--	98	75	--	65	75	--
AmbientTemp	75	t = 0	Yes	0	--	98	78	[redacted]	65	80	5
MetalTemp	75	t = 0	No	0	1	98	75	[redacted]	65	78	3
						Zone 1					
ZoneType	1	t = 1	No	1	--	112	1	0	65	1	0
ConveyorSpeed	10	t = 1	No	1	--	112	9.8	[redacted]	65	10.7	0.7
FAActual	None	t = 1	No	1	--	112	250	--	65	290	--
SillActual	None	t = 1	No	1	--	112	300	--	65	450	--
ZoneSet	None	t = 1	No	1	--	112	265	--	65	350	--
ZoneActual	None	t = 1	No	1	--	112	266	--	65	350	--
AmbientTemp	None	t = 1	No	1	--	112	255	--	65	287	--
MetalTemp	None	t = 1	No	1	--	112	167	--	65	193	--
						Zone 2					
ZoneType	1	t = 2	No	1	--	112	1	0	65	1	0
ConveyorSpeed	10	t = 2	No	1	--	112	9.8	[redacted]	65	10.7	0.7
FAActual	None	t = 2	No	1	--	112	250	--	65	290	--
SillActual	None	t = 2	No	1	--	112	300	--	65	450	--
ZoneSet	None	t = 2	No	1	--	112	295	--	65	340	--
ZoneActual	None	t = 2	No	1	--	112	295	--	65	340	--
AmbientTemp	None	t = 2	No	1	--	112	289	--	65	302	--
MetalTemp	250	t = 2	No	1	2	112	254	[redacted]	65	255	5
						Zone 3					
ZoneType	2	t = 3	No	2	--	112	2	0	65	2	0
ConveyorSpeed	None	t = 3	No	2	--	112	36.1	--	65	16.4	--
FAActual	None	t = 3	No	2	--	112	250	--	65	290	--
SillActual	None	t = 3	No	2	--	112	300	--	65	450	--
ZoneSet	None	t = 3	No	2	--	112	295	--	65	305	--
ZoneActual	None	t = 3	No	2	--	112	297	--	65	308	--
AmbientTemp	None	t = 3	No	2	--	112	292	--	65	300	--
MetalTemp	285	t = 3	No	2	3	112	283	[redacted]	65	281	4
						Zone 4					
ZoneType	2	t = 4	No	3	--	30	2	0	65	2	0
ConveyorSpeed	20	t = 4	No	3	--	30	18.9	1.1	65	19	[redacted]
FAActual	None	t = 4	No	3	--	30	300	--	65	290	--
SillActual	None	t = 4	No	3	--	30	360	--	65	450	--
ZoneSet	None	t = 4	No	3	--	30	300	--	65	305	--
ZoneActual	None	t = 4	No	3	--	30	301	--	65	305	--
AmbientTemp	None	t = 4	No	3	--	30	295	--	65	305	--
MetalTemp	285	t = 4	No	3	--	30	284	[redacted]	65	291	6
								8			1
Dynamic Timing:							0.00000 sec			0.01001 sec	
Persistent Timing:							0.20029 sec				

Table 4.2: ACRF Topcoat Oven Example 1 – Complete Results

for subnet 2 is 112 and for subnet 3 is 112. Table 4.2 contains the complete results from this example problem.

The next concern is the determination of the persistent parameters, of which we could possibly have several sets (as many sets as there are subnets). There will be a set of persistent parameters for each subnet so it can be determined which case the persistent parameters should be taken from. Utilizing the algorithm in Figure 4.2, with combination level $m = 10$, case 27 has the minimum error metric and hence the persistent values for SillSet = 280, FASet = 250, SD = 44, ED = 0 and VehicleType = 0 are used in the final solution.

The final step is to combine all the values found in step 4 and step 5 to form a complete solution case (see Table 4.2).

For comparison purposes throughout this work, a conventional CBR strategy that retrieves a single case and employs the same Euclidean distance metric as the CBR/DBN Retrieval/Adaptation algorithm will be utilized. The last three columns of Table 4.2 represent the results of the conventional CBR strategy; note the single case that is retrieved and the values for that single case. Similar columns exist for the CBR/DBN Retrieval/Adaptation algorithm; note the different case numbers for each subnet. Of particular interest are the columns labeled "Δ evidence" which calculate the difference of the retrieved value from the evidence as determined in step two of the CBR/DBN Retrieval/Adaptation algorithm. As an evaluation technique, the two "Δ evidence" columns can be compared with the smaller of the two "Δ evidence" values which indicates the superior of these two values. Note that the superior value for each attribute with evidence is shaded. Under this evaluation technique the CBR/DBN Retrieval/Adaptation algorithm does quite well as indicated by the shaded cells indicating the better value versus evidence, although at an efficiency penalty of 0.20029 seconds of additional computation time. This method of evaluation will be used for further examples presented in Chapter 5.

As further validation of the CBR/DBN Retrieval/Adaptation approach, a thermocouple run was processed through the ACRF oven with the solution parameters from this example for verification. This method of substantiation is not preferred because of the considerable cost, but currently it is the only method available. Figure 4.4 represents the results of the thermocouple run in chart form. This type of chart layout with Temperature along the "Y" axis and Time along the "X" axis is typically used in evaluating thermocouple runs. The two lines on the chart show the temperature as the vehicle passes through the oven, one is metal temperature whereas the other is ambient temperature. The data from Table 4.3 is also used in evaluating cure properties of the oven. From the table it can determined that the metal temperature was at or above 285°F for 21.83 minutes (the initial problem stated 285°F for 20 minutes) and the ramp rate was 27°F/minute (again the initial problem stated 25°F/minute) . These results are considered extremely good, especially considering that the current method of determining the set points is based solely on the experience of a domain expert (or process engineer) as well as many trial and error attempts.

Probe	**Time Above 285.0°F (mm:ss.t)**	**Time To Reach 285.0°F (mm:ss.t)**	**Slope (°F/sec)**
#1 (°F) Ambient	26:40.0	09:00.0	2.14
#2 (°F) Metal	21:50.0	13:05.0	0.45

Table 4.3: Thermocouple Results

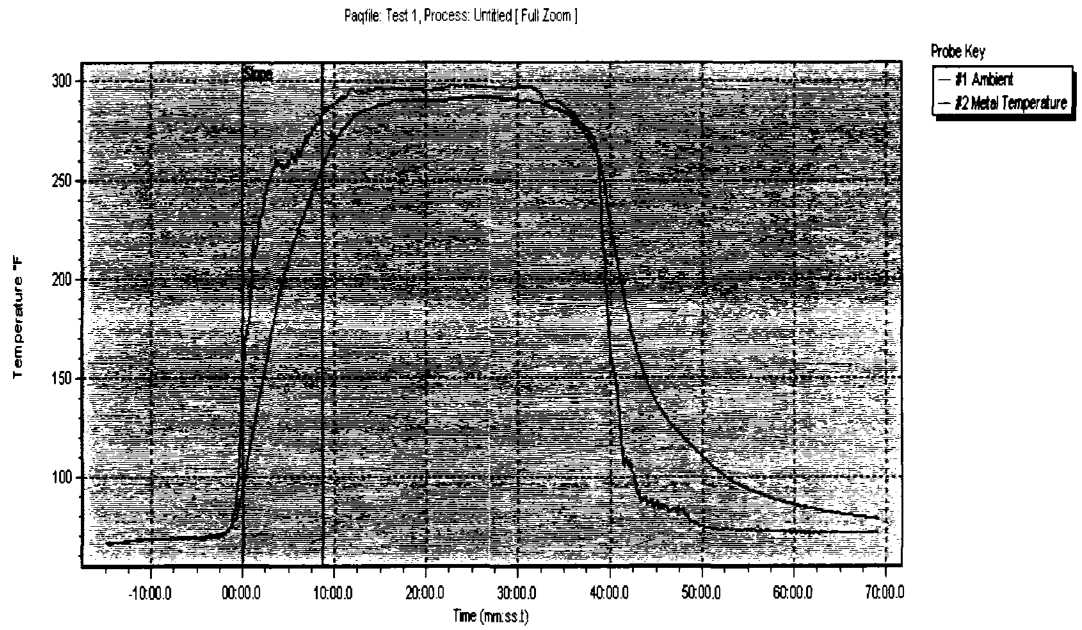


Figure 4.4: Temperature Chart

4.3 Complexity of the CBR/DBN Retrieval/Adaptation Algorithm:

The application (see Appendix A) developed to implement sections of the CBR/DBN Retrieval Adaptation algorithm initially inputs the entire case-base into RAM. Although this type of input practice could possibly use a large amount of memory, depending on the number of cases and the number of attributes in each case, the difficulties in obtaining a great number of cases in an industrial environment could justify this action. The bottle neck of the algorithm as implement is Step 5, where the Combination Level has a significant effect on complexity. The yet to be implemented steps 1 and 3 also contain computationally complex algorithmics, but could be assisted by some domain expert intervention, such as defining some obvious directed edges. Each step in the CBR/DBN Retrieval/Adaptation algorithm will now be analyzed for time complexity, that is the worst case computation time.

4.3.1 Step 1: Causal Knowledge Acquisition

Currently, the causal knowledge acquisition step is being completed by a domain expert in which case exact time complexity is not scientifically determinable. However, it is possible in theory to determine a BN or DBN from the data or in this situation the case-base. There are several researched Bayesian network learning algorithms including Recursive Autonomy Identification (RAI) [YL05], Inductive Causality (IC) [Pea00], Three Phase Dependency Analysis (TPDA) [CBL97] and the PC Algorithm [SSGM94]. In Tetrad, the Build module uses the "PC Algorithm" to create a Bayesian network from data. This algorithm is repeated in Figure 4.5 and analyzed next. In step A, a complete undirected graph is formed by taking each vertex in the set of vertices and connecting it to every other vertex. This takes in worst case $O(N_a^{\,2})$, where N_a is the number of attributes or vertices in the sample data. In step B, adjacencies are eliminated by finding conditional independence relations in the data in which the worst case is $O(N_a^{\,3})$. Implied in this step is the determination of independence which is generally done by finding the covariance's of the input data.

$$\text{Covariance} = \sigma_{ij} = \rho\sigma_i\sigma_j = E[(X_i - \mu_i)(X_j - \mu_j)]$$

The time taken to complete the covariance operation is $O(N_c^{\,2})$, where N_c is the number of sets of data (cases) to be inputted. Step C and D parses though the set of adjacencies found in step A so the worst case complexity is $O(N_a^{\,2})$ for each step. Reviewing the entire algorithm, worst case appears to be $O(N_a^{\,3})$. By defining some obvious edges a decrease in complexity could possibly be realized in the Tetrad program. In general, learning a Bayesian network from data is not a trivial task, most researchers considers it an NP-hard problem [CHM04] although [CBL97] appears to have an algorithm with a time complexity of $O(N_a^{\,2})$. In section 2.6 it was also noted that from experience the Build module in Tetrad often has problems with determining the direction of the arcs in real world problems.

Let Adjacencies(C,a) be the set of vertices adjacent to a in a graph C. (In the algorithm, the graph C is continually updated, so Adjacencies(C,a) is constantly changing as the algorithm progresses

A. Form the complete undirected graph on the vertex set V.
B. n = 0.
repeat
repeat
select an ordered pair of variables x and y that are adjacent in C such that Adjacencies(C,x)\{y} has cardinality greater than or equal to n, and a subset S of Adjacencies(C,x)\{y} of cardinality n, and if x and y are independent given S delete edge x – y from C and record S in Sepset(x,y) and Sepset(y,x)
until all ordered pairs of adjacent variables x and y such that Adjacencies(C,x)\{y} has cardinality greater than or equal to n and all subsets S of Adjacencies(C,x)\{y} of cardinality n have been tested for independence;
n = n + 1
until for each ordered pair of adjacent vertices x,y, Adjacencies(C,x)\{y} is of cardinality less than n.
C. *For each* triple of vertices x.y.z such that the pair x,y and the pair y,z are each adjacent in but the pair x, z are not adjacent in C, orient x – y – z are x →y→z if and only if y is not in Sepset(x,z).
D. *repeat*
1. if x→y, y and z are adjacent, x and z are not adjacent, and there is no arrowhead at y, then orient y – z as y→z.
2. if there is a directed path from x to y, and an edge between x and y, then orient x – y as x→y.
until no more edges can be oriented.

Figure 4.5: The PC Algorithm, repeated from [SSGM94]

4.3.2 Step 2: Situation Assessment

Since situation assessment in step 2 is a manual operation, it is also difficult to determine precisely. The time taken is highly dependent on the domain, knowledge of the domain and availability of data for the domain. It should be noted that situation assessment is a common process for most CBR systems and hence this time would be included in most CBR systems.

4.3.3 Step 3: Applying Bayesian Network Concepts

In [GVP90] three algorithms are presented for the determination of all independencies implied by a Bayesian network. From these algorithms the time required for the best algorithm (#1) to determine the all independencies is $O(|E|)$, where E is the number of edges in a network.

4.3.4 Step 4: Case Retrieval – Dynamic Variables

Figure 4.6 presents an analysis of step 4 of the CBR/DBN Retrieval/Adaptation algorithm. From line (1) the time to sort the cases is $N_c \log N_c$, where N_c is the number of cases if a good sort algorithm (e.g. Merge Sort) is used [BB96] [LRSC01]. It should be noted that the algorithm used in the application is not quite that efficient. The dominant terms will be the number of cases N_c and the number of subnets N_s which is what could be expected when the case-base is searched once for each subnet. A slight increase in efficiency could be obtained by searching the case-base only once and finding the best case in each subnet in this single search. This introduces some slight programmatic concerns, but these could be handled in a relevantly straightforward manner. As can be seen from Figure 4.6 the Dynamic attribute determination portion of the CBR/DBN Retrieval/Adaptation algorithm is linear in the number of cases and number of subnets. From the example presented in section 4.1 the time to determine the dynamic attributes is close to zero for a reasonable number of cases. The zero time seems to come from the inability to accurately measure computation time at such low time intervals.

N_c = Nnumber of cases in the case - base.
N_s = Numbder of subnets.
N_p = Number of persistent attributes.
N_d = Number of dynamic attributes.

$$N_c\left(\underbrace{(N_s * N_p)}_{\textit{Persistent}\text{ Attribute Distance}} + \underbrace{N_s + (N_s * N_d)}_{\textit{Dynamic}\text{ Attribute Distance}}\right) + \underbrace{(N_c * N_c) + (N_c * (N_c - 1) * N_s)}_{\textit{Rank}\text{ and Sort Cases}}$$

$$= N_c N_s (N_p + N_d + 1) + \text{Time to Sort Cases}$$

$$= N_c N_s (N_p + N_d + 1) + N_c \log N_c$$

(1)

From [BB96] [LRSC01] assuming the best sort algorithm is used

$$= O(N_c N_s) \quad \text{since } N_c >> N_p \ \& \ N_c >> N_d$$

Figure 4.6: Analysis of Step 4 (Dynamic Attribute Determination)

4.3.5 Step 5: Case Retrieval – Persistent Variables

The number of best cases from each subnet or combination level *m* to be used to determine the best persistent case can have a major effect on the efficiency of the CBR/DBN Retrieval/Adaptation algorithm. The determination of the persistent attributes as presented in this work is a major deviation from typical Euclidean distance retrieval type processes and is an added complexity that should be evaluated. When looking at the complexity of step 5 (Figure 4.7) things are much worse as expected and step 5 becomes the performance bottle-neck. Determining all combinations of m persistent attributes from each subnet becomes the efficiency quandary. From line (2) in Figure 4.7 it can be seen that the complexity of the persistent attribute determination is $O\left(m^{N_p}\right)$ or exponential in N_p. The number of persistent attributes N_p is generally fixed for each domain and this value is generally small, less than the total number of cases or at least not infinite. This leaves the determination of m which can be any value from one to the total number of cases in the case-base as the significant process in the algorithm and hence care must be taken when choosing m. A experimental time analysis of m that leads to a reasonable balance between accuracy and time is presented in the next chapter. The space complexity could also become unbearable if the results of all combinations were saved, so only the best case to-date information is saved.

$$m\left(N_s N_p + N_d + N_s N_p\right) * m\left(N_s N_p + N_d + N_s N_p\right) * \ldots \quad N_p \text{ times}$$

$$= m^{N_p} \text{ or } O\left(m^{N_p}\right) \qquad \text{Exponential time} \qquad (2)$$

Figure 4.7: Analysis of Step 5 (Persistent Attribute Determination)

4.3.6 Step 6: Case Adaptation

The time complexity for this adaptation step is negligible as it is somewhat of a formality to combine the dynamic and persistent attributes into a single case.

4.4 Euclidean Distance Metric:

This project exclusively uses a Euclidean distance metric [Ray99] to evaluate the closeness of a solution. The Euclidean distance metric is common and relatively simple in that the distance between two attributes of the same type is simply the difference between these two attributes.

A problem arises when there is an attempt to find the distance between cases with varying types of attributes, for example discrete variables, continuous variables with varying ranges, Boolean variables or dates. The distance between a discrete attribute with range 0 to 5 and a continuous attribute with range 0 to 500 can be great and perhaps screen the influence of the discrete variable. To alleviate this problem, all attributes are normalized to 100 (with Boolean being either 0 or 50) and a weight (0 to 10) is assigned to the attribute. The weight is assigned by the domain expert as a gauge of the influence of the attribute on the process. The Euclidean distance metric is explained here for completeness, although it has no significance on the actual CBR/DBN Retrieval/Adaptation algorithm and in theory any distance metric or closeness criteria can be used.

Chapter 5: Further Evaluation and Experimental Results

In this section, the CBR/DBN Retrieval/Adaptation algorithm will be evaluated further by providing several additional examples from the ACRF oven test domain and some addition dynamic environments found in research literature. As illustrated in these examples, the CBR/DBN Retrieval/Adaptation algorithm is a feasible option for the retrieval and adaptation components in a case-based reasoning environment. It should be noted that all testing for this thesis was accomplished on a Dell Optiplex GX720 with a Pentium 4 2.8 GHz processor and 512 MB RAM.

