Using CBR to Improve the Usability of Numerical Models

By

Fei Ling Woon

A Thesis submitted in partial fulfilment of the requirements for the Degree of Doctor of Philosophy in

School of Computing and Mathematical Sciences

University of Greenwich, London, UK

October, 2005

Sponsor:

Tunku Abdul Rahman College
University of Greenwich

TO

MY HUSBAND AND FAMILY

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to Professor Brian Knight for his invaluable advice and enthusiasm in the course of this research. Professor Knight is a very skillful, intelligent and dedicated researcher. Working with him has been a remarkable experience for me. He has always been an encouragement to me during my research. Without his supervision this work this work would not have been accomplished.

I would also like to thank my second supervisor, Dr. Miltos Petridis for his advice and patience. Dr. Petridis is an expert in software engineering. He has advised me on the technical aspects of building a successful Case-Based Reasoning (CBR) tool.

During my research, I have the privilege to work with a number of colleagues. They have been a great help to me by offering valuable technical advice in various engineering fields. They are: Dr. Mayur Patel, Dr. Pierre Chapelle, Dr. Fuchen Jia, Dr. Hua Lu and Professor Koulis Pericleous.

Furthermore, I am very grateful to my family and friends who has been a source of motivation to me. Particularly, to my father who has worked hard to make a living so that I have the opportunity to develop my potentials; to my husband, who has patiently read through every page of this thesis to correct any language and technical mistakes.

Finally, I would like to thank Tunku Abdul Rahman College (TARC) and the University of Greenwich for supporting me financially during my PhD study.

ABSTRACT

In this thesis we show that CBR systems can be constructed from numerical models, so as to improve their usability. It is shown that CBR models may be queried in a flexible manner, and that the user may formulate queries consisting of constraints over both "input" and "output" variables of the numerical model. It is also shown that the constraints may be formulated using either nominal or continuous variables. A generalization of the CBR retrieval process to include constraints over unified "input-output" space is formulated as a framework for the method.

The method is illustrated with practical engineering models: the pneumatic conveyor problem and the projectile problem. Comparisons are made on usability of CBR and numerical models for specific problems. It is shown that CBR models can answer questions difficult or impossible to formulate using numerical models, and that CBR models can be faster.

The thesis also addresses a latent problem with the general method, which is of importance generally. This is to do with interpolation over nominal values in unified space. A novel method is proposed for interpolation over nominal values, termed Generalised Shepard Nearest Neighbour method (GSNN). GSNN can utilise distance metrics defined on the solution space of a CBR system.

The properties and advantages of GSNN are examined in the thesis. A comparison is made with other CBR retrieval methods, using several examples, including the travel domain case base. It is shown that GSNN can out-perform conventional nearest neighbour methods. It is shown that GSNN has advantages in that it can find solutions not in the case base and it can find solutions not in the retrieval set. It is also shown that the performance of GSNN can be improved further by using it in conjunction with a diversity algorithm. The merit of using GSNN as a case selection component is examined, and it is shown that it can give good results in sparse case bases.

Finally the thesis concludes with a survey of numerical models where CBR construction can be useful, and where benefits can be expected.

CONTENTS

Part II Core Idea of the Research

Part III Case Studies

Part IV Conclusion and Future Work

Part V References and Appendices

PART I

THE RESEARCH PROBLEM

Chapter 1

Introduction

1.1 BACKGROUND

Numerical models of physical process are important tools used in engineering fields to predict the behaviour and the impact of physical elements, and hence improve their performance accordingly. Over the years, computational models have led to cost-effectiveness in the designs of physical systems. Various numerical models have been developed, for example, the simulation of a pneumatic conveyor, fire spread within buildings, evacuation in emergency, heat transfer, damage detection, etc. In particular, pneumatic conveying is an important transportation technology widely used in conveying solid bulk industry. Engineers of pneumatic conveying are concerned with the design of the conveyor [Kalman, 2000; Chapelle *et al.*, 2003] in order to reduce the damage to particles (e.g., sugar, rice, tea leaf).

These models calculate the output from the input parameters assigned by engineers. However, it is very often that engineers face problems that are queried in an inverse fashion. They may have a list of anticipated outputs and seek the right input parameters that will yield the expected outputs. To solve these inverse problems without solving the original problems directly will require a different computational model and often proves to be extremely difficult, if not impossible, to construct. Time and again, the only option left for the engineers to find the input parameters for anticipated outputs is the approach of trial and error based on their instinct or experiences.

In this study, the author investigates the feasibility of using Case-Based Reasoning techniques to improve the usability of numerical models, particularly in helping engineers to solve inverse and constraint problems.

1.2 RESEARCH QUESTIONS

1.2.1 Primary Question

The main objective of this study is to answer the following research question:

"Can Case-Based Reasoning methods be used to improve the usability of numerical models?"

Case-Based Reasoning (CBR) techniques have been proved to be an effective problems-solving tool over the recent decades. A number of researchers have used CBR and machine learning techniques to improve the usability of numerical models. These include: [Cheetham and Graf, 1997; Cheetham, 2001; Schwabacher *et al.*, 1998; Kalapanidas and Nikolaos, 2001]. Cheetham gives an account of a CBR tool developed to help users to select a subset of the allowable colourants for colour matching in plastics. The CBR tool was shown to be cost saving and to increase the colour matcher productivity. There may be advantages in using CBR to support engineers in the decision-making process, particularly for solving inverse and constraint problems in the numerical domains. This primary objective of this study arose from discussions with numerical modelers who found that mathematical models are often not very user-friendly in terms of obtaining the input parameters of a mathematical model to produce its expected outputs. Unlike database models, there is no equivalent of Structured Query Language (SQL) for querying numerical models, and this result in need of skilled programming in order to adapt a model for a particular use. Due to the characteristics and easy use of a CBR system, an intuitive idea was proposed in this study that a CBR system could be produced from a numerical model and that this system could be queried in a more flexible way to help engineers who endeavor to solve an inverse engineering application.

The question was taken as the main driver for the research in this study. The main impetus was to construct a general method for deriving useful CBR systems from numerical models, and to assess their usefulness as engineering aids. Although the methods are of general significance, they were developed with reference to two specific complete examples, to try to ensure that the important detail was not overlooked.

In the course of this study derived by the primary objective stated at the very beginning of this section, some interesting subsidiary problems were encountered. One of these (the nominal values interpolation problem) blossomed into a substantial question in its own right, and its solution occupies a major part of the thesis. Indeed the solution of the nominal values interpolation problem also gave rise to a new method of case base reduction, described in Chapter 7.

1.2.2 Subsidiary Questions

Some subsidiary research questions arose in the course of this investigation. These were:

1.2.2.1 Can the standard CBR models be generalized to a unified (problem:solution) space to allow flexible query modes?

This question arises from the need of engineers to define constraints dynamically during a consultation with the model of interest. In many cases they need to define a constraint over both "inputs" and "outputs". The usability of a numerical model is intimately connected with its queryability. The database model can be queried with SQL by selecting any set of variables to be inputs. Comparatively, in the standard CBR models, a case is represented as a <problem, solution> pair. The problem space and the solution space are treated as separate. To solve inverse or constraint problems faced by engineers such as the projectile model problem and the pneumatic conveyor problem described in Section 2.2 and 2.3, it is essential to investigate whether the standard CBR models can be generalized to a unified (problem:solution) space to allow flexible query modes so that engineers can define constraints in both spaces. In order to discuss the usability of CBR systems regarding engineering constraints we first need to formalize the concept of a unified space.

The answer to this question is presented in Chapter 4, Unified Problem:Solution Space in Case-Based Reasoning, in which a generalization of the standard CBR retrieval method that integrates the problem space and the solution space into a single query space is proposed. The retrieval method is proposed by means of the concept of nearest neighbours to a constraint region. In the method, a target may be expressed as a constraint region, *R*, specified by means of constraints over the unified space. It is shown that the standard CBR retrieval is a special case of this more general model. Its advantages are demonstrated by a variety of illustrative examples.

1.2.2.2 Can distance weighted interpolation be generalized to a unified space, particularly with regard to nominal values?

Interpolation is generally a good adaptation method, particularly in the real domain. However, in the case of mixed domains where there are continuous and nominal values such as discrete bend types and bend angles in the pneumatic conveyor problem [Woon *et al.*, 2005], it is doubtful if the same method can be applied to nominal values. Wilson and Martinez (1997) use Stanfill and Waltz's Value Difference Metric (1986) as the basis for interpolation. However, their method is not able to cope with solutions not in the case base. Therefore, further study is required to contrive a method for case based adaptation using interpolation over nominal values for inverse or mixed-constraint problems, where often a solution is not present in the case base.

Shepard's interpolation method [Shepard, 1968] is the basis of many interpolation methods [Mitchell, 1997; Franke and Nielson, 1980]. We take this as a starting point for research into the possibility of an interpolation method for nominal values. *Chapter 5 Interpolation over Nominal Values* presents a generalization of the Shepard's method for the inclusion of a distance metric defined in the solution space. The method is termed as Generalized Shepard Nearest Neighbour (GSNN) [Knight and Woon, 2003b, Knight and Woon, 2004b]. GSNN has been shown to out-perform conventional nearest neighbour methods on the Iris data set [Fisher, 1936], a simulated nominal value test problem and a benchmark case base from the travel domain [Lenz *et al.*, 1996]. The predictive power is shown to improve in efficiency when used in conjunction with a diverse retrieval algorithm.

1.2.2.3 Can the error function developed in the nominal value interpolation be useful in case selection?

This question arose as an outcome of the work on nominal values, and is not wholly pertinent to the core of this study. The generation of a case base from a numerical model raises an interesting question as to how many and which cases need to be generated. This is intimately connected with case selection techniques, which address the same problem. The error function developed for the nominal value interpolation has several potential advantages for case selection. Question 1.2.2.3 was a by-product of an investigation into how effective the error function is in case selection.

High dimensionality may make the CBR solution infeasible. In the pneumatic conveyor problem example (see Section 2.3) there are 7 degrees of freedom, and in order to cover the domain in reasonable detail, say 10 points in each dimension, it would require 10^7 cases in total. One approach to this problem is to use case selection techniques such as [Salamo and Golobardes, 2002; Wilson and Martinez, 2000; Kibler and Aha, 1987; Yang and Zhu, 2001; Symth and McKenna, 1998, 1999] to keep the case base small while trying to maintain the level of solution prediction accuracy.

In the domain of numerical models, there is a great deal of regularity in the model. One would expect fine detail to be well represented by some adaptive process such as interpolation. Interpolation has been proven effective in the real domain, also in nominal and mixed domains (based on the results obtained from using GSNN). There is the possibility to produce a relatively small efficient case base by using case selection techniques and interpolation techniques.

With this reasoning, a case selection method with the inclusion of a solution space metric is proposed. The method embedded GSNN in decremental reduction algorithms such as the Shrink algorithm [Kibler and Aha, 1987]. Results [Woon *et al.*, 2003a] show that the method can produce a sparse case base with good predictive power in the Iris data set and the pneumatic conveyor problem.

1.2.2.4 What sort of models can the method be used on?

The number of areas where numerical models are used is huge. The focus in this research is on the queryability of numerical models. Therefore, both the inputs and outputs of a numerical model play a significant role. The input and output data types and the degree of freedom have a strong effect on the performance of a CBR model. In general, these inputs and outputs can be real-valued and nominal-valued attributes. In this study, a survey of the scope of the method is presented in the concluding chapter. This points to engineering domains where the method can be deployed.

1.3 OBJECTIVES

The following is a list of objectives of this research work:

- To present a basis for the generalized version of CBR model based on a unified (problem:solution) space to allow flexible query mode
- To present a basis for the generalization of distance-weighted interpolation method to unified space, particularly with regard to nominal values
- To propose a case selection method to produce a relatively small efficient case base
- To implement and assess the proposed CBR approach in two specific numerical problems:
  (i) The Projectile model
  (ii) The Pneumatic Conveyor Problem
- To survey scope of the method

1.4 RESEARCH METHODOLOGY

This section is devoted to a delineation of the research methodology taken by this study. The rationale for each part of the research work is outlined, and its role in supporting the core of the study is explained. Fig. 1-1 shows a graphic of the research story, breaking it down into 4 stages.