5.1 Example 2 - ACRF Oven Domain, Powder Cure:

To solidify the ideas presented in Section 4.1, this section presents another real life example from the ACRF oven domain. The realistic example presented in section 4.1 was essentially a topcoat (basecoat and clearcoat) cure specification. In this example a powder primer cure is presented to indicate the robustness of the approach. Generally powder primer and topcoat cure specifications are at opposite ends of the spectrum as far a cure windows go.

The statement of the problem is as follows: given a paint cure window similar to Figure 3.3 that indicates that the optimal cure is 20 minutes @ 340°F for an SUV (discrete type = 1). This is a practical problem stated in a realistic manner. To add robustness to this example, several additional observations will be presented. It is required that the exhaust damper be closed (ED = 0%) to keep positive pressure at the oven entrance and exit. Additionally, a minimum time is required in the radiant zones (maximum speed in the radiant zones is 12.6 ft/min). After the first radiant zone the metal temperature should be 175°F and 250°F after the second radiant zone.

Since the domain is the same as section 4.1, the DBN presented in Figure 4.3 will be used as the initial DBN model. Step 2 of the CBR/DBN Retrieval/Adaptation algorithm is summarized in Table 5.1. Figure 5.1 illustrates the results of step 3 of the CBR/DBN Retrieval/Adaptation algorithm with the subnets separated by the vertical lines (five subnets for this example). Table 5.2 presents the results of the entire CBR/DBN Retrieval/Adaptation algorithm in a similar manner as in the previous example. Once again the algorithm does quite well as can be seen from the shaded cells indicating that a particular algorithm is improved over the other for that piece of evidence.

		Variable	Evidence	Reason
Persistent		Exhaust Damper	0	Observations
		Vehicle Type	0 (SUV)	Observations
Dynamic	t = 0	Zone Type	0 (not in oven)	Entering oven
		Ambient Temp	75°F	Assumption outside oven
		Metal Temp	75°F	Assumption outside oven
	t = 1	Zone Type	1 (Radiant)	Fixed Oven
		Metal Temp	175°F	Observations
		Conveyor Speed	12.6	Observations - Maximum Speed
	t = 2	Zone Type	1 (Radiant)	Fixed Oven
		Metal Temp	275°F	Observations
		Conveyor Speed	12.6	Observations - Maximum Speed
	t = 3	Zone Type	2 (Convection)	Fixed Oven
		Metal Temp	340°F	Cure Window
	t = 4	Zone Type	2 (Convection)	Fixed Oven
		Metal Temp	340	Cure Window
		Conveyor Speed	20	Cure Window

Table 5.1: Example 2 - Evidence (Step 2)

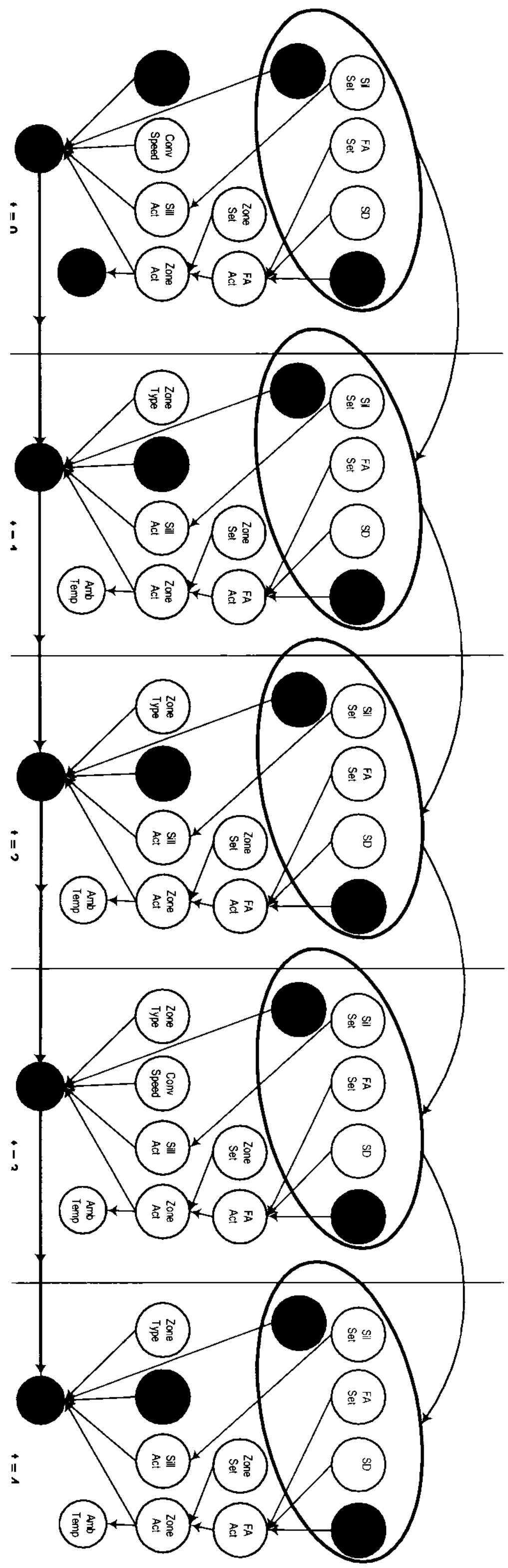


Figure 5.1: Example 2 - DBN reduced and partitioned into Subnets (Step 3)

Test: ACRF Domain **Comments:** Powder Coat Example 02 m = 10

		CBR/DBN Method							CBR w/ NN Distance		
				SubNet1				Δ			Δ
Attribute	**Evidence**	**Time**	**Barren**	**1**	**2**	**Case**	**Value**	**Evidence**	**Case**	**Value**	**Evidence**
Persistent											
RunID	None	All	No	N/A	N/A	31	32	--	20	21	--
RunDate	None	All	No	N/A	N/A	31	12/11/2001	--	20	10/4/2002	--
VehicleType	0	All	No	N/A	N/A	31	0	0	20	0	0
SupplyDamper	None	All	No	N/A	N/A	31	44	--	20	44	--
ExhaustDamper	0	All	No	N/A	N/A	31	0	0	20	0	0
FASetPoint	None	All	No	N/A	N/A	31	300	--	20	310	--
SillSetPoint	None	All	No	N/A	N/A	31	360	--	20	290	--
Zone 0											
ZoneType	0	t = 0	No	0	--	27	0	0	20	0	0
ConveyorSpeed	None	t = 0	No	0	--	27	12.6	--	20	16.4	--
FAActual	None	t = 0	No	0	--	27	300	--	20	290	--
SillActual	None	t = 0	No	0	--	27	243	--	20	450	--
ZoneSet	None	t = 0	No	0	--	27	75	--	20	75	--
ZoneActual	None	t = 0	No	0	--	27	75	--	20	75	--
AmbientTemp	75	t = 0	Yes	0	--	27	79	4	20	80	5
MetalTemp	75	t = 0	No	0	1	27	82	7	20	78	3
Zone 1											
ZoneType	1	t = 1	No	1	--	28	1	0	20	1	0
ConveyorSpeed	12.6	t = 1	No	1	--	28	13.4	0.8	20	11.5	1.1
FAActual	None	t = 1	No	1	--	28	300	--	20	290	--
SillActual	None	t = 1	No	1	--	28	343	--	20	450	--
ZoneSet	None	t = 1	No	1	--	28	330	--	20	350	--
ZoneActual	None	t = 1	No	1	--	28	329	--	20	350	--
AmbientTemp	None	t = 1	No	1	--	28	240	--	20	287	--
MetalTemp	175	t = 1	No	1	2	28	177	2	20	193	18
Zone 2											
ZoneType	1	t = 2	No	2	--	28	1	0	20	1	0
ConveyorSpeed	12.6	t = 2	No	2	--	28	13.4	0.8	20	11.5	1.1
FAActual	None	t = 2	No	2	--	28	300	--	20	290	--
SillActual	None	t = 2	No	2	--	28	380	--	20	450	--
ZoneSet	None	t = 2	No	2	--	28	420	--	20	340	--
ZoneActual	None	t = 2	No	2	--	28	420	--	20	340	--
AmbientTemp	None	t = 2	No	2	--	28	312	--	20	302	--
MetalTemp	275	t = 2	No	2	3	28	281	6	20	255	20
Zone 3											
ZoneType	2	t = 3	No	3	--	94	2	0	20	2	0
ConveyorSpeed	None	t = 3	No	3	--	94	56.4	--	20	49.2	--
FAActual	None	t = 3	No	3	--	94	290	--	20	290	--
SillActual	None	t = 3	No	3	--	94	450	--	20	450	--
ZoneSet	None	t = 3	No	3	--	94	305	--	20	305	--
ZoneActual	None	t = 3	No	3	--	94	308	--	20	308	--
AmbientTemp	None	t = 3	No	3	--	94	300	--	20	300	--
MetalTemp	340	t = 3	No	3	4	94	281	**59**	20	281	59
Zone 4											
ZoneType	2	t = 4	No	4	--	1	2	0	20	2	0
ConveyorSpeed	20	t = 4	No	4	--	1	22.7	2.7	20	19.7	0.3
FAActual	None	t = 4	No	4	--	1	290	--	20	290	--
SillActual	None	t = 4	No	4	--	1	450	--	20	450	--
ZoneSet	None	t = 4	No	4	--	1	305	--	20	305	--
ZoneActual	None	t = 4	No	4	--	1	305	--	20	305	--
AmbientTemp	None	t = 4	No	4	--	1	305	--	20	305	--
MetalTemp	340	t = 4	No	4	--	1	291	49	20	291	49
								5			2
	Dynamic Timing:						0.00000 sec			0.00000 sec	
	Persistent Timing:						0.25036 sec				

Table 5.2: Example 2 - Results

5.2 Example 3 – ACRF Oven Domain w/ Tetrad Generated Data:

In this example, the data for the case-base was generated from the structure of the dynamic Bayesian network using the Tetrad program introduced earlier. The ACRF oven domain DBN will be used as in the previous two examples. The DBN in Tetrad format is illustrated in Appendix C. By generating data using Tetrad, the causal relationships inherent in the case-base are maintained and hence the rationale behind this project is preserved. The data in the case-base itself does not make sense in the terms of the ACRF oven domain but the causal nature of the data is what is of interest. The evidence for this example was selected arbitrarily with the only restrictions being keeping the evidence between the minimum and maximum values for each attribute. The evidence is presented in Table 5.3 and the results are shown in Table 5.4.

		Variable	Evidence
Static		Vehicle Type	16
Dynamic	t = 0	Zone Type	15
		Conveyor Speed	20
		Ambient Temp	1
		Metal Temp	6
	t = 1	Zone Type	17
		Conveyor Speed	17
	t = 2	Zone Type	17
		Conveyor Speed	17
		Metal Temp	11
	t = 3	Zone Type	18
		Metal Temp	16
	t = 4	Zone Type	20
		Metal Temp	17
		Conveyor Speed	24

Table 5.3: ACRF Oven w/ Tetrad Data Example 3 - Evidence (Step 2)

Test: ACRF Oven **Comments:** Tetrad Data Ex03 m = 10

		CBR/DBN Method							CBR w/ NN Distance		
				SubNet1				Δ			Δ
Attribute	**Evidence**	**Time**	**Barren**	**1**	**2**	**Case**	**Value**	**Evidence**	**Case**	**Value**	**Evidence**
					Persistent						
RunID	None	All	No	N/A	N/A	191	192	--	142	141	--
RunDate	None	All	No	N/A	N/A	191	7/10/2005	--	142	5/21/2005	--
VehicleType	16.6	All	No	N/A	N/A	191	16.7	[illegible]	142	19.7	3.1
SupplyDamper	None	All	No	N/A	N/A	191	18.1	--	142	18.4	--
ExhaustDamper	None	All	No	N/A	N/A	191	18.3	--	142	18.3	--
FASetPoint	None	All	No	N/A	N/A	191	18.2	--	142	18.7	--
SillSetPoint	None	All	No	N/A	N/A	191	20.0	--	142	19.3	--
					Zone 0						
ZoneType	15.0	t = 0	No	0	--	103	17.7	[illegible]	142	18.4	3.4
ConveyorSpeed	20.0	t = 0	No	0	--	103	19.9	[illegible]	142	20.7	0.7
FAActual	None	t = 0	No	0	--	103	17.6	--	142	17.5	--
SillActual	None	t = 0	No	0	--	103	18.2	--	142	19.1	--
ZoneSet	None	t = 0	No	0	--	103	16.7	--	142	17.9	--
ZoneActual	None	t = 0	No	0	--	103	15.1	--	142	16.4	--
AmbientTemp	1.0	t = 0	Yes	0	--	103	1.0	0.0	142	1.0	0.0
MetalTemp	6.0	t = 0	No	0	1	103	11.9	5.9	142	18.9	12.9
					Zone 1						
ZoneType	17.0	t = 1	No	1	--	197	18.5	1.5	142	17.2	0.2
ConveyorSpeed	17.0	t = 1	No	1	--	197	16.9	0.1	142	17.1	0.1
FAActual	None	t = 1	No	1	--	197	21.8	--	142	17.2	--
SillActual	None	t = 1	No	1	--	197	18.3	--	142	20.0	--
ZoneSet	None	t = 1	No	1	--	197	19.7	--	142	19.6	--
ZoneActual	None	t = 1	No	1	--	197	24.5	--	142	18.4	--
AmbientTemp	None	t = 1	No	1	--	197	1.0	--	142	1.0	--
MetalTemp	None	t = 1	No	1	2	197	11.5	--	142	16.4	--
					Zone 2						
ZoneType	17.0	t = 2	No	2	--	197	15.8	1.2	142	18.4	1.4
ConveyorSpeed	17.0	t = 2	No	2	--	197	18.1	1.1	142	18.0	1.0
FAActual	None	t = 2	No	2	--	197	22.6	--	142	17.4	--
SillActual	None	t = 2	No	2	--	197	19.6	--	142	20.4	--
ZoneSet	None	t = 2	No	2	--	197	18.1	--	142	18.8	--
ZoneActual	None	t = 2	No	2	--	197	20.2	--	142	18.3	--
AmbientTemp	None	t = 2	No	2	--	197	1.0	--	142	1.0	--
MetalTemp	11.0	t = 2	No	2	3	197	15.3	4.3	142	19.3	8.3
					Zone 3						
ZoneType	18.0	t = 3	No	3	--	185	18.2	[illegible]	142	18.5	0.5
ConveyorSpeed	None	t = 3	No	3	--	185	19.6	--	142	18.7	--
FAActual	None	t = 3	No	3	--	185	23.4	--	142	19.1	--
SillActual	None	t = 3	No	3	--	185	17.8	--	142	19.9	--
ZoneSet	None	t = 3	No	3	--	185	16.8	--	142	17.6	--
ZoneActual	None	t = 3	No	3	--	185	16.9	--	142	17.3	--
AmbientTemp	None	t = 3	No	3	--	185	1.0	--	142	1.0	--
MetalTemp	16.0	t = 3	No	3	4	185	9.5	6.5	142	17.1	1.1
					Zone 4						
ZoneType	20.0	t = 4	No	4	--	150	19.6	0.4	142	19.5	0.5
ConveyorSpeed	17.0	t = 4	No	4	--	150	17.2	0.2	142	17.9	0.9
FAActual	None	t = 4	No	4	--	150	17.3	--	142	18.8	--
SillActual	None	t = 4	No	4	--	150	17.0	--	142	20.7	--
ZoneSet	None	t = 4	No	4	--	150	19.1	--	142	18.0	--
ZoneActual	None	t = 4	No	4	--	150	19.9	--	142	17.2	--
AmbientTemp	None	t = 4	No	4	--	150	1.0	--	142	1.0	--
MetalTemp	24.0	t = 4	No	4	--	150	15.5	8.5	142	16.6	7.4
								9			4
	Dynamic Timing:						0.00000 sec			0.00000 sec	
	Persistent Timing:						0.17015 sec				

Table 5.4: ACRF Oven w/ Tetrad Data Example 3 - Results

Again, when reviewing the "Δ from evidence" column for the CBR/DBN Retrieval/Adaptation algorithm versus the Standard CBR method, the CBR/DBN Retrieval/Adaptation algorithm does extremely well (9 grey cells vs. 4 cells). The persistent attribute evidence is also improved CBR/DBN Retrieval/Adaptation algorithm giving some credence for step 5.

5.3 Example 4 – Mildew DBN w/ Tetrad Generated Data:

In this fourth example, a DBN adapted from [Kja95] will be used for step 1 of the CBR/DBN Retrieval/Adaptation algorithm. In [Kja95] the sample DBN is used to estimate the amount of dry matter in a field of wheat over a specific period of time. This domain was chosen because of the presence of dynamic and persistent variables, however several additional persistent attributes have been added for illustration purposes. Again, Tetrad is used to create the case-base (200 cases) for this DBN so the causal relationships between attributes are maintained.

Figure 5.2 illustrates the "Dry Matter" DBN after step 3 of the CBR/DBN Retrieval Adaptation algorithm. The evidence variables were chosen fairly randomly with the exception that some degree of d-separation was sought and the evidence is within the minimum and maximum limits of that particular attribute. Table 5.5 presents the complete results of the CBR/DBN Retrieval Adaptation algorithm in a similar manner as the previous examples. When reviewing the "Δ from evidence" column for the CBR/DBN Retrieval/Adaptation algorithm versus the Standard CBR method the CBR/DBN Retrieval/Adaptation algorithm does extremely well (9 grey cells vs. 4 cells).