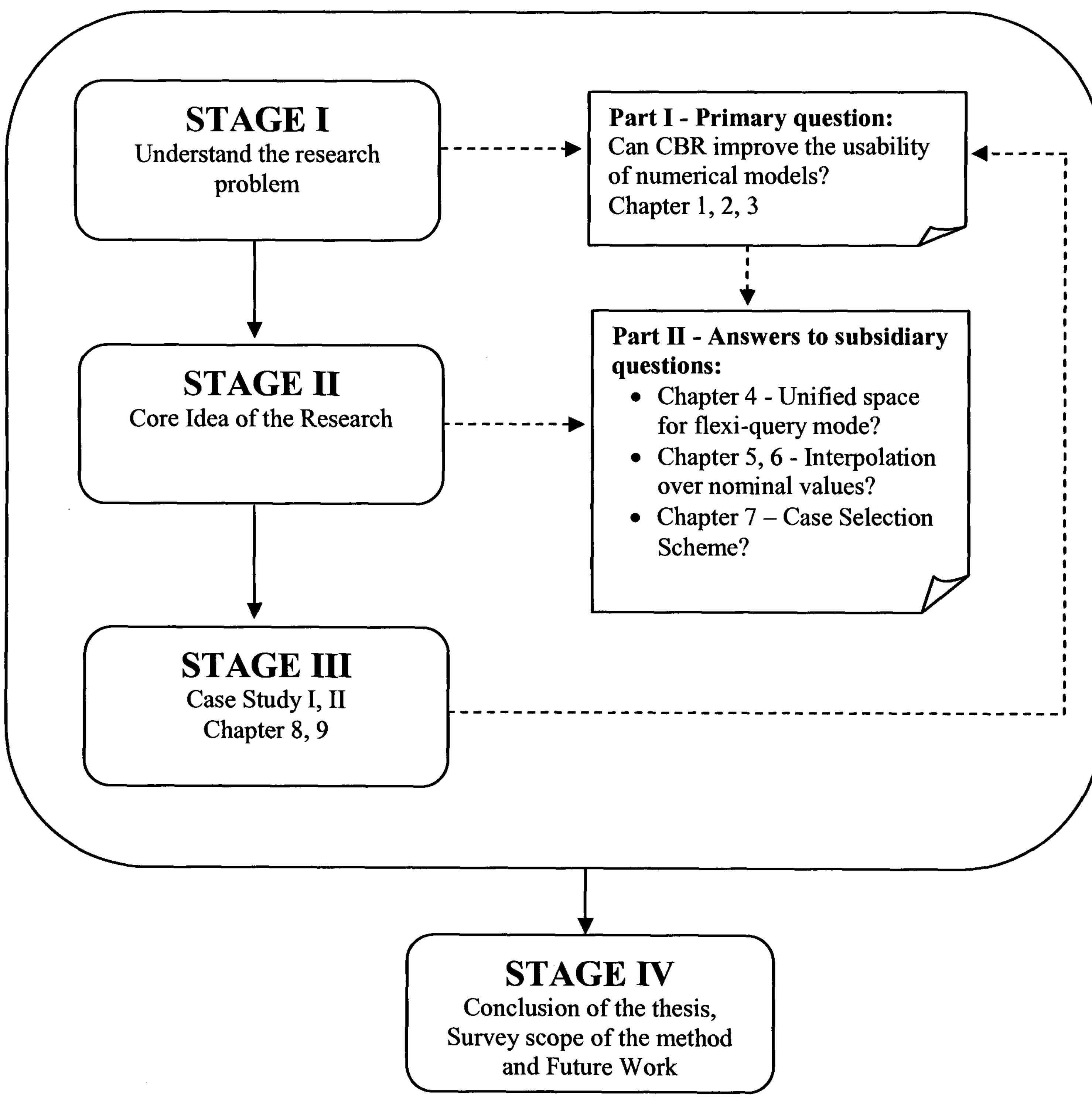


Figure 1-1. Outline of the Research Methodology

At the outset of this study, a brief outline of the research problem based on the subject area and the primary question (i.e., Section 1.2.1) were proposed. The project was divided into four stages as follows (Fig 1-1 is produced as a guide to this study.):

1. Stage I: Understanding the Research Problem
2. Stage II: Core Idea of the Research
3. Stage III: Case Studies
4. Stage IV: Conclusion of the Thesis and Future Work

1.4.1 Stage I

The first task was to find out the primary concern of this research and then to go through past research relevant to this research. The objective was to gather useful background information and to ensure that this project is not repeating the work of others but a contribution to knowledge.

In the meantime, it was useful to look at some examples of numerical models for more concrete definitions of the research problem. The objectives were to identify the advantages of using CBR for numerical models and to consider the problems that needed to be dealt with in the CBR approach. The project began by studying a simple projectile problem. In the course of investigation, it was discovered that in order to answer the primary question it was essential to break down the primary problem (i.e., Section 1.2.1) into several sub-problems. Consequently, a few subsidiary questions vital to this project had been identified.

1.4.2 Stage II

Once the main and side problems had been briefly defined, the next step was to tackle each of the sub-problems individually and then to examine if they could be integrated to solve the whole problem. To do this, it was essential to review relevant literature to ensure that the methods proposed to solve these sub-problems are original and to search possible existing research that might shed lights on these problems.

In the engineering domains, there may be a number of solutions to a given query where outputs are specified. Cases with close outputs may have very different inputs. Hence, retrieved solutions may not necessarily be adaptable. This may cause difficulties in adaptation as these cases are not *"conformable" for interpolation.* One of the major tasks of this study was to investigate the feasibility of extending standard CBR models to a unified (problem:solution) space to allow flexible query modes. This will affect the case representation and the metric defined on the query space. The idea was to generalize standard CBR models to a unified space so that engineers could specify inverse or constraint problems in a flexible manner, when necessary adaptation is then performed on cases that are close in the unified space. The generalized method has been examined in the projectile model and the pneumatic conveyor problem for illustration.

Another task was to identify a way to handle nominal values in constraints and interpolation. In the pneumatic conveyor problem (see Chapter 2) nominal values may appear in a physical system and it was doubtful that interpolation methods could be useful and effective for handling nominal values in adaptation [Chatterjee and Campbell, 1993; Wilson and Martinez 1997]. This shortcoming may cause difficulties in an engineering application in which such a constraint over the outputs is imposed that the inputs (e.g., bend type) of the application is not yet present in the case base or in the retrieved set.

We took Shepard's interpolation method [Shepard, 1968] to be the starting point for the possibility of an interpolation method for nominal values. Shepard's method was first expressed as a minimization problem. The interpolation equations are the conditions for an error function to be minimized. It was noticed that in this form the error function is only dependent on the (Euclidean) distances. Hence it was easy to generalize the error function to depend on distance metrics. In this way the minimization of the generalized error function gives generalized interpolation formulae. It is termed as Generalized Shepard Nearest Neighbour (GSNN) method. The method was first tested by the author [Knight and Woon, 2003b] on the Iris data set [Fisher, 1936] and a simulated nominal value test problem. The simulated test problem consists of two simulated case bases. These case bases were generated with node sets regularly spaced and randomly spaced respectively using a function adapted from the one previously used by Ramos and Enright (2001) to test out the Shepard method. We [Knight and Woon, 2004b] then tested the method on a benchmark case base from the travel domain [Lenz *et al.*, 1996]. Results from both of the two tests have shown that the method out-performs conventional nearest neighbour methods on these data sets.

However, we suspected that GSNN's performance suffers from the "extrapolation trap" as all other interpolation methods normally do. Analysis was therefore carried out on the false predictions of GSNN in the travel problem. The analysis has shown that GSNN does indeed not perform well when the members of the retrieved set clump together. In an attempt to find out a way to mitigate this problem, more tests were then conducted to examine the GSNN performance on diverse retrieval sets. Diverse sets are normally considered less likely to suffer from the "extrapolation trap". Again the travel problem was selected to test the GSNN performance on a retrieval set generated by a diversity algorithm. Results [Knight and Woon, 2004b] have shown that GSNN's performance is in fact improved considerably. The improvement is nearly twice as good as conventional nearest neighbour methods.

The analysis along this line was further performed both on the random simulated case bases and the regular case bases. It has once again been shown that a diverse retrieval set can improve GSNN performance. These two tests have proven that the diversity algorithm improves results from a random simulated case bases more than from a regular case bases. This is because regular case bases give rise to retrieval sets that are already quite diverse.

One of the by-products of the work on nominal values is the third subsidiary question stated in Section 1.2.2.3. In the numerical domains, in order to cover the domain in reasonable detail, a large number of cases might need to reside in the case base. The bigger the case base, the more solutions it can contain. But, high dimensionality may cripple the CBR approach. It is often not advisable to generate a dense case base because it may lead to a substantial increase of the searching time. However it is not advisable either to keep too small a case base, which is incapable of representing the domain. The error function developed for the GSNN nominal value interpolation has several possible advantages for case selection: (i) it relies on distance metrics defined in the problem and solution space, (ii) it can predict solutions not yet present in the case base and (iii) it can predict solutions not in the retrieval set. It was furthermore envisioned that there might be another advantage. Because the error function gives an estimate of how well a given case is predicted by its neighbours, it could be useful in case selection. The question was consequently raised: *1.2.2.3 Can the error function developed in the nominal value interpolation be useful in case selection?* Other case selection methods also use an estimate, but it is often just a binary estimate "selected" or "not selected". The error function gives more information than a binary value does and might therefore be a better determiner in case selection. To examine this conjecture, GSNN was embedded as a case selection component in a decremental reduction algorithm called "Shrink" [Kibler and Aha, 1987]. Then the revised Shrink Algorithm [Woon *et al.*, 2003a] was tested on the Iris data set and an engineering application case - pneumatic conveying particles problem.

The Shrink algorithm reduces a case base with one less case in each iteration. The case to be removed is taken from a candidate set. The candidate set is usually taken to be the set of cases such that estimated solution ($\hat{y}$) = true solution (y_t). If the candidate set is empty, the algorithm will halt.

We first implemented a new removal strategy such that a case with minimum error ($Min_R \left| y_t - \hat{y} \right|$) is to be removed in each iteration and this would mean that the candidate set is never empty. In fact $Min_R \left| y_t - \hat{y} \right|$ is also a measure of how important y_t is to the case base's predictive power. In fact a better measure can be derived from the GSNN error function. We proposed $I(\hat{y}) = Min\,(I(y))$ to be the removal condition. It is a measure of the deviation between the estimated value and its *k*- neighbours prediction. $I(y_t)$ is a measure of the deviation between the true value and its *k*-neighbour prediction. $I(y_t) - I(\hat{y})$ is a measure of how near y_t is to be selected as the estimated value. If $I(y_t) - I(\hat{y}) = 0$, then y_t will definitely be selected. If $I(y_t) - I(\hat{y}) >> 0$, then y_t would be a long way from being selected.

Results [Woon *et al.*, 2003a] from the running of the three test cases have shown that the use of the error function of the GSNN nominal value interpolation as a measure of case selection can produce a sparse case base with good predictive power in the Iris data set and the benchmark case base from the travel domain.

1.4.3 Stage III

With each of the sub-problems solved individually, it came to examine if the approaches developed for these problems could be combined to answer the primary question stated in Section 1.2.1: Can CBR improve the usability of numerical models?

The CBR approach was tested for its capability in two specific engineering applications – the pneumatic conveyor and the projectile problem, and has been shown to produce answers immediately to several real design problems, which would otherwise have required considerable engineering expertise and time.

1.4.4 Stage IV

With the success of the CBR approach in the two specific engineering applications, it is worthwhile to explore the potential application scope of the CBR approach as a general tool for engineers. Interviews were carried out with individual experts to examine the feasibility of using CBR in other areas of mathematical model application. Such potential areas are discussed in Chapter 10.

The concluding chapter presents the work completed with respect to the project scope and the contribution of this study to knowledge. Although this study has answered all the questions raised in Section 1.2, there is still room for future work, which is discussed in depth in Chapter 10.

1.5 MAJOR ACHIEVEMENTS

In this thesis there are several original contributions to knowledge in the fields of CBR. These are:

1. A method for creating useful CBR models from numerical models to assist engineers in searching for solutions of engineering problems.
2. A CBR framework allowing constraints to be defined over unified problem:solution space.
3. A novel method for interpolation over nominal values termed as Generalized Shepard Nearest Neighbour (GSNN) method.
4. The use of GSNN in conjunction with the diversity algorithm – an approach, which is capable of improving predictive performance.
5. Investigations of the use of the GSNN error function as an error measure in case selection.

Some of these have already been published in the following publications: [Knight and Woon, 2003a, 2003b, 2004a, 2004b]; [Woon *et al.*, 2003a, 2003b, 2003c, 2004, 2005].

1.6 OVERVIEW OF THE THESIS

The thesis is divided into five parts – *Part I: The Research Problem, Part II: Core Idea of the Research, Part III: Case Studies, Part IV: Conclusion and Future Work,* and finally *Part V: References and Appendices.*

1.6.1 Part I: The Research Problem

The objective is to establish the fundamental question of this study and the motivation that drives this research.

1.6.1.1 Chapter 1: Introduction

Chapter 1 gives an overview of the background information of this study. It follows by the main research problem and subsidiary questions. The objectives of this study and the methodology used in this research are also presented together with the major contribution of this work to knowledge.

1.6.1.2 Chapter 2: Motivation

Chapter 2 explains the motivation of this research. The general architecture of a CBR - Numerical model system is defined and discussed. This collaborative combination has application in engineering domains where numerical models are used. The domain is termed as *"Case Based Engineering"*. To discuss the advantages and characteristics of this general architecture, two illustrative examples: (1) the projectile model and (2) the pneumatic conveyor problem are used.

1.6.1.3 Chapter 3: Literature Review

Chapter 3 gives a summary of what has been done and what has not, in the relevant fields with respect to this study.

1.6.2 Part II: Core Idea of the Research

In Part II, the core idea of the research is discussed. The four chapters describe in turn the methods proposed to answer the subsidiary questions in Section 1.2.2.

1.6.2.1 Chapter 4: Unified Problem:Solution Space in Case-Based Reasoning

In this chapter, a generalization of the standard CBR retrieval method, which integrates problem space and solution space into a single query space, is presented. The retrieval method is proposed by means of the concept of nearest neighbours to a constraint region. It is shown that the standard CBR retrieval is a special case of this general model. The advantages of the general model are explained in connection with its more general query modes by means of a variety of illustrative examples.

1.6.2.2 Chapter 5: Interpolation over Nominal Values in the Unified Space

In this chapter, a generalization of distance-weighted interpolation method to unified space, particularly with regard to nominal values, is presented. The method relies on the definition of an error function in terms of distance metrics in the unified spaces. The retrieved solution is taken to minimize this error function. The algorithm is shown to out-perform conventional nearest neighbour methods on sparse problems.

1.6.2.3 Chapter 6: Interpolation on a Diverse Retrieval Set

Chapter 6 shows that the Generalization Shepard Nearest Neighbour (GSNN) interpolation method gives good prediction with interpolation on a diverse retrieval set. This is connected to the general property that interpolation methods normally perform better at interpolation than extrapolation. Interpolating on a diverse retrieval set usually include distant neighbours making sure that a target case is within a set of interpolation points.

1.6.2.4 Chapter 7: Case Selection Techniques

In this chapter, a case base reduction method, which uses a metric defined on the solution space, is presented. The reduction method is given for case bases in terms of a measure of the importance of each case to the predictive power of the case base. It is shown that the method can produce sparse case base with good predictive power demonstrated in some example problems.

1.6.3 Part III: Case Studies

Part III shows the realisation of the methods proposed in Part II, in two studies: (1) The Projectile Model and (2) The Pneumatic Conveyor Problem. These studies show how CBR can be used to improve the usability of numerical models.

1.6.3.1 Chapter 8: Case Study I: CBE-Projectile

In this chapter, the same projectile model that has been discussed in Chapter 4 Section 4.4 and 4.5, is re-presented. However, this chapter discusses the implementation of a CBR model used for the Projectile model. The objective is to show that CBR can be used as an intelligent assistant to help engineers to search for a solution of an inverse problem of a physical system. Results obtained have shown that the CBR model is capable of producing answers immediately to the Projectile problems, which would otherwise have required considerable efforts and time.

1.6.3.2 Chapter 9: Case Study II: CBE-Conveyor

This chapter discusses the implementation of a CBR model used for an engineering practice problem: the Pneumatic Conveyor Problem, a simulation model for the degradation of particles, developed by Chapelle *at el.* (2003). The discussion presented in this chapter is a further demonstration of the use of the CBR approach to improve the usability of numerical models.

1.6.4 Part IV: Conclusion and Future Work

Part IV sums up the study described in this thesis and points to the possible directions of future work.

1.6.4.1 Chapter 10: Conclusion and Future Work

Chapter 10 gives a summary of what has been done in this study and the major contribution of this research to knowledge. It also discusses the limitations of this work and possible future enhancement. The chapter leads to the discussion of potential future research interests. This includes a list of numerical applications for testing.

1.6.5 Part V: References and Appendices

Part V is a collection of references and appendices. A list of references is presented. The appendices include a list of figures and tables, a list of refereed publications of this work and selected papers, and a selection of screenshots from the general Case-Based Engineering (CBE) model.

Chapter 2

Motivation: Usability Problems of Numerical Models

2.1 INTRODUCTION

Numerical models of physical processes can provide useful advice to engineers in many fields. They are often designed to simulate the evolution of systems over time, and operate in a forward time direction. Generally, engineers will specify inputs $\underline{I} = (I_1, I_2, \ldots, I_k)$, and the model will calculate output $\underline{O}=(O_1, O_2, \ldots, O_l)$, where $\underline{O}$ is a function of $\underline{I}$.