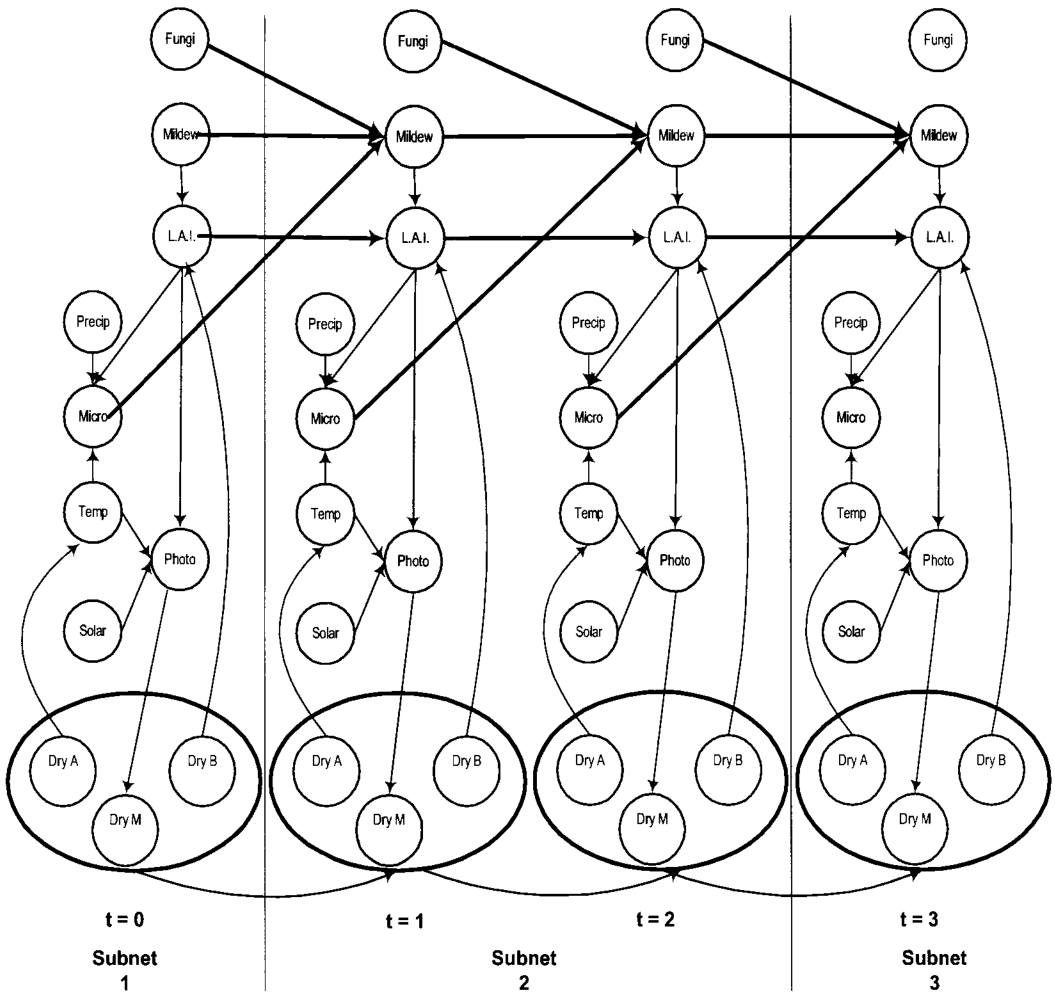


Figure 5.2: Mildew Example 4 - DBN after Step 3 (Adapted from [Kja95])

Test: Mildew **Comments:** Tetrad Data m = 10

Attribute	Evidence	Time	Barren	CBR/DBN Method SubNet1 1	2	Case	Value	Δ Evidence	Convential CBR w/ Euclidean Distance Case	Value	Δ Evidence
Persistent											
da	None	All	No	N/A	N/A	59	47.1	--	107	45.1	--
db	None	All	No	N/A	N/A	59	46.4	--	107	46.1	--
dm	60.7	All	No	N/A	N/A	59	60.1	[redacted]	107	57.3	3.4
Time t = 0											
fu	48.8	t = 0	No	0	1	31	47.0	[redacted]	107	44.2	4.6
ml	None	t = 0	No	0	1	31	45.4	--	107	49.1	--
mi	62.3	t = 0	No	0	1	31	59.7	[redacted]	107	51.2	11.1
la	None	t = 0	No	0	1	31	48.6	--	107	49.4	--
ph	None	t = 0	No	0	--	31	56.1	--	107	51.6	--
pr	None	t = 0	No	0	--	31	145.3	--	107	46.8	--
te	None	t = 0	No	0	--	31	49.8	--	107	45.5	--
so	42.1	t = 0	No	0	--	31	44.2	[redacted]	107	46.0	3.9
Time t = 1											
fu	49.3	t = 1	No	1	--	189	46.4	2.9	107	46.8	[redacted]
ml	45.3	t = 1	No	1	--	189	46.5	[redacted]	107	54.2	8.9
mi	52.1	t = 1	No	1	--	189	50.7	[redacted]	107	47.0	5.1
la	51.3	t = 1	No	1	--	189	46.9	4.4	107	48.1	[redacted]
ph	None	t = 1	No	1	--	189	48.2	--	107	48.9	--
pr	None	t = 1	No	1	--	189	45.9	--	107	45.8	--
te	None	t = 1	No	1	--	189	46.3	--	107	45.5	--
so	49.2	t = 1	No	1	--	189	46.0	3.2	107	46.8	[redacted]
Time t = 2											
fu	49.9	t = 2	No	1	2	189	44.8	5.1	107	44.8	5.1
ml	None	t = 2	No	1	2	189	47.9	--	107	60.1	--
mi	57.9	t = 2	No	1	2	189	46.5	11.4	107	57.2	[redacted]
la	None	t = 2	No	1	2	189	45.1	--	107	51.3	--
ph	None	t = 2	No	1	--	189	44.3	--	107	46.7	--
pr	None	t = 2	No	1	--	189	42.5	--	107	49.5	--
te	None	t = 2	No	1	--	189	47.8	--	107	44.6	--
so	40.9	t = 2	No	1	--	189	41.0	[redacted]	107	39.8	1.1
Time t = 3											
fu	None	t = 3	No	2	--	107	0.0	--	107	0.0	--
ml	65.3	t = 3	No	2	--	107	65.1	0.2	107	65.1	0.2
mi	55.6	t = 3	No	2	--	107	48.1	7.5	107	48.1	7.5
la	43.2	t = 3	No	2	--	107	47.2	4.0	107	47.2	4.0
ph	None	t = 3	No	2	--	107	46.0	--	107	46.0	--
pr	None	t = 3	No	2	--	107	44.1	--	107	44.1	--
te	None	t = 3	No	2	--	107	44.3	--	107	44.3	--
so	47.6	t = 3	No	2	--	107	48.3	0.7	107	48.3	0.7
								7			4
	Dynamic Timing:						0.00000 sec			0.00000 sec	
	Persistent Timing:						0.19027 sec				

Table 5.5: Mildew Example 4 - Results

5.4 Example 5 – BATMobile Sensor DBN w/ Tetrad Generated Data:

In the final example, an adaptation of the BATMobile's sensory DBN [FHKR95] will be used as the starting DBN. In this work autonomous control of vehicles is researched using a foundation based on dynamic probabilistic networks. The DBN used is adapted again by adding additional persistent attributes and only a single car is considered. Tetrad is used to create the case-base.

Figure 5.3 illustrates the "BATmobile" DBN after step 3 of the CBR/DBN Retrieval Adaptation algorithm. Evidence was chosen in a similar manner as Example 4.

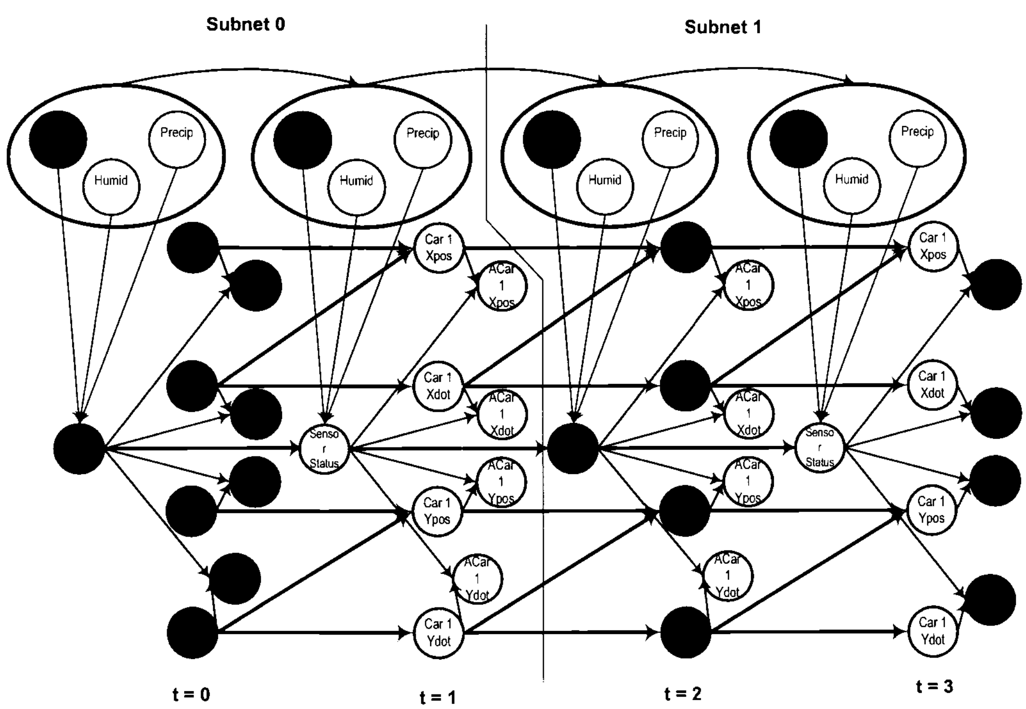


Figure 5.3: Batmobile Example 5 - DBN after Step 3 (Adapted from [FHKR95])

Table 5.6 presents the complete results of the CBR/DBN Retrieval Adaptation algorithm in a similar manner as the previous examples. When reviewing the "Δ from evidence"

Test: BatMobile Sensor **Comments:** Tetrad Data m = 10

		CBR/DBN Method							Convential CBR w/ Euclidean Distance		
				SubNet1				Δ			Δ
Attribute	Evidence	Time	Barren	1	2	Case	Value	Evidence	Case	Value	Evidence
						Persistent					
te	52.6	All	No	N/A	N/A	142	53.8	[redacted]	48	58	5.4
hu	None	All	No	N/A	N/A	142	49.7	--	48	54.4	--
pr	None	All	No	N/A	N/A	142	53.3	--	48	52.3	--
						Time t = 0					
ss	47.3	t = 0	No	0	--	100	59.0	[redacted]	48	60.6	13.3
cxp	None	t = 0	No	0	--	100	55.2	--	48	55.6	--
cxd	None	t = 0	No	0	--	100	52.5	--	48	55.5	--
cyp	None	t = 0	No	0	--	100	52.9	--	48	52.7	--
cyd	None	t = 0	No	0	--	100	54.1	--	48	48.9	--
axp	62.3	t = 0	No	0		100	65.6	**3.3**	48	63.1	[redacted]
axd	71.6	t = 0	No	0	--	100	67.8	[redacted]	48	57.5	14.1
ayp	70.4	t = 0	No	0	--	100	54.5	15.9	48	56.3	[redacted]
ayd	57.3	t = 0	No	0	--	100	58.0	[redacted]	48	59.7	2.4
						Time t = 1					
ss	None	t = 1	No	0	1	100	60.1	--	48	56.7	--
cxp	None	t = 1	No	0	1	100	53.0	--	48	56.4	--
cxd	None	t = 1	No	0	1	100	59.2	--	48	54.0	--
cyp	None	t = 1	No	0	1	100	57.6	--	48	46.8	--
cyd	None	t = 1	No	0	1	100	51.4		48	49.8	
axp	None	t = 1	No	0	--	100	62.1	--	48	57.3	--
axd	None	t = 1	No	0	--	100	65.0	--	48	55.7	--
ayp	None	t = 1	No	0	--	100	67.2	--	48	58.8	--
ayd	None	t = 1	No	0	--	100	59.5	--	48	54.6	--
						Time t = 2					
ss	46.1	t = 2	No	1	--	48	55.8	9.7	48	55.8	9.7
cxp	62.3	t = 2	No	1	--	48	56.3	6.0	48	56.3	6.0
cxd	60.4	t = 2	No	1	--	48	57.5	2.9	48	57.5	2.9
cyp	31.6	t = 2	No	1	--	48	45.6	14.0	48	45.6	
cyd	38.6	t = 2	No	1	--	48	48.8	10.2	48	48.8	10.2
axp	None	t = 2	No	1	--	48	65.4	--	48	65.4	--
axd	None	t = 2	No	1	--	48	50.8	--	48	50.8	--
ayp	None	t = 2	No	1	--	48	47.8	--	48	47.8	--
ayd	None	t = 2	No	1	--	48	54.5	--	48	54.5	--
						Time t = 3					
ss	None	t = 3	No	1	--	48	71.6	--	48	71.6	--
cxp	None	t = 3	No	1	--	48	63.1	--	48	63.1	--
cxd	None	t = 3	No	1	--	48	57.1	--	48	57.1	
cyp	None	t = 3	No	1	--	48	39.4	--	48	39.4	--
cyd	None	t = 3	No	1	--	48	48.3	--	48	48.3	--
axp	90.6	t = 3	No	1	--	48	83.5	7.1	48	83.5	7.1
axd	70.4	t = 3	No	1	--	48	70.8	0.4	48	70.8	0.4
ayp	93.6	t = 3	No	1	--	48	59.1	34.5	48	59.1	34.5
ayd	46.4	t = 3	No	1	--	48	61.1	14.7	48	61.1	14.7
								4			2
	Dynamic Timing:						0.00000 sec			0.00000 sec	
	Persistent Timing:						0.16023 sec				

Table 5.6: BATmobile Example 5 - Results

column for the CBR/DBN Retrieval/Adaptation algorithm versus the Standard CBR method, the CBR/DBN Retrieval/Adaptation algorithm does extremely well (4 shaded cells vs. 1 cell).

Zone	Atribute	Unnormalized Values: Evidence	Unnormalized Values: CBR/DBN Retrieved Value	Unnormalized Values: Conventional CBR Retrieved Value	Normalizied & Weighted Values: Evidence	Normalizied & Weighted Values: CBR/DBN Retrieved Value	Normalizied & Weighted Values: Conventional CBR Retrieved Value	Normalized & Weighted Absolute from Evidence: CBR/DBN Retrieved Value	Normalized & Weighted Absolute from Evidence: Conventional CBR Retrieved Value
	Example 1								
-	Vehicle Type	0	0	7	0	0	600	[redacted]	600
0	Ambient Temp	75	78	80	90	94	96	[redacted]	6
0	Metal Temp	75	75	78	135	135	140	[redacted]	5.4
1	Conv Speed	10	9.8	10.7	160	157	171	[redacted]	11.2
2	Conv Speed	10	9.8	10.7	160	157	171	[redacted]	11.2
2	Metal Temp	250	254	255	450	457	459	[redacted]	9
3	Metal Temp	285	283	281	513	509	506	[redacted]	7.2
4	Conv Speed	20	18.9	19	320	302	304	17.6	[redacted]
4	Metal Temp	285	284	291	513	511	524	[redacted]	10.8
				Totals	2616	2597	3252	[redacted]	682.2
							# Best	[redacted]	1
	Example 2								
0	Ambient Temp	75	79	80	90	95	96	[redacted]	6
0	Metal Temp	75	82	78	135	148	140	12.6	[redacted]
1	Conv Speed	12.6	13.4	11.5	202	214	184	[redacted]	17.6
1	Metal Temp	175	177	193	315	319	347	[redacted]	32.4
2	Conv Speed	12.6	13.4	11.5	202	214	184	[redacted]	17.6
2	Metal Temp	275	281	255	495	506	459	[redacted]	36
4	Conv Speed	20	22.7	19.7	320	363	315	43.2	[redacted]
				Totals	3210	3205	3072	[redacted]	226
							# Best	[redacted]	2
	Example 3								
-	Vehicle Type	16.6	16.7	19.7	2656	2672	3152	[redacted]	496.0
0	Zone Type	15	17.7	18.4	1313	1549	1610	[redacted]	297.5
0	Conv Speed	20	19.9	20.7	2286	2274	2366	[redacted]	80.0
0	Metal Temp	6	11.9	18.9	174	345	549	[redacted]	374.5
1	Zone Type	17	18.5	17.2	1488	1619	1505	131.3	[redacted]
2	Zone Type	17	15.8	18.4	1488	1383	1610	[redacted]	122.5
2	Conv Speed	17	18.1	18	1943	2069	2057	125.7	[redacted]
2	Metal Temp	11	15.3	19.3	319	444	560	[redacted]	241.0
3	Zone Type	18	18.2	18.5	1575	1593	1619	[redacted]	43.8
3	Metal Temp	16	9.5	17.1	465	276	496	188.7	[redacted]
4	Zone Type	20	19.6	19.5	1750	1715	1706	[redacted]	43.8
4	Conv Speed	17	17.2	17.9	1943	1966	2046	[redacted]	102.9
4	Metal Temp	24	15.5	16.6	697	450	482	246.8	[redacted]
				Totals	20638	20885	22312	[redacted]	2191.8
							# Best	[redacted]	4
	Example 4								
-	dm	60.7	60.1	57.3	182	180	172	[redacted]	10.2
0	fu	48.8	47	44.2	244	235	221	[redacted]	23.0
0	mi	62.3	59.7	51.2	374	358	307	[redacted]	66.6
0	so	42.1	44.2	46	379	398	414	[redacted]	35.1
1	fu	49.3	46.4	46.8	247	232	234	14.5	[redacted]
1	ml	45.3	46.5	54.2	362	372	434	[redacted]	71.2
1	mi	52.1	50.7	47	313	304	282	[redacted]	30.6
1	la	51.3	46.9	48.1	308	281	289	26.4	[redacted]
1	so	49.2	46	46.8	443	414	421	28.8	[redacted]
2	mi	57.9	46.5	57.2	347	279	343	68.4	[redacted]
2	s0	40.9	41	39.8	368	369	358	[redacted]	9.9
				Totals	3566	3423	3475	[redacted]	304.1
							# Best	[redacted]	4
	Example 5								
-	te	52.6	53.8	58	158	161	174	[redacted]	16.2
0	ss	47.3	59	60.6	378	472	485	[redacted]	106.4
0	axp	62.3	65.6	63.1	436	459	442	23.1	[redacted]
0	axd	71.6	67.8	57.5	501	475	403	[redacted]	98.7
0	ayp	70.4	54.5	56.3	493	382	394	111.3	[redacted]
0	ayd	57.3	58	59.7	401	406	418	[redacted]	16.8
				Totals	2367	2355	2315	[redacted]	342.4
							# Best	[redacted]	2

Table 5.6: Example Summary

It should also be noted that in the five examples presented in this work, the persistent variables, as determined by the CBR/DBN Retrieval/Adaptation algorithm, are all improved as compared to the conventional CBR method. The next section will present a real time evaluation of determination of the persistent variables

In Table 5.6, a table summarizing of the five examples offered in this thesis is presented. This table contains only the attributes in which a different in the value is retrieve by the CBR/DBN Adaptation/Retrieval algorithm and the standard CBR technique. This table contains the normalized and weight attribute values for comparison purposes. Again the shaded cells represent the superior value based on the evidence. The CBR/DBN Retrieval/Adaptation algorithm performs extremely well in all examples.