While a model of this type is of great help for engineers to gain insight into the real physical system represented by the model and both effectively and efficiently assist them in designing a better system, time and again engineers are caught up with an inverse problem of the original model. In fact, in many cases where the original model is derived to represent a system in the real world, the primary concern of the designer is to seek ideal inputs of the model that will produce given outputs. In addition, engineers often want to add constraints dynamically to outputs while searching for the right inputs. To solve these inverse and constraint problems directly will require a different computational model, often difficult or impossible to construct. For many practical systems where it is very logical and easy to develop the model to represent the real system of interest, it would be extremely difficult, if not impossible, to derive the corresponding inverse problem in an explicit or analytical way. Engineers find themselves with no option but to resort to an iterative search method: running the original model, looking at the results and changing the inputs accordingly for another run. In effect, the engineer is judiciously generating cases from the numerical model.

Experimental data collected by engineers are often used to assist in their design tasks. It is not rare that this data is more detailed and reliable than modeled data; covering all possible experiments scenario. On the other hand, in some circumstances [Kalman, 2000; *Jia et al.*, 2003; Simcox *et al.*, 1992] where experimental data are expensive to produce; it is desirable to utilize numerical models to generate a database, which may then be tested for accuracy of prediction. A database model of a process of this type may be represented by a set of stored predicates:

$$P\,(I_1,\ldots,I_k,O_1,\ldots,O_l)$$

Such a model can be queried quite flexibly using SQL by specifying either inputs or outputs, and constraints. However, such a model also suffers from some disadvantages:

- It can be a very large database, particularly if k and l are large, or if high accuracy is required.
- Queries using SQL can often give null results if the database is kept small.

The idea under examination in this research is to use a CBR system generated using a numerical model as a flexible query engine for engineers. The question we want to ask is this: "*Given an output, can a CBR system help the engineer to find the right input? Or, if an exact solution cannot be found, can a near solution to the target problem be of use?*". The answer to this question requires the use of CBR in collaboration with numerical models in problem solving. This collaborative combination has particular application in engineering domains where numerical models are used. We term this domain *"Case Based Engineering"* (CBE).

In this chapter, we study the illustrative examples of a projectile model and a pneumatic conveyor model in Section 2.2 and 2.3. The problems with the database approach are discussed in Section 2.4. In Section 2.5 we define and discuss the general characteristics of the CBE architecture. We look at the advantages of this architecture in the engineering domain and discuss the problems, which we encounter in attempting to set up a generally flexible query tool. We conclude in Section 2.6 with a summary of this chapter.

2.2 ILLUSTRATIVE EXAMPLE I: THE PROJECTILE MODEL

A simple projectile model is used to illustrate how a CBR tool can be used to help engineers in finding right input parameters for given outputs. The projectile model simulates the flight of a cannonball shot over flat ground in a given time interval, T. The model receives two inputs; v = velocity, θ = angle of gun. To plot the trajectory, each point is represented by (x,y) where $x = v\ t\ Cos\ \theta$ and $y = v\ t\ Sin\ \theta - 5t^2$ where $t \in [0,T]$.

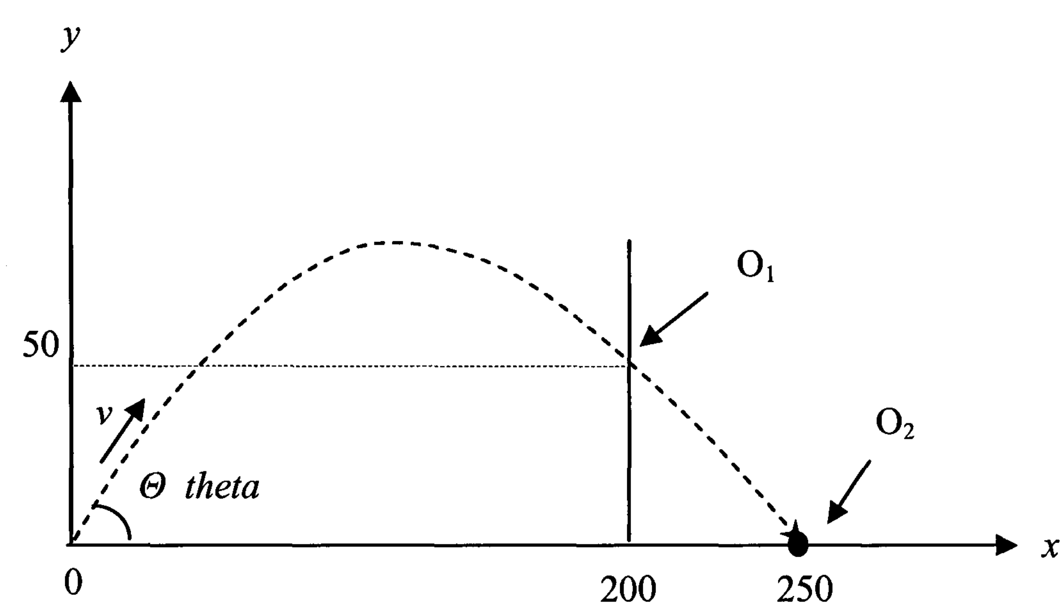


Figure 2-1. Visualization of the trajectory of a cannonball shot over flat ground aiming a target at 250 m

2.2.1 Example Problem

Consider the problem that a gunner tries to hit the target at x=250m on level ground, and also to clear a wall of height = 50m, at x =200m (see Fig. 2-1). From the gunner's point of view the problem is this: Can the right angle of gun and initial velocity be determined so that the cannonball will clear the wall at height = 50m? In this example, a range of possible velocity values and angle of gun are available, where $v \in [20,100]$ and $\theta \in [0.2,1.4]$. What is the best set-up to use?

The problem can be formulated as follows, representing the model in the form of a predicate, where θ is the angle of the gun, v the initial velocity. The inputs and outputs for the model can be represented as a set of predicates:

$P\ (I_1, I_2, O_1, O_2)$

Where $I_1 = theta,$

$I_2 = v,$

$O_1 =$ *the height of the cannon ball at the wall, when x=200m,*

$O_2 =$ *the projectile target.*

With this notation, the gunner's problem can be posed as a query:

$P(?I_1,\ ?\ I_2,\ O_1, O_2),$

$?\ I_1 \in [0.2, 1.4],\ ?\ I_2 \in [20, 100],$

$O_1 >= 50\ ,$

$O_2 = 250\ .$

As an SQL query this is (for example):

Select I_1, I_2, O_1, O_2 FROM P,

Where $O_1 >= 50$ AND $O_2 = 250$

AND (I_1 is BETWEEN *0.2* AND *1.4*)

AND *(* I_2 is BETWEEN *20* AND *100)*;

2.3 ILLUSTRATIVE EXAMPLE II: THE PNEUMATIC CONVEYOR PROBLEM

The second illustrative problem is derived from a practical engineering application: the pneumatic conveyor problem. Pneumatic conveying is an important transportation technology in conveying solid bulks in industry. Attrition of powders and granules during pneumatic conveying is a problem that has existed for a long time. One of the major industry concerns is to investigate how parameters such as air velocity, loading ratio, the angle of the bend, etc. affect degradation. For example, Hilbert (1984) studied different bend structures. Marcus *et al.* (1985) have investigated the pressure loss of different bends. Agarwal *et al.* (1985) considered acceleration length due to bends and the effects of phase density, etc. Weinberger and Shu (1986) examined the effects of the curvature radius of a bend on the transition velocity (the gas velocity at which minimum pressure drop occurs). Bell *et al.* in their studies (1996) discovered that air velocity has the prime effect on the attrition rate. Such knowledge is of great use in the design of conveyors. Often, engineers are more concerned with what input parameters will produce a desirable size distribution of particles. Fig. 2-2 shows the schematic diagram of a sample pneumatic conveyor. Particles are fed into a hopper and are transported to a receiver using a pneumatic conveyor.

This engineering application represents a typical scenario in which extreme difficulties arise to construct the inverse or constraint problem from the original model. It is almost impossible to find a mathematical expression for the inverse problem and hence we are not able to seek its solutions by solving a mathematical system either analytically or numerically. It is a typical scenario that engineers are left with few options but the inefficient trial and error approach. It is for a system of this type that the CBE approach is expected to make a significant difference in helping engineers to design the system of interest both efficiently and effectively.

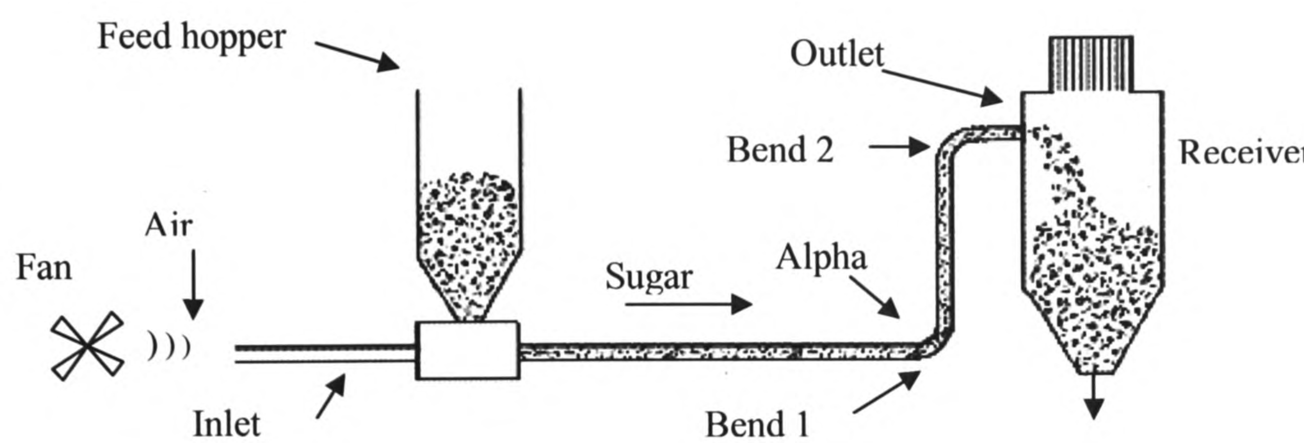
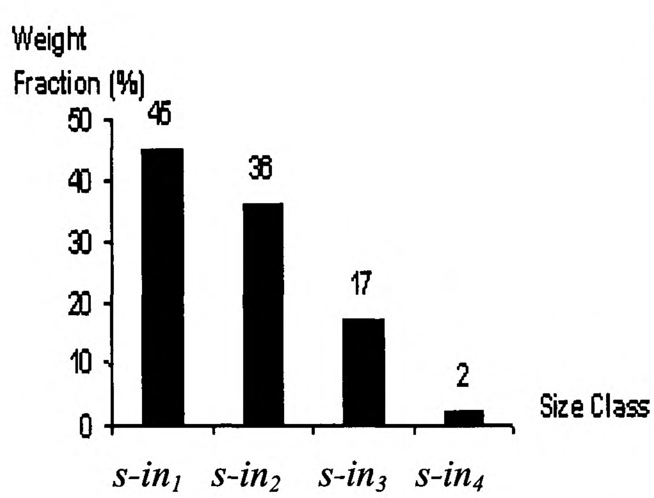


Figure 2-2. The schematic diagram of a sample pneumatic conveyor



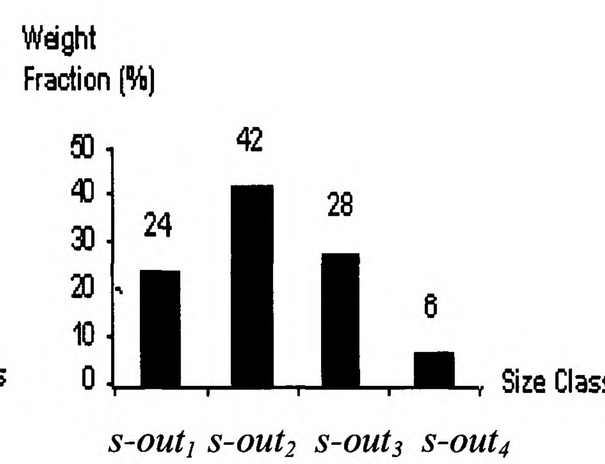


Fig. 2-3. Particle size distribution at the inlet **Fig. 2-4. Particle size distribution at the outlet**

2.3.1 Example Problem

From a designer's point of view the problem is this: Can the right bend type, angle of bend, and air velocity for a given particulate, e.g., sugar or tea., be determined so that there is not too much dust formed (i.e., very small particles) particles in the output receiver? Take for example: there are 4 types of bend, 3 angles of bend and three pipe diameters available. There may be only low power fans available sometimes, so the air velocity may be constrained. What is the best set-up to use?

Again, the engineering problem can be formulated as follows, representing the model in the form of a predicate, where *bend1 and bend2* are the types of bend, *bend angle, vair* is the air velocity, $s\text{-}in_1$, $s\text{-}in_2$, $s\text{-}in_3$, $s\text{-}in_4$ are size distributions of particles going in, and $s\text{-}out_1$, $s\text{-}out_2$, $s\text{-}out_3$, $s\text{-}out_4$ are size distributions of particles coming out of the conveyor (i.e., Fig.2-2, Fig.2-3, Fig.2-4).

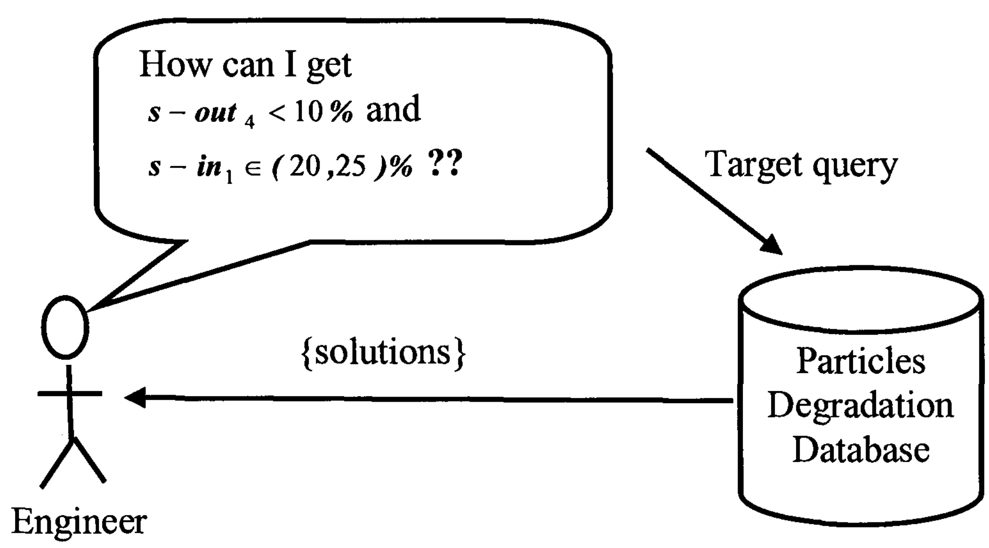


Figure 2-5. An engineer makes a query to the Particles Degradation Database

The inputs and outputs for the model can be represented as a set of predicates:

$P(I_1, I_2, ..., I_7, O_1, O_2, O_3, O_4)$

Where $I_1 = bend1$, $I_2 = alpha$, $I_3 = vair$,

$I_4 = s\text{-}in_1, ..., I_7 = s\text{-}in_4, O_1 = s\text{-}out_1, ..., O_4 = s\text{-}out_4$.