5.5 Evaluation of the Combination Level - m :

The persistent attribute determination segment of the algorithm can have significant time complexity consequences. Consider the worst case, choosing $m = 137$ for the ACRF Oven Topcoat example presented in section 4.1, the CBR/DBN Retrieval/Adaptation algorithm then takes 828.78 seconds (~13 minutes and 48 seconds) to determine the best persistent case (see Table 5.7). This leads to the question of what is an appropriate value for the combination level and what is more important accuracy or speed of the algorithm. As mentioned earlier, the number of persistent attributes N_p is generally fixed, thus complexity generally is polynomial in the value m and care should be taken in choosing m. Figure 5.4 plots the persistent case error metric ε versus the time required to compute the metric for the example presented in section 5.1. This information verifies that the CBR/DBN Retrieval/Adaptation Algorithm is polynomial in

m	Dynamic Average Seconds	Persistent Average Seconds	Best Persistent Case	Error Metric ε_n
1	0.00	0.15	112	185.6
2	0.00	0.15	112	175.6
5	0.00	0.17	120	153.6
10	0.00	0.19	120	102.0
15	0.00	0.25	104	98.8
20	0.00	0.53	104	95.6
25	0.00	1.06	104	95.6
30	0.00	2.06	105	15.0
35	0.00	3.64	105	5.8
40	0.00	6.12	105	5.8
45	0.00	9.62	105	5.8
50	0.00	14.61	105	4.2
55	0.00	21.33	105	4.2
60	0.00	30.08	105	4.2
65	0.00	41.34	105	3.8
70	0.01	55.95	105	3.2
75	0.00	73.69	105	3.2
80	0.00	95.85	105	3.2
85	0.00	122.72	105	3.2
90	0.00	153.84	105	2.0
95	0.00	190.43	105	1.8
100	0.00	234.60	105	1.8
105	0.00	284.47	43	0.0
110	0.00	342.13	43	0.0
115	0.00	409.49	43	0.0
120	0.00	486.99	43	0.0
125	0.00	573.36	7	0.0
130	0.00	672.37	7	0.0
135	0.00	781.38	7	0.0
137	0.00	828.78	7	0.0

Table 5.7: Determining the Best Persistent Case from Example 1

growth when compared to m, as mentioned in the previous section. Once again, after reviewing Figure 5.4 it should be noted that at $m > 35$ ($\varepsilon = 5.8$ and time = 3.64 seconds) there is little change in ε and the time required is still reasonable. Table 5.8 and Figure 5.5 provide analysis of the ACRF Oven with Tetrad generated data Example 3 with results comparable to that of example 1. There appears to be a region where there is a reasonable trade-off between accuracy and time complexity. The domain area of the CBR/DBN Retrieval/Adaptation algorithm could have an effect on the determination of m since either a low error metric (greater accuracy) could be crucial or perhaps the time required to compute m could be of greater importance. It should also be noted that at this time there is no provision for ties in the determination of the best persistent case as can be seen from Table 5.7 where the best persistent case changes from 43 to 7 when m goes from 120 to 125, yet the error metric remains at zero.

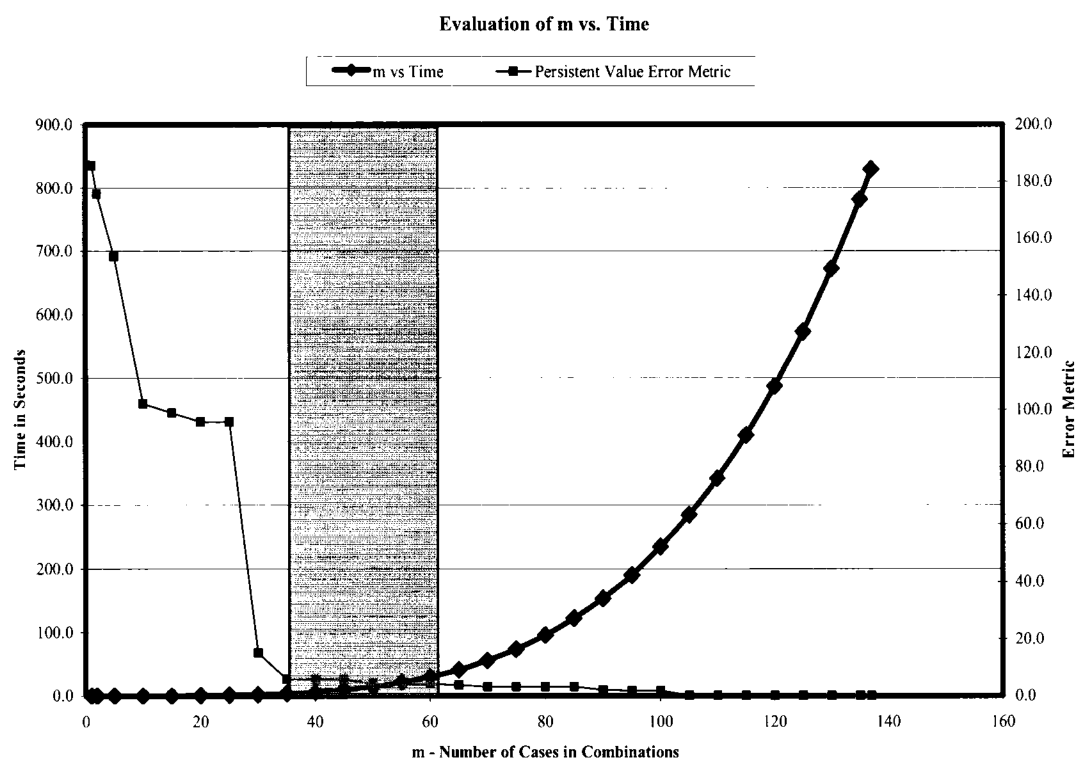


Figure 5.4: Evaluation of the Combination Level (m) from Example 1

m	Dynamic Average Seconds	Persistent Average Seconds	Best Persistent Case	Error Metric ε_n
1	0.0	0.17	1149	1179.1
2	0.0	0.18	103	816.0
5	0.0	0.23	196	587.4
10	0.0	0.19	190	577.3
15	0.0	0.38	196	546.2
20	0.0	0.70	79	426.3
25	0.0	1.36	79	411.3
30	0.0	2.68	48	387.3
35	0.0	4.88	48	387.3
40	0.0	8.24	48	385.2
45	0.0	13.11	48	382.4
50	0.0	19.82	48	382.4
55	0.0	28.94	48	380.8
60	0.0	40.95	48	380.8
65	0.0	56.37	48	380.8
70	0.0	75.81	48	380.8
75	0.0	99.74	48	380.8
80	0.0	129.08	48	380.8
85	0.0	164.54	48	380.8
90	0.0	206.87	48	380.7
95	0.0	256.66	48	380.7
100	0.0	315.15	111	379.4
105	0.0	383.02	111	379.4
110	0.0	461.38	111	379.4
115	0.0	551.13	111	379.4
120	0.0	653.52	48	379.0
125	0.0	769.49	131	359.8
130	0.0	918.18	131	359.8
135	0.0	1053.93	131	359.8
140	0.0	1224.24	131	359.8
145	0.0	1409.67	131	359.8
150	0.0	1594.91	131	359.8
155	0.0	1841.03	131	359.8
160	0.0	2092.45	131	359.8
165	0.0	2334.18	131	359.8
170	0.0	2625.43	131	359.8
175	0.0	2944.22	131	359.8
180	0.0	3325.66	131	359.8
185	0.0	3679.74	172	344.2
190	0.0	4169.58	172	317.3
195	0.0	4590.06	172	317.2
199	0.0	4984.51	172	317.2

Table 5.8: Determining the Best Persistent Case from Example 3

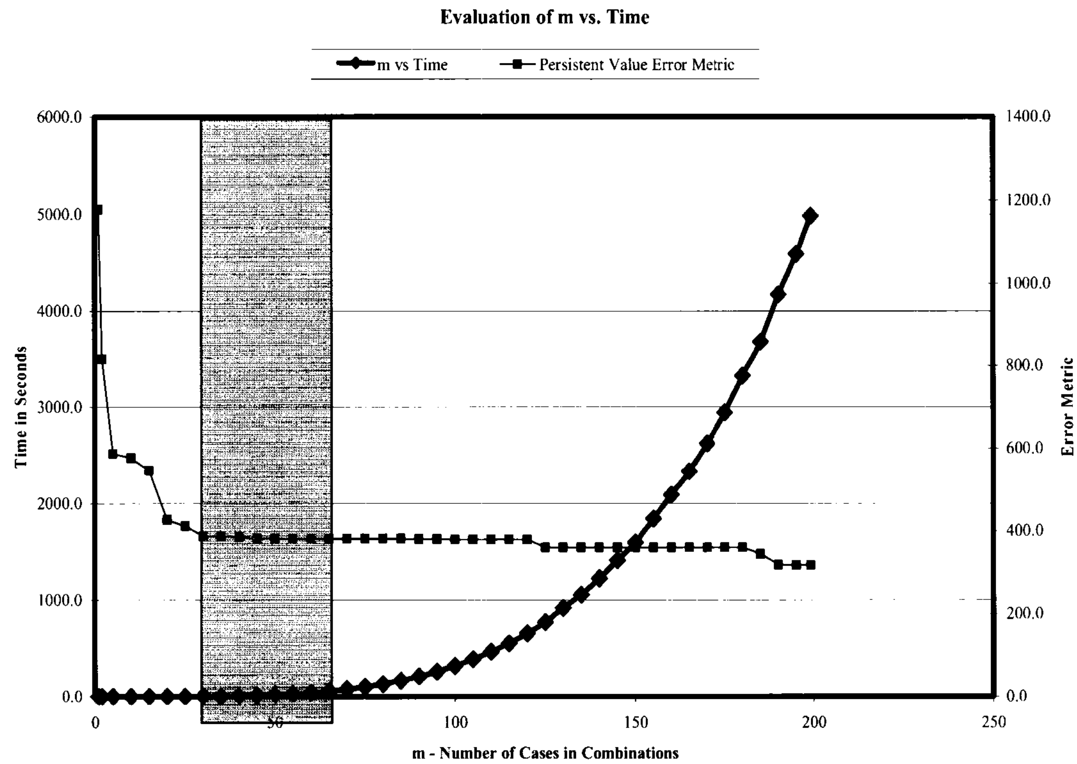


Figure 5.5: Evaluation of the Combination Level (m) from Example 3

Chapter 6: Conclusions and Future Directions

The CBR reasoning methodology seems to fit the definition of artificial intelligence by way of the parallel nature of CBR and how humans begin reasoning or decision processes by remembering past experiences. The methodology of CBR is well developed and researched, but specific details in the implementation are open ended. This presents copious opportunities for research and development into the details of how the methodology of CBR is actually implemented. Motivation for this thesis comes from these opportunities to apply additional artificial intelligence theory in the implementation of CBR systems. Bayesian networks, dynamic Bayesian networks and probabilistic causal models are ideas that have artificial intelligence origins and have been well studied and documented. By utilizing these artificial intelligence concepts in a CBR system a robust CBR system has been developed that assists in two of the more difficult problems in CBR, namely retrieval and adaptation. The idea of using several artificial intelligence ideas to solve a problem is not new in itself, but the techniques used in this thesis lead to some innovative ideas. Furthermore, the real-life industrial test environment presented adds validity and appeal to this research.

This thesis presents a dynamic reasoning system based on the case-based reasoning paradigm that utilizes the causal knowledge elicited from experts or inherent in the data (case-base) to perform the retrieval and adaptation steps. By representing the causal knowledge inherent in a process, in the form of a dynamic Bayesian network and performing analysis using some well known Bayesian network techniques, several independent subnets are formed. These subnets are conditionally independent given the evidence determined from the problem case. From each of these individual subnets a best case(s) is retrieved, realizing this independence of each subnet giving some evidence. The dynamic attribute values from each subnet are determined from the best case retrieved from each of these subnets. The persistent parameters are then determined by calculating an error metric for each of the best cases previously established from the independent subnets. This error metric takes into consideration a number of combinations of best cases determined from the independent subnets and calculate a two part computation involving the variance from evidence and a statistical computation. The final solution case is the combination of the dynamic variables from these best cases and persistent variables from the case with the minimum error metric. The ideas presented in this thesis were demonstrated using a real life dynamic industrial environment with some persistent parameters. Several examples from the real life domain and some dynamic Bayesian network research literature were presented to concrete the ideas of utilizing the causal information inherent in the data (or case-base) and an evaluation was presented comparing standard case-based reasoning techniques. These evaluations validate the ideas presented in this work and give credibility to the CBR/DBN Retrieval/Adaptation algorithm as implemented in a complete CBR system. In all examples the CBR/DBN algorithm out performed standard CBR techniques. The additional complexity that the CBR/DBN Retrieval/Adaptation algorithm contains is analyzed to justify the process.

A major goal of future research is to automate Step 1 – Causal Knowledge Acquisition, of the CBR/DBN Retrieval/Adaptation algorithm. This will allow the application to learn the dynamic Bayesian network from the initial case-base, thereby eliminating the need to elicit this knowledge from experts. Automation of Step 3 – Applying Bayesian Network Concepts, is also planned in the future. The use of Bayesian

Network mechanics on the DBN would require significant effort but would add to the robustness of the application. Implementation and evaluation in different domains is also a key to the success of this project. The domain illustrated in this paper is a robust and realistic environment with real life data, but additional evaluation is also forthcoming. Currently, the work presented here relies exclusively on the causal structure reflected by the DBN. However, it would be desirable to explore the possibility of using the probability distribution in assessing similarities as well as in the adaptation phase.

Appendix A - CBR/DBN Retrieval/Adaptation Application Overview

The CBR/DBN Retrieval Adaptation algorithm is currently implemented in the C++ programming language using the Borland C++ Builder© Version 6.0 [Sai03] [LouC01] [HBSA01] development environment. The choice of programming language and development environment was based solely on user experience and knowledge and should not have had a major impact on the CBR/DBN Retrieval Adaptation algorithm itself. The concern with using this development environment is that it is not a Microsoft® platform and hence integration of third party components is often difficult or impossible.

The representation of the knowledge for the application is done in a standard text format file as are most of the files required for the application. The cases are represented as a simple vector of attributes with each case on a different line and separate attributes represented in columns. Although this format for the case-base is extremely simple it serves the purpose quite well for the case-bases used. A more progressive approach to knowledge representation could have involved an XML document representation that could provide increased portability. The knowledge representation is actually independent of the CBR/DBN Retrieval/Adaptation algorithm and can be modified to accommodate large more complicated sets of data without affecting the operation of the algorithm itself. It should also be noted that the entire case-base is read into RAM and hence read/write operations do not have a significant effect on time complexity. However for large case bases these ideas could be revisited. Figure A.1 illustrates the simple layout of the ACRF oven case base.

The CBR/DBN Retrieval/Adaptation Application consists of three main dialogs or components: process files, dbn components and results of inference dialogs. Initially a process file must be created that details the input files. Figure A.2 illustrates the "modify process file" dialog box that allows the user to modify a current process file. The idea behind a process file is that it contains all of the information about the data for a process. There can be several process files for different process domains, hence adding an element of domain independence to the application. In the new or modify process file dialog box, all the details of the attributes for the process are entered including name, weight, type, etc. The user can create, modify, and delete process files thereby adding great flexibility to the application. Process files are saved with the ".pro" file extension in a standard text format, in the "processes" directory. The sections of a process file are illustrated in Appendices B.1, B.2, and B.3.

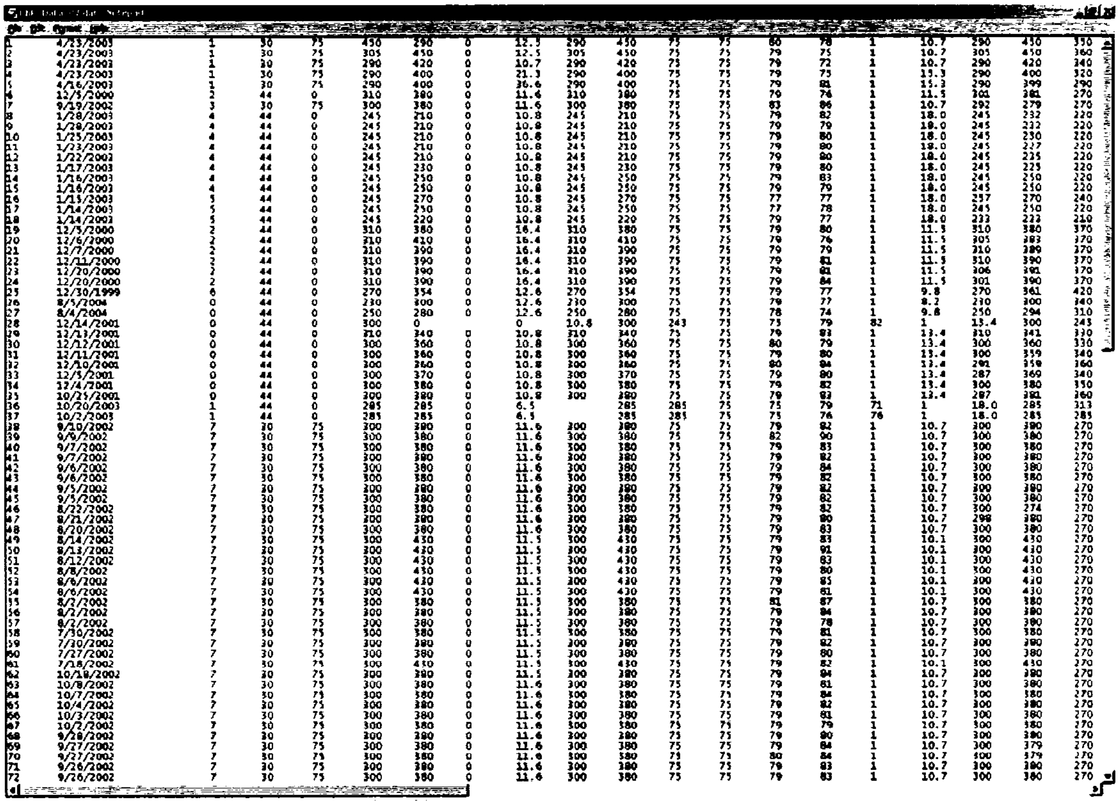


Figure A.1: Case Representation

The second main component involves the setting up of the process DBN. It is assumed that a process file has already been setup and the DBN has the necessary links to an appropriate process file. In this set of dialogs the user can create, modify and delete DBN files. Associated process files, evidence, barren nodes and subnets are entered here and stored as "*.dbn" files in the "dbn" directory. Basically, the *.dbn files contain all the information about the dynamic Bayesian network created in step one of the algorithm. Again, these files are in standard text format for convenient out of application editing. Figure A.3 illustrates the modify DBN dialog box and Appendices B.4, B.5 and B.6 present several sections of the *.dbn file.

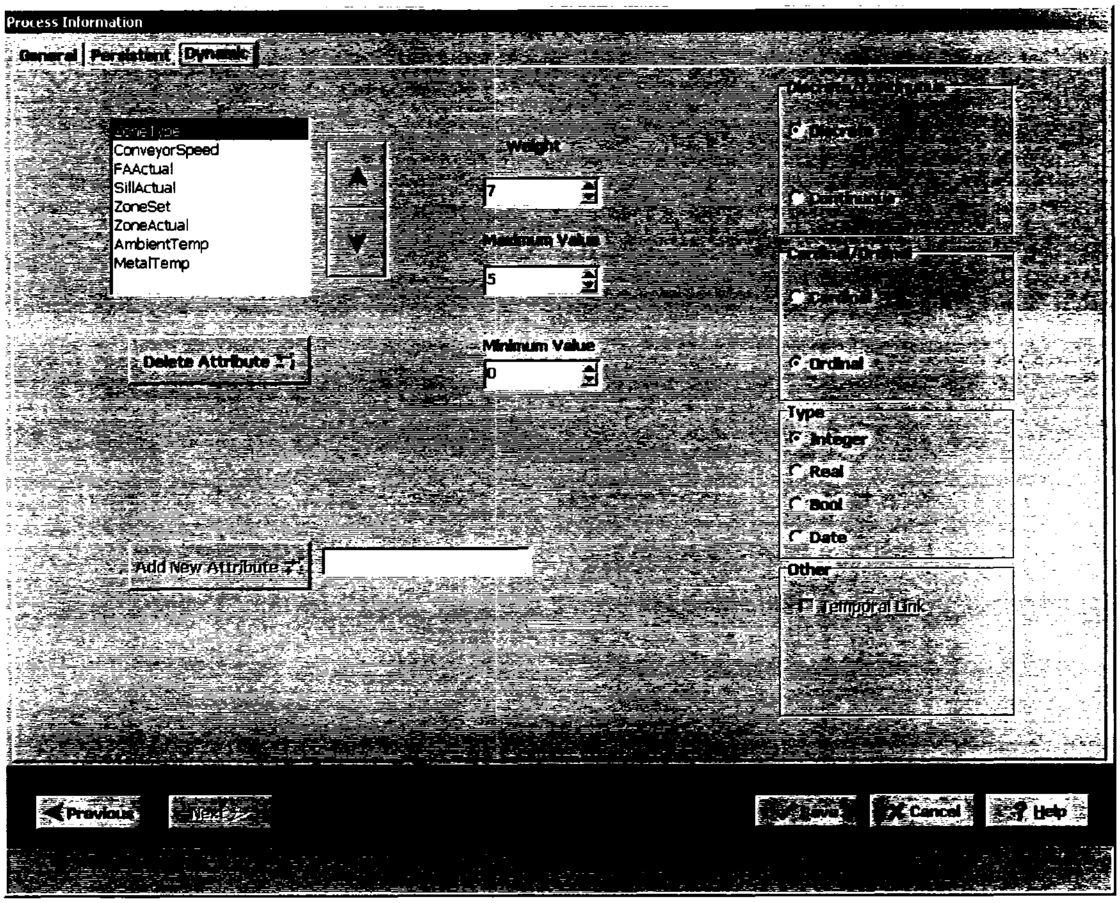


Figure A.2: Process Setup

The third major module is the inference command which actually performs the CBR/DBN Retrieval Adaptation algorithm. This action will ask the user for a "DBN" file and a data file (the case-base) and perform the CBR/DBN Retrieval Adaptation algorithm based on options outlined in the setup menu item. The number of cases m involved in the persistent attribute determination (default = 10) can be adjusted in the setup menu. By selecting the "Inference – Go" menu item and selecting a "DBN" file and a data file the inference process in under way. This process could take some time depending on the value of m, but when complete, all of the information regarding the inference process is displayed in a tabbed dialog box, as illustrated in Figure A.3. This dialog box contains all the information regarding the inference just preformed in a convenient tabbed sheet form. In particular, the Best Case sheet contains all information regarding the selection of the best case and the general sheet contains some timing information. All this information can be saved in a text format by selecting the save button and naming the file. The file is illustrated in Appendices B.7, B.8, B.9, B.10 and B.11.

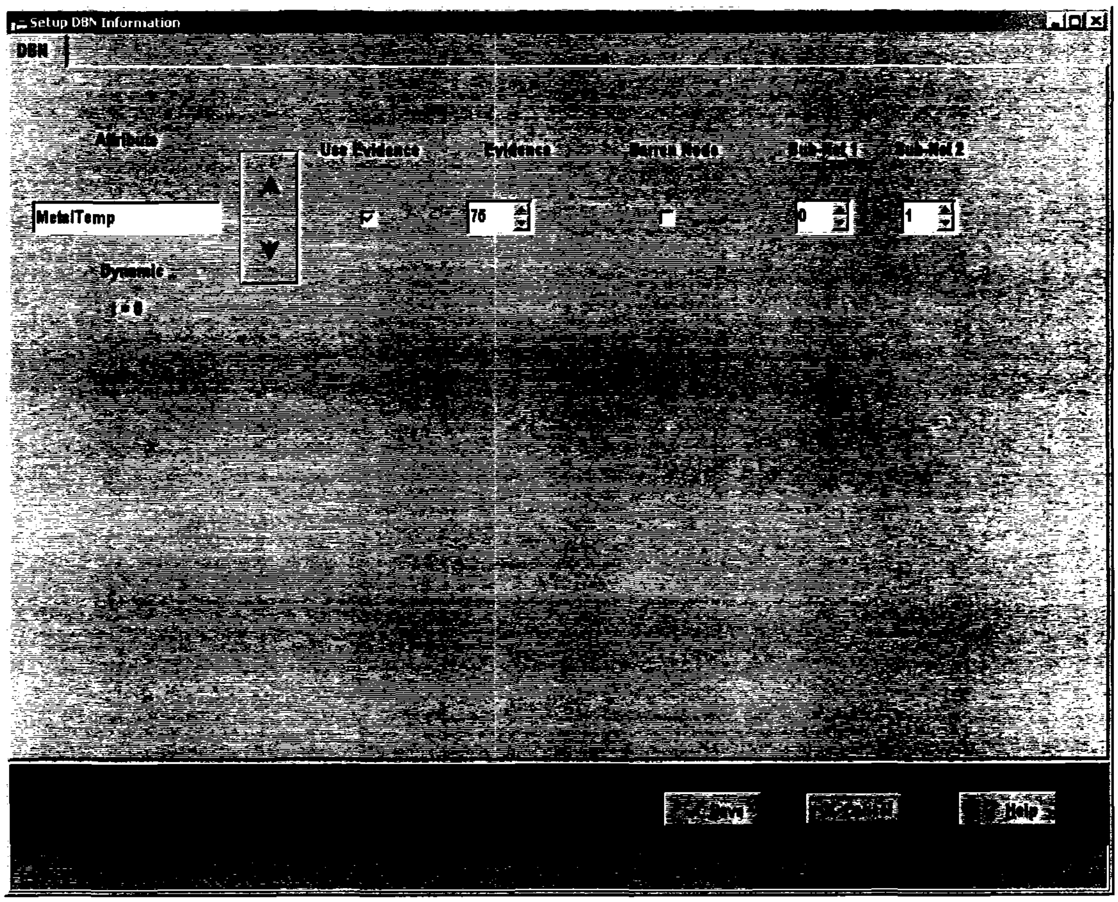


Figure A.3: DBN Setup

In summary, to use the CBR/DBN Retrieval/Adaptation application setup the process file, setup the DBN file, select how many cases (m) to be used in the determination of the persistent attributes and select Inference – Go. The CBR/DBN

Retrieval/Adaptation application provides an easy to use interface for the algorithm with domain independence qualities. It provides flexibility to operate on a variety of domains, through the use of process files and DBN files, something all CBR applications strive to achieve. The two absent mechanisms of the application are: the fact that the DBN created in step one cannot be determined automatically and the fact that Bayesian inference cannot be completed automatically. Both of these two issues, although not implausible, have been investigated and reviewed briefly in previous sections. It is hoped that in future versions of this application that implementations of these two mechanisms can be included to provide a robust application.

Appendix B – CBR/DBN Retrieval/Adaptation Application Files

In this appendix, the files associated with the CBR/DBN Retrieval/Adaptation algorithm are outline. The examples presented are from Example 3 - The ACRF Oven Domain Example with Tetrad data. Figure B.1 presents the "General" section of a process file "*.pro", which contains general configuration information regarding the process. Examples of information contained in this section are the name of the process and the numbers of persistent, dynamic and time slices in the process.

```
[General]
ProcessName        =ACRF Oven #2
ProcessNo          =2
NoPersistent       =7
NoDynamic          =8
NoTimeslices       =5
```

Figure B.1: Process file – General Section Configuration

In Figure B.2, a persistent attribute of the process is configured as can be found in a process file. Each persistent attribute in the process contains a configuration as shown in Figure B.2. Information particular to each persistent attribute such as the attribute name, type of attribute, the weight, a maximum value and a minimum value are configured in this section. One of these sections exists for each persistent attribute.

In Figure B.3, a dynamic attribute of the process is configured as can be found in a process file. Each dynamic attribute in the process contains a configuration as shown in Figure B.3. Information particular to each dynamic attribute such as the attribute name, type of attribute, the weight, a maximum value and a minimum value are configured in this section. One of these sections exists for each persistent attribute.

```
[Persistent0]
Persistent          =RunID
DiscreteConituous   =1
CardinalOrdinal     =1
IRBDType            =0
Weight              =1
TemporalLink        =0
MaxValue            =500
MinValue            =0
```

Figure B.2: Process file – Persistent Attribute Configuration

```
[Dynamic0]
Dynamic             =ZoneType
DiscreteConituous   =1
CardinalOrdinal     =0
IRBDType            =1
Weight              =7
TemporalLink        =0
MaxValue            =22
MinValue            =14
```

Figure B.3: Process file – Dynamic Attribute Configuration

The DBN file contains configuration information regarding the dynamic Bayesian network model of the process and the evidence regarding such. Figure B.4 presents the "General" section of a DBN file "*.dbn", which contains general configuration information regarding process. Examples of information contained in this section are the name of the dynamic Bayesian network, the process file associated with the DBN, the and the numbers of persistent, dynamic and time slices in the process as well as the number of subnets in the pruned and sectioned process DBN.

In Figure B.5, a persistent node of a process DBN is configured as can be found in a DBN file. Each persistent node in the DBN contains a configuration as shown in Figure

B.5. Information specific to a persistent node such as the node name, evidence if available and an indication of whether or not the node is a barren node. There is one of these sections for each persistent attribute.

```
[General]
DBNName             = ACRF Tetrad Data
ProcessFile         =C:\CBR_Tighe\Processes\ACRF_Oven 10.pro
NoPersistent        =7
NoDynamic           =8
NoAttributes        =15
NoTotalAttributes   =47
NoTimeslices        =5
NoSubnets           =4
```

Figure B.4: DBN file – General Section Configuration

```
[PersistentNode0]
Name          =RunID
UseEvidence   =0
IRBDType      =0
IEvidence     =-1
REvidence     =-1
BEvidence     =-1
DEvidence     =0
BarrenNode    =0
SubNet1       =-1
SubNet2       =-1
Weight        =1
MaxValue      =500
MinValue      =0
```

Figure B.5: DBN file – Persistent Node Configuration

In Figure B.6, a dynamic node of a process DBN is configured as can be found in a DBN file. Each dynamic node in the DBN contains a configuration as shown in Figure B.5. Information particular to each persistent node such as the node name, evidence if available, whether or not the node is a barren node, and the subnets that the node is in. There is one of these sections for each persistent attribute.

```
[DynamicNode - 0 - 0]
Name            =ZoneType
UseEvidence     =1
IRBDType        =1
IEvidence       =-1
REvidence       =15
BEvidence       =-1
DEvidence       =0
BarrenNode      =0
SubNet1         =0
SubNet2         =-1
Weight          =7
MaxValue        =22
MinValue        =14
```

Figure B.6: DBN file – Dynamic Node Configuration

In the next several sections the results from the inference operation will be reviewed. These results can be reviewed either in the tabbed dialog box at run time or saved to an output text file (*.out) for further analysis. In Figure B.7 a summary of the inference results is presented including comments, the combination level and timing information. In Table B.1 the original process dynamic Bayesian network is displayed as the problem case. In Table B.2, the original process DBN is displayed as a solution case, that is, with the retrieved and adapted persistent and dynamic variables. The table also displays the cases from where the solution values were found.

```
Inference ID    = ACRF 10
 Comments       = Tetrad Data with ACRF Oven DBN

                Number of Persistent Attributes                      = 7
                Number of Dynamic Attributes                         = 8
                Number of Time Slices                                = 5
                Number of SubNets                                    = 4
                Number of Persistent Combination Attributes (m)      = 130
                Time for Dynamic Inference                           = 0.000000
                Time for Persistent Inference                        = 918.180278
                Best Persistent Case                                 = 131
                Error Metric Case                                    = 359.8
```

Figure B.7: Inference Results – Summary

Table B.1: Inference Results – Problem Case

**************** Dynamic Bayesian Network Summary ******************

Attribute Name	P/D	Type	Time	Evidence	Barren	SubNet1	SubNet2	Weight	Max Value	Min Value
RunID	Persistent	Integer	All	None	No	N/A	N/A	1	500	0
RunDate	Persistent	Date	All	None	No	N/A	N/A	1	10000	0
VehicleType	Persistent	Real	All	16.0	No	N/A	N/A	8	21	16
SupplyDamper	Persistent	Real	All	None	No	N/A	N/A	3	22	15
ExhaustDamper	Persistent	Real	All	None	No	N/A	N/A	3	22	15
FASetPoint	Persistent	Real	All	None	No	N/A	N/A	4	22	15
SillSetPoint	Persistent	Real	All	None	No	N/A	N/A	4	21	13
ZoneType	Dynamic	Real	t = 0	15.0	No	0	--	7	22	14
ConveyorSpeed	Dynamic	Real	t = 0	20.0	No	0	--	8	22	15
FAActual	Dynamic	Real	t = 0	None	No	0	--	6	25	11
SillActual	Dynamic	Real	t = 0	None	No	0	--	6	23	13
ZoneSet	Dynamic	Real	t = 0	None	No	0	--	4	22	14
ZoneActual	Dynamic	Real	t = 0	None	No	0	--	8	26	9
AmbientTemp	Dynamic	Real	t = 0	1.0	Yes	0	--	6	2	1
MetalTemp	Dynamic	Real	t = 0	6.0	No	0	1	9	31	0
ZoneType	Dynamic	Real	t = 1	17.0	No	1	--	7	22	14
ConveyorSpeed	Dynamic	Real	t = 1	17.0	No	1	--	8	22	15
FAActual	Dynamic	Real	t = 1	None	No	1	--	6	25	11
SillActual	Dynamic	Real	t = 1	None	No	1	--	6	23	13
ZoneSet	Dynamic	Real	t = 1	None	No	1	--	4	22	14
ZoneActual	Dynamic	Real	t = 1	None	No	1	--	8	26	9
AmbientTemp	Dynamic	Real	t = 1	None	No	1	--	6	2	1
MetalTemp	Dynamic	Real	t = 1	None	No	1	--	9	31	0
ZoneType	Dynamic	Real	t = 2	17.0	No	1	--	7	22	14
ConveyorSpeed	Dynamic	Real	t = 2	17.0	No	1	--	8	22	15
FAActual	Dynamic	Real	t = 2	None	No	1	--	6	25	11
SillActual	Dynamic	Real	t = 2	None	No	1	--	6	23	13
ZoneSet	Dynamic	Real	t = 2	None	No	1	--	4	22	14
ZoneActual	Dynamic	Real	t = 2	None	No	1	--	8	26	9
AmbientTemp	Dynamic	Real	t = 2	None	No	1	--	6	2	1
MetalTemp	Dynamic	Real	t = 2	11.0	No	1	2	9	31	0
ZoneType	Dynamic	Real	t = 3	18.0	No	2	--	7	22	14
ConveyorSpeed	Dynamic	Real	t = 3	None	No	2	--	8	22	15
FAActual	Dynamic	Real	t = 3	None	No	2	--	6	25	11
SillActual	Dynamic	Real	t = 3	None	No	2	--	6	23	13
ZoneSet	Dynamic	Real	t = 3	None	No	2	--	4	22	14
ZoneActual	Dynamic	Real	t = 3	None	No	2	--	8	26	9
AmbientTemp	Dynamic	Real	t = 3	None	No	2	--	6	2	1
MetalTemp	Dynamic	Real	t = 3	16.0	No	2	3	9	31	0
ZoneType	Dynamic	Real	t = 4	20.0	No	3	--	7	22	14
ConveyorSpeed	Dynamic	Real	t = 4	17.0	No	3	--	8	22	15
FAActual	Dynamic	Real	t = 4	None	No	3	--	6	25	11
SillActual	Dynamic	Real	t = 4	None	No	3	--	6	23	13
ZoneSet	Dynamic	Real	t = 4	None	No	3	--	4	22	14
ZoneActual	Dynamic	Real	t = 4	None	No	3	--	8	26	9
AmbientTemp	Dynamic	Real	t = 4	None	No	3	--	6	2	1
MetalTemp	Dynamic	Real	t = 4	24.0	No	3	--	9	31	0