The designer's problem can be posed as a query (Fig. 2-5):

$P(?I_1, ?I_2, ?I_3, I_4, I_5, I_6, I_7, O_1, O_2, O_3, O_4)$,

$?I_1 \in \{$ *long radius, short radius, blinded tee, turbulence drum* $\}$,

$?I_2 \in \{30, 45, 90\}$, $I_5 < 25\%$, $I_5 > 20\%$, $O_4 < 10\%$.

As a SQL query this is (for example):

Select I_1, I_2, I_3 FROM P,

Where (I_1 = *'long radius'* OR I_1 = *'short radius'*

OR I_1 = *'blinded tee'* OR I_1 = *'turbulence drum'*)

AND ($I_2 = 30$ OR $I_2 = 45$ OR $I_2 = 90$)

AND $I_4 < 25$ AND $I_4 > 20$

AND $O_4 < 10$;

2.4 THE DATABASE APPROACH

Given a large enough database, the database approach should give a range of possible solutions. However, there are a number of problems associated with the database method. These include the following:

- The database may be expensive to produce.
- High dimensionality may make the database solution infeasible. In the conveyor problem, there are 7 degrees of freedom, and in order to cover the domain in reasonable detail, say, 10 points are needed in each dimension, it would give 10^7 records altogether.
- The solution set could be very large and hence unhelpful in decision making.
- Equality constraints might mean there are no solutions at all to the SQL query.

Some of these problems can be addressed by means of a CBR model. The problem of database size may be reduced somewhat by means of a sparse database of important cases. Although at first sight the dimensional catastrophe is still present, there is still the possibility to produce a relatively small efficient case base to replace a large database. One reason for suspecting this is that the focus in this study is the domain of numerical models. In this field, there is a great deal of regularity in the model, and one would expect fine detail to be well represented by some adaptive process such as interpolation. This should allow a great reduction in storage. Interpolative CBR systems have been studied by Chatterjee and Campbell (1993), and by Knight and Woon (2003b), who propose a generalization of Shepard's method [Shepard, 1968] known as GSNN (details can be found in Chapter 5).

Other problems associated with the simple database model such as the large solution set problem and the equality constraints problem may also be alleviated in the CBR approach. CBR retrieval is on the whole more amenable to usability questions than SQL does, giving cases ordered by closeness to input criteria. It will always give answers, and they might be ordered according to user needs.

2.5 THE CASE-BASED REASONING (CBR) APPROACH

In this section, we show how CBR can be used as a flexible query engine to assist engineers to solve inverse or constraint problems. Ideally engineers would like to be able to express their problem constraints without worrying whether the variables are inputs or outputs. Sometimes they need the right inputs for given outputs; sometimes they know some inputs and some outputs. For example, in the conveyor problem the engineer may only have certain bend types available for a design. This is expressed as a constraint on inputs. They may also need to be certain that not too much small particle dust appears in the output receiver: this is a constraint on outputs. The CBE architecture proposed here is designed to handle constraints of this type, allowing the engineer to define any constraints over the unified input and output space.

2.5.1 The General Case Based Engineering (CBE) Architecture

Before examining the special problems we encounter in CBE, we first describe the collaborative CBE architecture, between the engineer, CBR system and numerical model. We also include the case base maintainer as a separate agent in the overall architecture. These agents work together by sharing individual knowledge and expertise to solve inverse and constraint problems.

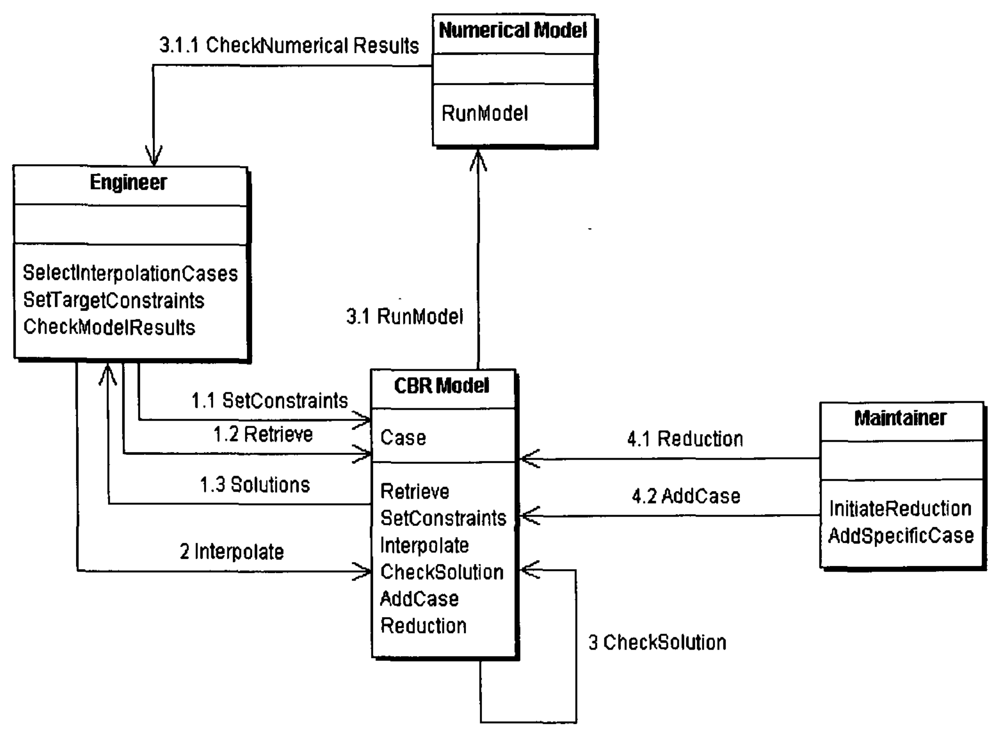


Figure 2-7. The UML collaboration diagram of the general CBE architecture

Fig. 2-7 is a UML collaboration diagram showing the interactions between these agents. The sequence of steps in a typical query session is as follows:

1.1 The engineer defines a set of constraints over input-output space defining the problem.

1.2 The CBR model retrieves cases near to the problem definition.

1.3 The CBR model presents a list of useful cases to the engineer. The engineer can examine these cases, and possibly redefine the problem if the initial definition was not complete, or was incorrect in some way. There is also an opportunity for the engineer to select some of the retrieved cases manually for the next phase (adaptation). This would be useful in situations where the engineer needs to have more 'hands–on' control of the whole retrieval process

2. The engineer requests the CBR model to perform interpolation on a retrieved set of cases. The retrieved set may be that selected by the engineer, or simply the *k* nearest neighbours. It has also been shown in [Knight & Woon, 2004b] that interpolation can work better on diverse sets. The interpolation phase needs often to be able to deal with nominal values, and to handle a variety of constraints. It also has to make sure that the interpolation set is conformable for interpolation; sometimes two solutions,

though close in the problem space, are not at all close in the solution space and should not be used for interpolation. We examine these problems of interpolation later in Section 2.5.2.2

3.1 The adapted solution produced by the CBR system has values for all inputs and outputs. It is now possible to run the model against the inputs, and verify the outputs.