Table B.2: Inference Results – Solution Case

**************** Best Case Summary ******************

Attribute Name	P/D	Type	Time	Evidence	Barren	Case #	Value	SubNet1	SubNet2
RunID	Persistent	Integer	All	None	No	132	133	N/A	N/A
RunDate	Persistent	Date	All	None	No	132	5/12/2005	N/A	N/A
VehicleType	Persistent	Real	All	16.0	No	132	19.0	N/A	N/A
SupplyDamper	Persistent	Real	All	None	No	132	19.2	N/A	N/A
ExhaustDamper	Persistent	Real	All	None	No	132	19.0	N/A	N/A
FASetPoint	Persistent	Real	All	None	No	132	20.5	N/A	N/A
SillSetPoint	Persistent	Real	All	None	No	132	18.9	N/A	N/A
ZoneType	Dynamic	Real	t = 0	15.0	No	103	17.7	0	--
ConveyorSpeed	Dynamic	Real	t = 0	20.0	No	103	19.9	0	--
FAActual	Dynamic	Real	t = 0	None	No	103	17.6	0	--
SillActual	Dynamic	Real	t = 0	None	No	103	18.2	0	--
ZoneSet	Dynamic	Real	t = 0	None	No	103	16.7	0	--
ZoneActual	Dynamic	Real	t = 0	None	No	103	15.1	0	--
AmbientTemp	Dynamic	Real	t = 0	1.0	Yes	103	1.0	0	--
MetalTemp	Dynamic	Real	t = 0	6.0	No	103	11.9	0	1
ZoneType	Dynamic	Real	t = 1	17.0	No	197	18.5	1	--
ConveyorSpeed	Dynamic	Real	t = 1	17.0	No	197	16.9	1	--
FAActual	Dynamic	Real	t = 1	None	No	197	21.8	1	--
SillActual	Dynamic	Real	t = 1	None	No	197	18.3	1	--
ZoneSet	Dynamic	Real	t = 1	None	No	197	19.7	1	--
ZoneActual	Dynamic	Real	t = 1	None	No	197	24.5	1	--
AmbientTemp	Dynamic	Real	t = 1	None	No	197	1.0	1	--
MetalTemp	Dynamic	Real	t = 1	None	No	197	11.5	1	--
ZoneType	Dynamic	Real	t = 2	17.0	No	197	15.8	1	--
ConveyorSpeed	Dynamic	Real	t = 2	17.0	No	197	18.1	1	--
FAActual	Dynamic	Real	t = 2	None	No	197	22.6	1	--
SillActual	Dynamic	Real	t = 2	None	No	197	19.6	1	--
ZoneSet	Dynamic	Real	t = 2	None	No	197	18.1	1	--
ZoneActual	Dynamic	Real	t = 2	None	No	197	20.2	1	--
AmbientTemp	Dynamic	Real	t = 2	None	No	197	1.0	1	--
MetalTemp	Dynamic	Real	t = 2	11.0	No	197	15.3	1	2
ZoneType	Dynamic	Real	t = 3	18.0	No	185	18.2	2	--
ConveyorSpeed	Dynamic	Real	t = 3	None	No	185	19.6	2	--
FAActual	Dynamic	Real	t = 3	None	No	185	23.4	2	--
SillActual	Dynamic	Real	t = 3	None	No	185	17.8	2	--
ZoneSet	Dynamic	Real	t = 3	None	No	185	16.8	2	--
ZoneActual	Dynamic	Real	t = 3	None	No	185	16.9	2	--
AmbientTemp	Dynamic	Real	t = 3	None	No	185	1.0	2	--
MetalTemp	Dynamic	Real	t = 3	16.0	No	185	9.5	2	3
ZoneType	Dynamic	Real	t = 4	20.0	No	150	19.6	3	--
ConveyorSpeed	Dynamic	Real	t = 4	17.0	No	150	17.2	3	--
FAActual	Dynamic	Real	t = 4	None	No	150	17.3	3	--
SillActual	Dynamic	Real	t = 4	None	No	150	17.0	3	--
ZoneSet	Dynamic	Real	t = 4	None	No	150	19.1	3	--
ZoneActual	Dynamic	Real	t = 4	None	No	150	19.9	3	--
AmbientTemp	Dynamic	Real	t = 4	None	No	150	1.0	3	--
MetalTemp	Dynamic	Real	t = 4	24.0	No	150	15.5	3	--

In Table B.3, a summary of the normalized and weighted distance from the problem case for each case in the case-base is presented. Each case is listed vertically and the distance from the problem case for each subnet is listed horizontally.

Table B.3: Inference Results – Subnet Distance Summary

**************** Nearest Neighbor Distance Summary ******************

Case	SubNet - 0	SubNet - 1	SubNet - 2	SubNet - 3
1	1,406.4	1,891.3	956.4	976.4
2	1,341.7	1,313.0	708.9	1,205.9
3	1,312.0	1,567.0	957.0	852.0
4	871.1	1,624.7	466.0	824.9
5	1,438.4	1,609.7	1,210.3	1,119.7
6	1,440.7	1,384.7	777.3	1,101.8
7	881.0	1,415.5	519.1	818.8
8	1,343.3	1,523.0	782.7	933.3
9	1,276.3	1,020.4	557.8	1,212.3
10	1,574.3	1,760.1	1,275.8	1,254.5
11	1,289.0	1,535.1	620.8	941.0
12	1,242.4	1,484.0	870.4	1,007.6
13	1,373.1	1,583.4	741.9	881.5
14	1,220.3	1,507.2	999.7	1,198.4
15	1,416.8	1,869.5	846.6	856.4
16	1,092.4	1,270.2	562.4	807.4
17	1,207.6	1,876.9	650.4	701.7
18	1,007.5	1,084.5	379.4	806.7
19	920.3	1,085.5	444.8	1,201.9
20	1,319.0	1,346.7	724.7	1,020.4
21	1,668.1	1,608.0	1,000.0	1,308.1
22	876.6	1,257.1	586.4	847.2
23	1,246.2	1,670.1	726.0	1,200.2
24	1,208.2	1,583.7	672.0	885.9
25	1,424.5	1,472.0	875.4	1,563.5
26	829.3	1,317.2	367.7	900.3
27	1,182.5	1,455.5	623.5	881.5
28	970.1	1,290.9	546.0	912.0
29	1,497.5	1,881.4	1,232.9	1,307.5
30	980.3	1,518.4	889.6	927.6
31	974.1	1,081.9	563.4	1,224.4
32	864.1	1,170.3	616.0	1,226.4
33	1,491.5	1,904.1	1,201.1	1,089.1
34	739.1	764.1	821.4	1,393.8
35	1,057.5	1,114.2	503.5	1,249.3
36	1,318.6	1,763.8	1,200.3	1,160.0
37	1,233.4	1,247.4	1,025.2	960.3
38	1,309.8	1,324.1	757.3	1,150.7
39	1,220.4	1,352.8	798.0	1,368.9
40	1,385.8	1,649.3	689.3	1,082.2
41	1,362.2	1,972.4	975.3	762.9
42	1,188.4	1,027.6	558.0	1,130.0

Table B.3: Inference Results – Subnet Distance Summary (continued)

43	1,536.2	1,914.2	1,144.1	1,268.7
44	1,667.3	1,739.5	1,358.3	1,156.5
45	1,007.1	1,217.6	379.4	1,052.8
46	1,203.2	1,459.5	742.2	1,001.6
47	1,562.0	1,779.2	887.2	1,187.2
48	1,233.9	1,367.6	838.3	658.9
49	1,178.7	1,144.9	499.8	1,197.7
50	1,483.9	1,566.6	675.3	1,349.0
51	1,372.7	1,914.9	1,190.1	1,137.6
52	1,336.4	1,942.7	899.9	1,018.0
53	1,068.8	1,314.2	556.1	729.6
54	1,133.0	1,464.1	634.0	900.5
55	961.5	1,248.8	564.8	837.3
56	1,411.4	1,488.9	935.9	1,095.1
57	1,358.9	2,051.0	1,347.0	1,375.2
58	1,530.3	1,674.6	776.1	1,077.1
59	1,257.3	1,373.2	796.2	702.6
60	1,374.6	1,659.4	699.5	921.2
61	920.9	1,124.2	939.0	911.8
62	1,356.7	1,427.3	767.2	964.9
63	1,374.4	1,758.0	592.2	900.8
64	1,168.9	1,612.9	958.3	1,106.2
65	1,288.3	1,602.9	1,006.3	884.9
66	866.3	830.7	765.4	1,237.7
67	1,407.2	1,646.7	963.0	941.4
68	993.6	1,428.5	617.3	1,010.2
69	1,211.9	1,141.4	668.9	748.0
70	1,121.4	1,619.1	943.3	1,178.1
71	1,037.7	1,680.9	722.8	554.4
72	1,029.7	888.4	654.3	987.2
73	1,450.4	1,701.5	1,209.1	1,153.6
74	1,383.4	2,117.1	977.6	1,037.2
75	1,230.0	2,023.8	765.9	880.5
76	1,311.5	1,688.8	680.3	1,081.8
77	950.9	1,769.9	967.3	1,080.0
78	1,493.9	2,217.0	1,601.0	1,642.7
79	1,424.0	1,485.9	802.5	911.8
80	729.5	912.7	479.4	1,226.9
81	1,217.4	1,828.9	920.4	882.2
82	1,272.9	1,461.0	797.9	792.2
83	1,440.9	2,156.3	1,273.7	986.7
84	1,255.5	1,303.3	600.1	708.6
85	1,025.8	1,397.5	683.8	771.0
86	1,337.2	1,938.5	704.9	922.6
87	1,220.2	1,738.8	1,056.5	1,067.0
88	1,061.5	1,936.4	632.0	947.7
89	1,591.0	2,221.5	1,376.5	1,345.5
90	1,275.6	1,240.9	486.7	644.1
91	1,032.0	1,335.4	837.2	990.9
92	1,394.5	1,865.4	1,392.4	1,131.9
93	1,204.3	1,585.6	896.2	1,326.1
94	991.3	1,561.0	523.0	968.7
95	1,397.2	1,894.1	1,078.7	1,073.3
96	1,377.2	2,307.5	1,032.4	1,110.4

Table B.3: Inference Results – Subnet Distance Summary (continued)

97	1,369.1	1,517.7	680.2	661.8
98	1,181.7	1,829.3	627.8	784.7
99	1,091.9	1,467.5	879.1	903.5
100	1,772.7	2,024.1	1,360.2	1,272.8
101	911.3	1,543.0	534.6	1,057.2
102	1,333.4	2,058.2	1,052.6	1,005.7
103	547.0	976.0	449.0	983.0
104	1,001.2	953.1	359.8	882.1
105	1,041.3	1,311.3	735.8	1,130.0
106	1,079.6	1,192.1	589.2	1,059.8
107	1,248.9	1,473.3	937.3	809.7
108	1,104.4	1,481.9	625.9	807.8
109	1,290.8	1,419.0	694.2	1,250.4
110	1,219.4	1,027.9	427.9	618.4
111	1,073.6	1,273.4	596.2	1,061.7
112	1,235.7	1,133.5	596.1	952.6
113	1,422.6	1,893.0	872.2	1,198.6
114	1,320.3	1,577.2	869.2	989.1
115	1,290.2	1,462.7	683.5	873.8
116	1,637.6	1,692.5	1,094.9	995.7
117	1,276.9	1,362.9	730.9	927.0
118	1,166.1	1,751.0	897.3	1,051.2
119	1,745.3	2,188.4	1,616.0	1,448.5
120	1,816.3	2,429.2	1,503.9	1,477.4
121	1,163.6	1,151.5	531.0	1,173.8
122	1,010.2	1,466.5	874.2	957.5
123	1,333.1	1,814.0	817.0	1,019.9
124	1,003.1	1,627.5	1,014.1	1,089.2
125	1,192.0	2,004.2	893.3	998.8
126	1,156.2	1,615.1	1,108.2	825.6
127	1,203.2	1,838.2	856.6	1,137.7
128	1,525.1	1,882.1	1,472.7	1,614.8
129	1,232.8	1,742.2	586.0	819.9
130	1,235.1	1,634.5	884.4	1,083.8
131	1,148.4	1,327.8	725.2	1,099.1
132	1,282.9	1,599.8	794.4	1,062.0
133	1,084.6	1,439.6	643.8	995.6
134	1,069.8	1,321.6	800.1	1,058.3
135	1,238.9	1,408.9	667.9	873.9
136	1,405.1	1,439.2	672.4	870.2
137	918.5	1,513.4	832.7	1,025.0
138	893.7	1,231.8	439.6	733.5
139	1,173.8	1,694.0	849.8	1,173.3
140	1,516.6	1,666.2	1,156.4	1,115.4
141	1,849.3	2,032.1	1,131.0	1,210.0
142	895.7	1,036.9	466.4	538.9
143	1,222.0	1,328.8	448.6	794.2
144	601.6	918.3	486.7	1,253.8
145	1,075.8	1,470.5	698.8	879.0
146	1,407.6	1,769.1	815.9	888.0
147	1,406.6	1,738.3	715.7	1,136.9
148	1,232.7	1,497.0	824.8	1,049.3
149	1,434.9	1,289.0	538.9	839.8

Table B.3: Inference Results – Subnet Distance Summary (continued)

150	1,027.2	1,316.8	391.7	462.8
151	1,369.5	1,761.7	1,139.6	1,099.0
152	1,073.2	1,364.3	911.5	1,214.2
153	1,011.1	1,554.3	854.8	1,046.3
154	1,416.7	1,895.7	1,100.0	1,039.4
155	838.6	1,349.3	473.6	966.6
156	1,011.6	1,048.6	608.1	762.2
157	1,416.1	1,694.9	877.6	1,047.0
158	681.0	1,665.9	664.7	928.5
159	1,052.5	1,754.8	842.0	1,110.2
160	860.3	1,305.2	389.0	1,165.5
161	1,366.9	1,420.5	739.0	1,221.9
162	1,283.2	1,690.7	630.0	890.7
163	1,506.6	1,909.2	1,119.9	1,155.1
164	1,311.5	1,584.4	720.2	646.1
165	1,436.8	1,987.5	995.2	903.5
166	1,315.1	1,481.3	985.9	1,233.8
167	887.2	1,189.9	480.9	846.5
168	1,403.6	1,554.2	865.3	770.2
169	1,257.2	1,788.1	848.6	1,018.5
170	1,239.2	1,509.3	675.5	929.5
171	1,499.3	2,150.3	1,124.7	1,194.4
172	1,266.7	1,438.9	745.3	793.4
173	1,614.0	1,998.5	1,210.4	1,398.2
174	844.7	1,482.0	674.8	663.3
175	1,091.2	1,519.6	715.2	798.8
176	824.5	1,189.0	497.5	922.4
177	1,423.8	1,853.1	901.2	1,110.3
178	1,466.9	2,127.9	1,108.5	1,265.2
179	1,681.1	1,935.0	1,097.2	1,132.1
180	993.3	1,150.4	489.1	966.5
181	1,599.3	1,960.3	966.7	760.1
182	1,795.1	2,106.9	1,270.2	1,366.4
183	948.8	1,432.1	656.2	933.9
184	1,514.0	1,748.5	866.2	837.9
185	953.5	1,018.2	317.3	939.8
186	1,306.1	1,594.9	652.8	1,085.4
187	1,315.4	1,626.4	767.2	993.6
188	1,439.8	1,601.2	564.6	1,062.0
189	1,121.3	1,010.6	600.6	1,111.6
190	1,084.4	1,435.6	498.8	895.9
191	613.9	1,377.9	382.7	849.7
192	1,409.2	1,877.5	997.4	997.4
193	1,391.9	1,230.3	877.3	1,005.0
194	906.9	1,005.3	491.6	1,107.2
195	1,023.6	1,324.4	766.5	836.5
196	1,230.2	1,554.1	1,010.1	1,073.4
197	845.8	729.9	378.7	1,056.6
198	1,416.7	1,704.5	1,004.6	835.3
199	1,797.3	1,732.7	1,283.4	1,049.7

In Table B.4, a summary of the case rank (best being 1, and worst being 199) for each case in the case-base with respect to each subnet is presented.

Table B.4: Inference Results – Case Rank by. subnet

*************** Case Rank Summary *****************

Rank	SubNet - 0	SubNet - 1	SubNet - 2	SubNet - 3
1	103	197	185	150
2	144	34	104	142
3	191	66	26	71
4	158	72	197	110
5	80	80	45	90
6	34	144	18	164
7	176	104	191	48
8	26	103	160	97
9	155	194	150	174
10	174	189	110	17
11	197	185	138	59
12	160	9	19	84
13	32	42	143	53
14	66	110	103	138
15	4	142	4	69
16	22	156	142	181
17	7	31	155	156
18	167	18	80	41
19	138	19	167	168
20	142	35	90	85
21	194	61	144	98
22	101	112	180	82
23	137	69	194	172
24	19	49	176	143
25	61	180	190	175
26	183	121	49	18
27	77	32	35	16
28	185	176	7	108
29	55	167	94	107
30	28	106	121	7
31	31	45	101	129
32	30	193	149	4
33	94	138	28	126
34	180	90	53	198
35	68	37	9	195
36	104	55	42	55
37	124	22	16	184
38	45	16	31	149
39	18	111	188	167
40	122	149	55	22
41	153	28	129	191
42	156	84	22	3
43	195	160	106	15

Table B.4: Inference Results – Case Rank by. Subnet (continued)

44	85	105	63	136
45	150	2	112	115
46	72	53	111	135
47	91	150	84	145
48	71	26	189	75
49	105	134	156	13
50	159	38	32	27
51	35	195	68	104
52	88	131	11	81
53	53	143	27	65
54	134	91	108	24
55	152	20	98	146
56	111	155	162	162
57	145	39	88	190
58	106	117	54	26
59	190	152	133	54
60	133	48	17	63
61	175	59	186	99
62	99	191	72	165
63	16	6	183	79
64	108	85	158	61
65	189	135	135	28
66	70	7	69	60
67	54	109	24	176
68	131	161	136	86
69	126	62	174	117
70	121	68	50	30
71	118	183	170	158
72	64	190	97	170
73	139	172	76	8
74	49	136	115	183
75	98	133	85	185
76	27	27	40	11
77	42	46	109	67
78	125	82	145	88
79	46	115	60	112
80	127	54	86	122
81	93	122	2	37
82	17	99	175	62
83	24	145	147	180
84	69	25	164	155
85	81	107	71	94
86	110	166	20	1
87	87	108	131	103
88	14	174	23	83
89	39	12	117	72
90	143	79	105	114
91	75	56	161	91
92	196	148	13	187
93	148	14	46	133
94	129	170	172	116
95	37	137	38	192
96	48	97	66	125

Table B.4: Inference Results – Case Rank by. Subnet (continued)

97	130	30	75	46
98	112	175	195	193
99	135	8	62	102
100	170	11	187	12
101	12	101	58	68
102	23	196	6	52
103	107	168	8	169
104	84	153	132	123
105	169	94	59	20
106	59	50	82	137
107	172	3	39	74
108	82	114	134	154
109	90	13	79	153
110	9	24	146	157
111	117	164	123	148
112	132	93	34	199
113	162	186	148	118
114	65	132	137	45
115	11	188	91	197
116	115	65	48	101
117	109	21	159	134
118	186	5	15	106
119	38	64	169	111
120	164	126	139	188
121	76	70	153	132
122	3	4	127	87
123	166	187	168	95
124	187	124	184	196
125	36	130	114	58
126	20	67	12	77
127	114	40	113	76
128	123	60	122	40
129	102	158	25	130
130	52	140	193	186
131	86	23	157	33
132	2	58	99	124
133	8	71	130	56
134	62	76	47	151
135	57	162	30	131
136	41	116	125	6
137	161	139	93	64
138	97	157	118	194
139	151	73	52	159
140	51	198	177	177
141	13	199	152	96
142	63	147	81	189
143	60	87	56	140
144	96	44	107	5
145	74	129	61	42
146	40	184	70	105
147	193	118	1	92
148	92	159	3	179
149	95	63	64	147

Table B.4: Inference Results – Case Rank by. Subnet (continued)

150	168	10	67	51
151	136	151	181	127
152	1	36	77	38
153	147	146	41	73
154	67	77	74	163
155	146	47	166	44
156	192	169	165	36
157	56	123	192	160
158	157	81	14	139
159	198	98	21	121
160	154	127	198	70
161	15	177	65	47
162	113	92	196	171
163	177	15	124	49
164	79	17	37	14
165	25	192	96	113
166	149	29	102	23
167	165	128	87	19
168	5	1	95	2
169	188	113	116	141
170	6	95	179	9
171	83	154	154	152
172	73	33	126	161
173	178	163	178	31
174	50	43	163	32
175	33	51	171	80
176	78	179	141	166
177	29	88	151	66
178	171	86	43	35
179	163	52	140	109
180	184	181	51	144
181	140	41	36	10
182	128	165	33	178
183	58	173	73	43
184	43	125	5	100
185	47	75	173	29
186	10	100	29	21
187	89	141	182	93
188	181	57	83	89
189	173	102	10	50
190	116	182	199	182
191	44	74	57	39
192	21	178	44	57
193	179	171	100	34
194	119	83	89	173
195	100	119	92	119
196	182	78	128	120
197	199	89	120	25
198	120	96	78	128
199	141	120	119	78

Appendix C – Tetrad III Files

In this appendix, the files associated with the Tetrad software are reviewed. The examples presented are from Example 3 - The ACRF Oven Domain Example with Tetrad data. In Figure C.1, a graph structure in the format required by Tetrad is illustrated. The first line indicates that the file is a graph definition and each of the following lines define an arc from the first attribute to the second attribute. For example fa denotes the persistent attribute, fresh air set point in the oven domain, fa0 denotes the actual fresh air temperature in at time zero. The first line of Figure C.1, indicates that there is an arc from the fresh air set point to the actual fresh air temperature at time zero. Other notations include: sd = supply damper, ed = exhaust damper, ss = sill set point, zs = zone set point, sa = actual sill temperature, zt = actual zone temperature, at = ambient temperature, mt = metal temperature, cs = conveyor speed, and vt = vehicle type. A number after the attribute code indicates a dynamic variable and the time slice of the dynamic variable.

```
/graph
fa  fa0
sd  fa0
ed  fa0
ss  sa0
zs0  za0
fa0  za0
za0  at0
za0  mt0
vt  mt0
cs0  mt0
zt0  mt0
sa0  mt0
mt0  mt1
fa  fa1
sd  fa1
ed  fa1
ss  sa1
za1  at1
za1  mt1
vt  mt1
cs1  mt1
zt1  mt1
sa1  mt1
mt1  mt2
fa  fa2
sd  fa2
ed  fa2
ss  sa2
zs2  za2
fa2  za2
za2  at2
za2  mt2
vt  mt2
cs2  mt2
mt2  mt3
fa  fa3
sd  fa3
ed  fa3
ss  sa3
zs3  za3
fa3  za3
za3  at3
za3  mt3
vt  mt3
cs3  mt3
zt3  mt3
sa3  mt3
mt3  mt4
fa  fa4
sd  fa4
ed  fa4
fa4  za4
Za4  at4
Za4  mt4
vt  mt4
Cs4  mt4
zt4  mt4
sa4  mt4
fa1 za1
sa2 mt2
zs4 za4
zs1 za1
zt2 mt2
Ss sa4
```

Figure C.1: Tetrad III – Graph File from Example 3

In Figure C.2 and Figure C.3, the linear equation model is presented. The file is built by using the "BuildModel" command with the name of the graph file from Figure C.1 as an argument.

```
/graph
 fa fa0  1.4781
 fa fa1  1.4755
 fa fa2  1.3196
 fa fa3  0.7034
 fa fa4  1.2279
 fa0 za0  0.0259
 sd fa0  0.5469
 sd fa1  0.5137
 sd fa2  0.9903
 sd fa3  0.6291
 sd fa4  0.4778
 ed fa0  0.9254
 ed fa1  0.9345
 ed fa2  0.8551
 ed fa3  0.3129
 ed fa4  0.1331
 ss sa0  0.9701
 ss sa1  0.9729
 ss sa2  0.9076
 ss sa3  0.3941
 ss sa4  0.7563
 sa0 mt0  1.1695
 zs0 za0  1.2657
 za0 mt0  1.3572
 mt0 mt1  0.9287
 vt mt0  0.7393
 vt mt1  0.7509
 vt mt2  1.3607
 vt mt3  1.0462
 vt mt4  0.7404
 cs0 mt0  1.0177
 zt0 mt0  0.6987
 mt1 mt2  0.2556
 fa1 za1  1.1214
 sa1 mt1  0.1608
 zs1 za1  0.8569
 za1 mt1  0.6759
 cs1 mt1  1.3154
 zt1 mt1  0.0831
 mt2 mt3  1.2163
 fa2 za2  0.0002
 sa2 mt2  1.3039
 zs2 za2  0.0829
 za2 mt2  0.7264
 cs2 mt2  0.3175
 zt2 mt2  0.1490
 mt3 mt4  0.6593
 fa3 za3  0.0465
 sa3 mt3  0.1525
 zs3 za3  0.9944
 za3 mt3  1.0367
 cs3 mt3  1.1562
 zt3 mt3  1.2245
 fa4 za4  0.0348
 sa4 mt4  0.9079
 zs4 za4  1.4347
 za4 mt4  1.4250
 cs4 mt4  1.4548
 zt4 mt4  0.3964
```

Figure C.2: Tetrad III – Linear Equation File - Graph Section from Example 3

The data for the Tetrad III examples presented in this thesis was generated by the "Monte" function with an argument of a linear equation model as presented in Figure C.2 and C.3. This data file is illustrated in Figures C.4, C.5 and C.6. In Figure C.4, the header section is presented which contains the network structure with some parameters and distributions that are specified in the command line. Figure C.5 details the structural model created for the data. Figure C.6 contains a portion of the data that was generated for Example 3. It should be noted that this data was normalized and weighted before use in the example.

/linearmodel

Variable	Dist.	Type	Parameters	
	fa	Normal	0	1
	fa0	Normal	0	1
	sd	Normal	0	1
	ed	Normal	0	1
	ss	Normal	0	1
	sa0	Normal	0	1
	zs0	Normal	0	1
	za0	Normal	0	1
	mt0	Normal	0	1
	vt	Normal	0	1
	cs0	Normal	0	1
	zt0	Normal	0	1
	mt1	Normal	0	1
	fa1	Normal	0	1
	sa1	Normal	0	1
	zs1	Normal	0	1
	za1	Normal	0	1
	cs1	Normal	0	1
	zt1	Normal	0	1
	mt2	Normal	0	1
	fa2	Normal	0	1
	sa2	Normal	0	1
	zs2	Normal	0	1
	za2	Normal	0	1
	cs2	Normal	0	1
	zt2	Normal	0	1
	mt3	Normal	0	1
	fa3	Normal	0	1
	sa3	Normal	0	1
	zs3	Normal	0	1
	za3	Normal	0	1
	cs3	Normal	0	1
	zt3	Normal	0	1
	mt4	Normal	0	1
	fa4	Normal	0	1
	sa4	Normal	0	1
	zs4	Normal	0	1
	za4	Normal	0	1
	cs4	Normal	0	1
	zt4	Normal	0	1

Figure C.3: Tetrad III – Linear Equation File – Linear Model Section Example 3

The Generating Model
Linear Structural Equation Model

Distribution over exogenous variables

Error term for	Distributional Family	Parameters	
fa	Normal	Mean: 0.0000	Variance: 1.0000
fa0	Normal	Mean: 0.0000	Variance: 1.0000
sd	Normal	Mean: 0.0000	Variance: 1.0000
ed	Normal	Mean: 0.0000	Variance: 1.0000
ss	Normal	Mean: 0.0000	Variance: 1.0000
sa0	Normal	Mean: 0.0000	Variance: 1.0000
zs0	Normal	Mean: 0.0000	Variance: 1.0000
za0	Normal	Mean: 0.0000	Variance: 1.0000
mt0	Normal	Mean: 0.0000	Variance: 1.0000
vt	Normal	Mean: 0.0000	Variance: 1.0000
cs0	Normal	Mean: 0.0000	Variance: 1.0000
zt0	Normal	Mean: 0.0000	Variance: 1.0000
mt1	Normal	Mean: 0.0000	Variance: 1.0000
fa1	Normal	Mean: 0.0000	Variance: 1.0000
sa1	Normal	Mean: 0.0000	Variance: 1.0000
zs1	Normal	Mean: 0.0000	Variance: 1.0000
za1	Normal	Mean: 0.0000	Variance: 1.0000
cs1	Normal	Mean: 0.0000	Variance: 1.0000
zt1	Normal	Mean: 0.0000	Variance: 1.0000
mt2	Normal	Mean: 0.0000	Variance: 1.0000
fa2	Normal	Mean: 0.0000	Variance: 1.0000
sa2	Normal	Mean: 0.0000	Variance: 1.0000
zs2	Normal	Mean: 0.0000	Variance: 1.0000
za2	Normal	Mean: 0.0000	Variance: 1.0000
cs2	Normal	Mean: 0.0000	Variance: 1.0000
zt2	Normal	Mean: 0.0000	Variance: 1.0000
mt3	Normal	Mean: 0.0000	Variance: 1.0000
fa3	Normal	Mean: 0.0000	Variance: 1.0000
sa3	Normal	Mean: 0.0000	Variance: 1.0000
zs3	Normal	Mean: 0.0000	Variance: 1.0000
za3	Normal	Mean: 0.0000	Variance: 1.0000
cs3	Normal	Mean: 0.0000	Variance: 1.0000
zt3	Normal	Mean: 0.0000	Variance: 1.0000
mt4	Normal	Mean: 0.0000	Variance: 1.0000
fa4	Normal	Mean: 0.0000	Variance: 1.0000
sa4	Normal	Mean: 0.0000	Variance: 1.0000
zs4	Normal	Mean: 0.0000	Variance: 1.0000
za4	Normal	Mean: 0.0000	Variance: 1.0000
cs4	Normal	Mean: 0.0000	Variance: 1.0000
zt4	Normal	Mean: 0.0000	Variance: 1.0000

Figure C.4: Tetrad III – Data File – Header Section from Example 3

Structural Equations
----
fa = e1
sd = e3
ed = e4
ss = e5
sa0 = 0.970ss + e6
zs0 = e7
vt = e10
cs0 = e11
zt0 = e12
fa1 = 1.475fa + 0.514sd + 0.934ed + e14
sa1 = 0.973ss + e15
zs1 = e16
za1 = 1.121fa1 + 0.857zs1 + e17
cs1 = e18
zt1 = e19
fa2 = 1.319fa + 0.990sd + 0.855ed + e21
sa2 = 0.908ss + e22
zs2 = e23
za2 = 0.000fa2 + 0.083zs2 + e24
cs2 = e25
zt2 = e26
fa3 = 0.703fa + 0.629sd + 0.313ed + e28
sa3 = 0.394ss + e29
zs3 = e30
za3 = 0.046fa3 + 0.994zs3 + e31
cs3 = e32
zt3 = e33
fa4 = 1.228fa + 0.478sd + 0.133ed + e35
sa4 = 0.756ss + e36
zs4 = e37
za4 = 0.035fa4 + 1.435zs4 + e38
cs4 = e39
zt4 = e40
fa0 = 1.478fa + 0.547sd + 0.925ed + e2
za0 = 0.026fa0 + 1.266zs0 + e8
mt0 = 1.169sa0 + 1.357za0 + 0.739vt + 1.018cs0 + 0.699zt0 + e9
mt1 = 0.929mt0 + 0.751vt + 0.161sa1 + 0.676za1 + 1.315cs1 + 0.083zt1 + e13
mt2 = 1.361vt + 0.255mt1 + 1.304sa2 + 0.726za2 + 0.317cs2 + 0.149zt2 + e20
mt3 = 1.046vt + 1.216mt2 + 0.152sa3 + 1.037za3 + 1.156cs3 + 1.224zt3 + e27
mt4 = 0.740vt + 0.659mt3 + 0.908sa4 + 1.425za4 + 1.455cs4 + 0.396zt4 + e34

Figure C.5: Tetrad III – Data File – Structure Equations from Example 3

```
/Continuousraw
200
fa      fa0     sd      ed      ss      sa0     zs0     za0     mt0     vt
cs0     zt0     mt1     fa1     sa1     zs1     za1     cs1     zt1     mt2
fa2     sa2     zs2     za2     cs2     zt2     mt3     fa3     sa3     zs3
za3     cs3     zt3     mt4     fa4     sa4     zs4     za4     cs4     zt4
 0.9850  0.6570 -1.1163  0.8115  1.2959  0.4334 -1.0452 -1.5883 -1.8794  0.6009
-1.3897 -0.2841  2.3312  0.3495  1.9547 -0.5993 -0.5342  2.0853  1.2062  1.2310
 1.1331  0.4423  0.3353 -0.3564 -0.6165 -0.0925  0.9318  1.7588  0.7586  0.6118
 0.4502 -0.0763 -1.8141  3.4874  1.8192  1.6252 -0.6512 -0.7385  1.0983  1.0836
 0.8556  2.6106  0.9462  1.1997 -1.3288 -1.9751 -0.1815 -0.0892 -0.8692  0.4348
 0.2480  1.0844 -1.2820  4.1214 -2.8490 -0.5212  1.7082 -1.5212 -0.8999 -3.2579
 3.0700 -2.1967 -0.5314 -0.7281  1.2036 -1.0166 -5.7608  2.1596 -0.1928  0.7767
 0.9636 -0.8578 -0.7958 -6.5467  1.8575 -0.7858 -0.3981 -2.0652  0.4321  0.5993
 0.6781 -0.6582 -0.4098 -1.0634  1.5611  0.4565 -0.2150  0.6303  3.4299  0.4564
 0.8842  0.1093  6.1988 -0.3237  1.5091  0.6014  1.2090 -0.1065  0.2413  0.0479
-1.8822  0.4643  0.8316 -2.7024 -0.5041 -1.2233  1.3935 -1.2675  0.2715  0.4494
 0.0124  0.3283  1.4448  2.6459  0.6304  1.7335  1.8096  0.8678 -1.2502 -0.3942
 0.1353 -1.2391  0.1466 -1.2846 -1.2153 -0.8452  0.3450  1.7960  1.4220 -1.0511
 0.9539 -1.4151 -0.0389 -2.2873 -1.6479 -0.7030 -3.9477  1.2989 -0.5116 -1.5824
```

Figure C.6: Tetrad III – Data File – Raw Data Section (partial) from Example 3

Bibliography

[AD98] Angehrn, Albert A. & Dutta, Soumitra. "Case-Base Decision Support." *Communications of the ACM* 41, 5 (1998): 157 - 165.

[Aha98] Aha, David W. "The Omnipresence of Case-Based Reasoning in Science and Application." *Knowledge Based Systems* (Berlin Heidelberg, Germany) 11 (1998): 261 - 273.

[AL98] Aamodt, Agnar & Langesth, Helge. "Integrating Bayesian Networks into Knowledge-Intensive CBR." *AAAI Workshop on Case-Based Reasoning Integrations*, 1998.

[All84] Allen, James F. "Towards a General Theory of Time." *Artificial Intelligence* (North-Holland) 23 (1984): 123-154.

[AP94] Aamodt, Agnar & Plaza, Enric. "Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches." *AICom - Artificial Intelligence Communications, IOS Press* 7, 7 (1994): 39 - 59.

[BB96] Brassard, Gilles & Bratley Paul. *Fundamentals of Algorithmics*. Englewood Cliffs, N.J.: Prentice-Hall, 1996.

[BCSV04] Bandini, Stefania; Colombo, Ettore; Sartori, Fabio & Vizzari, Giuseppe. "Case-Based Reasoning and Production Process Design: The Case of P-Truck Curing." *Advances in Case-Based Reasoning: 7th European Conference on Case-Based Reasoning (ECCBR 2004) - Lecture Notes in Artificial Intelligence 3155*, Funk, Peter & Gonzales Calero, Pedro A..

Berlin Heidelberg, Germany, Springer-Verlag. 2004. Madrid, Spain - August 30th - September 2nd, 2004.

[Ber01] Bergmann, Ralph. "Highlights of the European INRECA Project." *Case-Based Reasoning Research and Development, 4th International Conference on Case-Based Reasoning (ICCBR 2001) - Lecture Notes in Computer Science 2080*, Aha, David W. & Watson, Ian. Berlin Heidelberg, Germany, Springer-Verlag. 2001. Vancouver, BC, Canada - July 30th - August 2nd, 2001.

[BH95] Breese, John S. & Heckerman, David. "Decision-Theoretic Case-Based Reasoning." *Technical Report MSR-TR_95-03*, Redmond, WA, Microsoft Research, Advanced Technology Division, Microsoft Corporation. 8/1995.

[BM01] Bandini, Stefania & Manzoni, Sara. "CBR Adaptation for Chemical Formulation." *Case-Based Reasoning Research and Development, 4th International Conference on Case-Based Reasoning (ICCBR 2001) - Lecture Notes in Computer Science 2080*, Aha, David W. & Watson, Ian. Berlin Heidelberg, Germany, Springer-Verlag. 2001. Vancouver, BC, Canada - July 30th - August 2nd, 2001.

[Bri05] Bridge, Derek. "The Virtue of Reward: Performance, Reinforcement and Discovery in Case-based Reasoning" *Case-Based Reasoning Research and Development: The 6th International Conference on Case-Based Reasoning (ICCBR 2005) - Lecture Notes in Artificial Intelligence 3620*, Munoz-Avila Hector & Ricci, Francesco. Berlin Heidelberg, Germany, Springer-Verlag. 2005. Chicago, IL, USA.

[BSS02] Bandini, Stefania; Manzoni, Sara & Simone, Carla. "Tuning Production Processes through a Case-Based Reasoning Approach." *Advances in Case-Based Reasoning - 6th European Conference on Case-Based Reasoning (ECCBR 2002) - Lecture Notes in Computer Science 2416*, Craw, Susan & Preece, Alun. Berlin Heidelberg, Germany, Springer-Verlag. 2002. Aberdeen, Scotland, UK - September, 4th - 7th, 2002.

[BV98] Barrientos, M. A. & Vargas J. E. "A Framework for the Analysis of Dynamic Processes Based on Bayesian Networks and Case-Based Reasoning." *Expert Systems with Applications* (Elsevier Science Ltd.) 15 (1998): 287 - 294.

[CBL97] Cheng, Jie; Bell David A. & Lui, Weiru. "An Algorithm for Bayesian Network Construction from Data." *Proceedings of the 6th International Workshop on Artificial Intelligence and Statistics AI&STAT97, 1997.*

[Cha00] Charniak, Eugene. "Bayesian Networks without Tears." *AI Magazine*, 2000, 50 - 63.

[Che01] Cheetham, William. "Benefits of Case-Based Reasoning in Color Matching." *Case-Based Reasoning Research and Development, 4th International Conference on Case-Based Reasoning (ICCBR 2001) - Lecture Notes in Computer Science 2080*, Aha, David W. & Watson, Ian. Berlin Heidelberg, Germany, Springer-Verlag. 2001. Vancouver, BC, Canada - July 30th - August 2nd, 2001.

[CHM04] Chickering, David Maxwell; Heckerman, David & Meek Christopher. "Large-Sample Learning of Bayesian Networks is NP-Hard." *Journal of Machine Learning Research 5*, 2004, 1287 - 1330

[CJR01] Craw, Susan; Jarmulak, Jacek & Rowe, Ray. "Learning and Applying Case-Based Adaptation Knowledge." *Case-Based Reasoning Research and Development, 4th International Conference on Case-Based Reasoning (ICCBR 2001) - Lecture Notes in Computer Science 2080*, Aha, David W. & Watson, Ian. Berlin Heidelberg, Germany, Springer-Verlag. 2001. Vancouver, BC, Canada - July 30th - August 2nd, 2001.

[Coo90] Cooper, G. F. "The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks." *Artificial Intelligence* 42 (1990): 393-405.

[CVG01] Cheetham, William; Varma, Anil & Goebel, Kai. "Case-Based Reasoning at General Electric." *The Fourteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2001)*, Russell, Ingrid & Kolen, John F.., American Association for Artificial Intelligence. 2001. Key West, Florida - May 21st - 23rd, 2001.

[Dat98] DataPaq Inc. "Oven Tracker User Manual for use with the Insight Software Issue 1", DataPaq Part No. MA5000, Wilmington, MA, 1998

[Din98] Dingsoyr, Tirgeir. "Retrieval of Cases by Using a Bayesian Network." *AAAI Workshop on Case-Based Reasoning Integrations*, 1998.

[DK98] Dean, Thomas & Kanazawa, Keiji. "A model for reasoning about persistence and causation" *Computational Intelligence, Volume 3, Issue 3*, Blackwell Publishers Inc, Cambridge MA, August 1990, Pages 142 – 150.

[FHKR95] Forbes, Jeff; Huang, Tim; Kanazawa, Keiji & Russel, Stuart. "The BATmobile Towards a Bayesian Automated Taxi." In *Proceedings*

Fourteenth International Joint Conference on Artificial Intelligence, Montreal, PQ, 1995.

[FLMN99] Fuchs, Beatrice; Lieber, Jean; Mille, Alain & Napoli Amedeo. "Towards a Unified Theory of Adaptation in Case-Based Reasoning." *The 3rd International Conference on Case-Based Reasoning and Development (ICCBR 99) - Lecture Notes in Computer Science 1650*, Althoff, Klaus-Dieter; Bergmann Ralph & Branting, Karl. Berlin Heidelberg, Germany, Springer-Verlag. 1999. Seeon Monostary, Germany - July 27th - July 30th, 1999.

[Fox00] Fox, Susan Eileen. "A Unified CBR Architecture for Robot Navigation." *Advances in Case-Based Reasoning: 5th European Workshop (EWCBR 2000) - Lecture Notes in Artificial Intelligence 1898*, Blanziere, E & Portinale, L.. Berlin Heidelberg, Germany, Springer-Verlag. 2000. Trento, Italy - September, 2000.

[Gom04] Gomes, Paulo. "Software Design Retrieval Using Bayesian Networks and Word Net." *Advances in Case-Based Reasoning: The 7th European Conference on Case-Based Reasoning (ECCBR 2004) - Lecture Notes in Artificial Intelligence 3155*, Funk, Peter & Gonzales Calero, Pedro A.. Berlin Heidelberg, Germany, Springer-Verlag. 2004. Madrid, Spain - August 30th - September 2nd, 2004.

[GVP90] Geiger, Dan; Verma, Thomas & Pearl Judea. "d-separation: From Theorems to Algorithms." *Uncertainty in Artificial Intelligence 5,* Henrion M., Schachter, R.D, Kanal, L.N. & Lemmer, J.K. Elservier Science Publishers, North Holland, 1990

[HB02] Hejazi, Mohmoudreza & Badie, Kambiz. "A New Approach to Solution Adaptation and Its Applications for Design Purposes." *Advances in Case-Based Reasoning: 6th European Conference on Case-Based Reasoning (ECCBR 2002) - Lecture Notes in Computer Science 2416*, Craw, Susan & Preece, Alun. Berlin Heidelberg, Germany, Springer-Verlag. 2002. Aberdeen, Scotland, UK - September, 4th - 7th, 2002.

[HBSA01] Hollingworth, Jarrod; Butterfield, Dan; Swart, Bob & Allsop, Jamie. *C++ Builder 5 - Developer's Guide.* Indianapolis, IN: SAMS Publishing, 2001.

[HT95] Hinkle, David & Toomey, Christopher. "Applying Case-Based Reasoning to Manufacturing." *Artificial Intelligence* (American Association for Artificial Intelligence) Spring (1995): 65 - 73.

[Int03] Intel. *Probabilistic Network Language: User Guide and Reference Manual.* Intel Corporation, 2003.

[JAS02] Jaere, Martha Dorum; Aamodt, Agnar & Skalle, Pal. "Representing Temporal Knowledge for Case-Based Prediction." *Advances in Case-Based Reasoning: The 6th European Conference on Case-Based Reasoning (ECCBR 2002) - Lecture Notes in Computer Science 2416*, Craw, Susan & Preece, Alun. Berlin Heidelberg, Germany, Springer-Verlag. 2002. Aberdeen, Scotland, UK - September, 4th - 7th, 2002.

[Jen99] Jenson, Finn V. "Lecture Notes on Bayesian Networks and Influence Diagrams.," Aalborg University, Department of Computer Science. December, 1999.

[Jen02] Jenson, Finn V. *Bayesian Networks and Decision Graphs.* New York, NY: Springer-Verlag, 2001.

[Kja95] Kjaerulff, Uffe. "dHugin: A Computational System for Dynamic Time-Sliced Bayesian Networks." *International Journal of Forecasting, Special Issue on Probability Forecasting* (Elsevier Science) II, 1 (March, 1995): 89 - 111.

[Kol93] Kolodner, Janet. *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1993.

[LAV94] Lekkas, G. P.; Arouris, N. M. & Viras, L.L. "Case-Based Reasoning in Environmental Monitoring Applications." *Applied Artificial Intelligence* 8, 3 (1994): 349-376.

[LD97] Lin, Yan & Drusdsel, Mark J. "Computational Advantages of Relevance Reasoning in Bayesian Belief Networks." *In Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, San Francisco, CA, Morgan Kaufmann Publishers Inc., 1997

[Lea03] Leake, David B. *Case-Based Reasoning: Experiences, Lessons & Future Directions*. Menlo Park, CA and Cambridge, MA: American Association for Artificial Intelligence and The MIT Press, 1996.

[Lou03] Loudon, Kyle. *C++ Pocket Reference*. Sebastopop, CA: O'Reilly& Associates, 2003.

[LRSC01] Leiserson, Charles E.; Rivest, Ronald L.; Stien, Clifford & Cormen, Thomas H. *Introduction to Algorithms, Second Edition*. The MIT Press, 2001.

[LS03] Laskano, Elena & Sierra, Basilio. "Bayes-Nearest: A New Hybrid Classifier Combining Bayesian Network and Distance Based Algorithms." *Progress in Artificial Intelligence: The 11th Portuguese Conference of*

Artificial Intelligence (EPIA 2003) - Lecture Notes in Artificial Intelligence 2902, Pires Moura, Fernando & Abreu, Salvado. Berlin Heidelberg, Germany, Springer-Verlag. 2003. Beja, Portugal - December 4th 2003.

[Man01] Mantaras, Ramon Lopez de. "Case-Based Reasoning." *Machine Learning and Its Application* (Berlin Heidelberg, Germany) Lecture Notes in Computer Science 2049 (2001): 127 - 145.

[MBFCM90] Miller, G.; Beckwith, R.; Fellbaum, C.; Cross, D. & Miller, K.J. "Introduction to Wordnet: An On-Line Lexical Database." *International Journal of Lexicography* 3 (1990): 235-244.

[MCMC05] Sanchez-Marre, Miquel; Cortes, Ulises; Martinez, Montse; & Comas, Joaquim & Rodriquez-Roda. "An Approach for Temporal Case-Based Reasoning: Episode-Based Reasoning." *Case-Based Reasoning Research and Development: The 6th International Conference on Case-Based Reasoning (ICCBR 2005) - Lecture Notes in Artificial Intelligence 3620*, Munoz-Avila Hector & Ricci, Francesco. Berlin Heidelberg, Germany, Springer-Verlag. 2005. Chicago, IL, USA.

[Mur02] Murphy, Kevin. "Dynamic Bayesian Networks: Representation, Inference and Learning." PhD diss., University of California Berkley. 2002.

[Pea88] Pearl, Judea. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufman, 1988.

[Pea00] Pearl, Judea. Causality: Models, Reasoning, and Inference. New York, NY: The Press Syndicate of the University of Cambridge, 2000.

[Ray99] Raynor, William. *The International Dictionary of Artificial Intelligence.* Chicago, IL: Glenlake Publishing Company LTD, 1999.

[RH02] Riordan, Denis & Hansen, Bjarne. "A Fuzzy Case-Based System for Weather Prediction." *Engineering Intelligent Systems* 3 (2002): 139 - 145.

[RI01] Roth-Berghofer, Thomas & Iglezakis, Ioannis. "Six Steps in Case-Based Reasoning: Towards a Maintenance Methodology for Case-Based Reasoning Systems." *The 9th German Workshop on Case-Based Reasoning (GWCBR 2001)*, 2001. Baden-Baden, Germany.

[RN03] Russel, Stuart & Norvig, Peter. *Artificial Intelligence: A Modern Approach, Second Edition.* Upper Saddle River, NJ: Pearson Education Inc., 2003.

[RR01] Roth-Berghofer, Thomas & Reinartz, Thomas. "MaMa: A Maintenance Manual for a Case-Based Reasoning System." *Case-Based Reasoning Research and Development: The 4th International Conference on Case-Based Reasoning (ICCBR 2001) - Lecture Notes in Computer Science 2080*, Aha, David W. & Watson, Ian. Berlin Heidelberg, Germany, Springer-Verlag. 2001. Vancouver, BC, Canada: July 30th - August 2nd, 2001.

[SA00] Schiaffino, Silvia N. & Amandi, Analfa. "User Profiling with Case-Based Reasoning and Bayesian Networks." *IBERAMIA-SIBS 2000 Open Discussion Track,* 2000.

[Sai03] Sai Holachima, Satya. *C++ Builder 6: Developers Guide.* Plano, TX: Wordware Publishing Inc., 2003.

[SG02] Schmidt, R & Gierl L. "Case-Based Reasoning for Prognosis of Threatening Influenza Waves." In *Advances in Data Mining*, Perner P.. 99 - 108. Berlin:2002.

[Sha86] Shachter, Ross D. "Evaluating Influence Diagrams." *Operations Research* February, 1986 (February, 1986).

[SSGM94] Scheines R.; Spirites P.; Glymour C. & Meek C. *Tetrad II: Tools for Causal Modeling*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1994.

[TKM96] Tirri, Henry; Kontkanen, Petri & Myllymaki, Petri. "A Bayesian Framework for Case-Based Reasoning." *Advances in Case-Based Reasoning - The 3rd European Workshop on Case-Based Reasoning 1996 (EWCBR96) - Lecture Notes in Computer Science 1168*, Smyth, B & Cunningham, P. Berlin Heidelberg, Germany, Springer-Verlag. 1996. Lausanne, Switzerland: November 14th - 16th, 1996.

[TT05] Tighe, Christopher A. & Tawfik, Ahmed. "Using Causal Knowledge to in Case Based Reasoning about Dynamic Processes" *International Conference on Case-based Reasoning: Workshop on Similarities, Processes and Workflows*, Chicago, IL, August 2005

[Wat97] Watson, Ian. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. San Francisco, California: Morgan Kaufmann Publishers, Inc., 1997.

[Wat99] Watson, Ian. "Case-Based Reasoning Is a Methodology Not a Technology." *Knowledge Based Systems* (Berlin Heidelberg, Germany) 12 (1999): 303 - 308.

[Whi05] White, Larry. "Using CBR to Build Dealer Satisfaction and Get a Strong ROI." *6th International Conference of Case-Based Reasoning: Industry Day Proceedings*, Goker, Mehmet H. & Cheetham, William. Chicago, Ill, August, 2005. Pages 35 - 41.

[WOB98] Wiebe, Janyce; OHara, Tom & Bruce, Rebecca. "Constructing Bayesian Networks from Word Net for Word-Sense Disambiguation: Representational and Processing Issues." *Conference on the Use of Word Net in Natural Language Processing Systems*, Somerset, NJ, Association for Computational Linguistics. 1998. Pages 23 - 30.

[Xia96] Xiang, Y. "A Probabilistic Framework for Cooperative Multi-Agent Distributed Interpretation and Optimization of Communication." *Artificial Intelligence* 87, 1-2 (1996): 295-342.

[Xia03] Xiang, Y. "Comparison of Multiagent Inference Methods in Multiply Sectioned Bayesian Networks." *International Journal Approximate Reasoning* 33, 3 (2003): 235-254.

[YL05] Yehezkel, Raan & Lerner, Boaz. "Recursive Autonomy Identification for Bayesian Network Structure Learning." *The 10th International Workshop on Artificial Intelligence & Statistics, AISTATS 2005,* January 6-8, 2005. Barbados, pages 429 – 436.

Vita Auctoris

Christopher Allan Tighe was born in Windsor, Ontario Canada on October 30, 1964. He graduated from Riverside Secondary School in Windsor, Ontario and enrolled at the University of Windsor. In 1989 he graduated with B.C.S. degree from the School of Computer Science. Upon graduation Chris pursued a career in the automotive paint business with PPG Canada Inc., a major paint supplier to automotive corporations. In 1999 he joined DaimlerChrysler Canada as the supervisor of the Automotive Coatings Research Facility. In 2001, Chris enrolled in the School of Computer Science at the University of Windsor to pursue a Master of Science degree. Chris' research interest includes intelligent tutoring systems and several other areas of artificial intelligence including case-based reasoning, uncertainty reasoning, temporal reasoning and probabilistic reasoning.