3.1.1 The simulation results are then presented to the engineer who can decide whether the solution is acceptable. It may well be that they may need to return to Step 1.3 and select a different set for adaptation. In situations where there is a large difference between the modelled and adapted solution, we have the possibility to add the new modelled case to the case base. The addition of a new case will give reason to return to Step 1, and the session can continue with the new case base.

There are also two interactions shown in the collaboration diagram separate from those described above, which involve the case base maintainer. These are:

4.1 Generation of the initial case base. This must depend upon the dimensionality of the problem space, and the cost of model generation. For fast models and low dimensionality, we can simply produce a regular dense database. Regular here means that we simply divide each dimension into equal steps and impose a Cartesian grid of points on the domain. However, for high cost, long run time models of high dimensionality (e.g., computational fluid dynamics models), the case base would of necessity be sparse, and we would have to rely on the effectiveness of the interpolation scheme. It is often not simple to cover the case base uniformly. Some regions of the case dimensions are not practical, i.e., not naturally occurring.

4.2 Subsequent maintenance of the case base such as adding and removing cases is managed by the Case Base Maintainer, which may utilize case reduction schemes co-operating with the numerical model.

In fact, we can gain some insight of the collaborative architecture in the scenario of how fire investigators try to solve inverse problems (i.e., determine the fire origin and the exact fire development process). Often fire investigators are acting on previous cases from experience in order to find the location of a fire source and interpret how the fire has developed. They either carried out real fire tests or run numerical model to confirm their findings. However, such experimental data can be expensive to produce. In this case, a case base can be generated using a numerical model in addition to real fire data. By acting inversely, the engineer can input a list of expected outputs obtained from the post-fire evidence and query the case base to find the possible locations of a fire source. When necessary the CBR model performs additional simulations to confirm the fire investigator findings.

For the numerical model to be queried in a flexible manner (i.e., with no distinction between inputs and outputs), it would be desirable to preserve the convenient property of databases in the CBE architecture so that the inverse problem becomes just another query. However, there are a number of problems need to be dealt with. These problems are discussed in Section 2.5.2.

2.5.2 Elements of the CBE Architecture

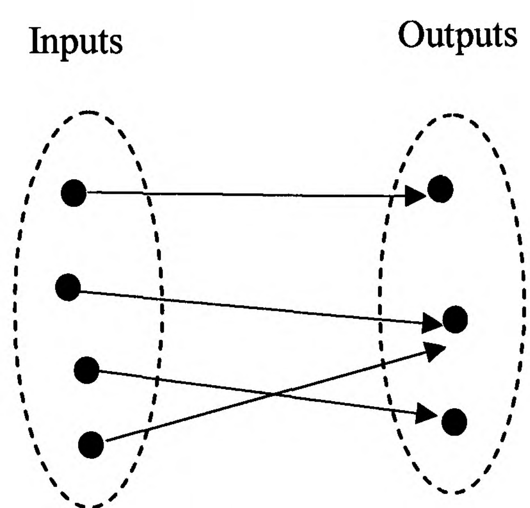
In this section we examine in detail some of the special issues that arise in the design of a working CBE system. These are mainly due to the need of the engineer to search and interpolate over the whole input – output space. This entails four main problems, which we discuss here. The first problem is the handling of constraints. The second problem is that when querying numerical models in an inverse manner, one cannot assume single-valued solutions. The third problem is to do with interpolation over nominal values. The fourth problem is the definition of metrics on the query space.

2.5.2.1 Constraints

Both the projectile model and the conveyor problem are constraint problems. For instance, the SQL example given in Section 2.3.1 shows two types of constraints: continuous constraints of the type "Where $s\text{-}out_1$<10" and nominal constraints like "(*bend1 = long radius OR bend1 = short radius OR bend1 = blinded tee OR bend1 = turbulence drum*)". Both constraints will involve inputs and outputs. One approach to dealing with constraints is to run the SQL query given above on the case base itself, beforehand, thus retrieving the nearest neighbour solution that satisfies the constraint. However, for sparse case bases this may not be desirable; it may give poor solutions when adaptation is required.

2.5.2.2 Multi-valued Case Mappings

Numerical models are generally deterministic in nature, so that $\underline{\mathbf{O}}$ is given as a single valued function of $\underline{\mathbf{I}}$. The inverse problem cannot be assumed as single valued. As in Fig. 2-8, there may be several solutions to a given query where outputs are specified. For 1-NN retrieval (i.e., *k*-Nearest Neighbour (*k*-NN) where k=1, [Cover and Hart, 1967]), this gives little problem, since the multiple nearest cases may be ordered as equal for the user to select. For *k*-NN, it is not desirable to interpolate between cases which are not close in the input domain because outputs, which are close in the output domain may have diverse inputs. Bergmann *et al.* (2001) address a problem that the similarity of cases in the problem space does not always correspond to the usefulness of the cases in solving the problem. Fig. 2-9 shows an inappropriate interpolation (adaptation using C_1 and C_2) and that there is a much more relevant case (i.e., C_3 - in the input space) to a given target constraint, which may be a better candidate case than C_2 in adaptation. This problem is related to the constraint problem in Section 2.5.2.1. Although C_3 may be a better candidate case than C_2 in adaptation, C_3 does not satisfy the target constraint. As a result, C_3 will not be retrieved. One approach to this problem could be to perform adaptation on the cases that are close in the unified (problem:solution) space. This would require standard CBR models to be extended to a unified space so that engineers can define their problem constraints without worrying whether the variables are inputs or outputs.



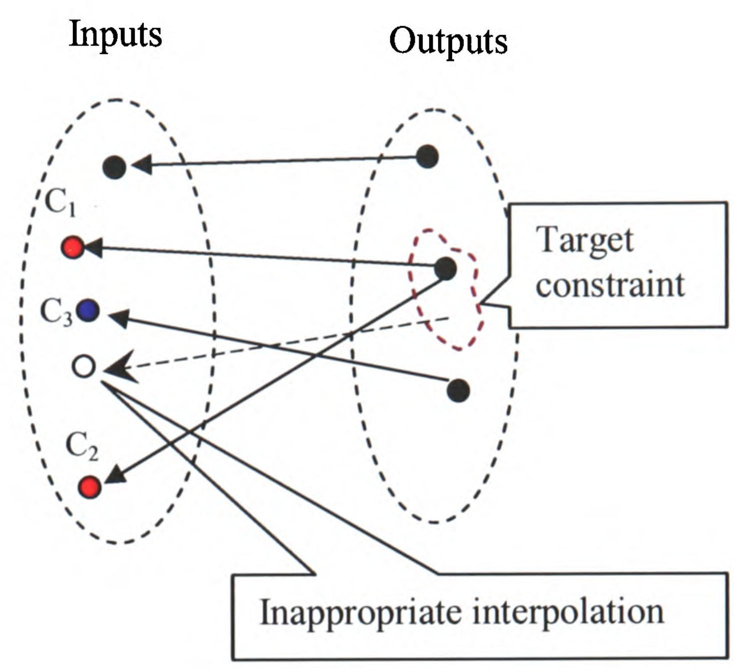


Fig. 2-8. Direct problem (single-valued solutions) **Fig. 2-9. Inverse problem (multiple solutions)**

2.5.2.3 Nominal Values

The third problem is the handling of nominal values in constraints and interpolation. The inputs of numerical models are often nominal values, and not ordered linearly. In the conveyor example discussed in Section 2.3.1, there are two attributes that are nominal in nature: the bend type (e.g., long radius, short radius elbow, turbulence drum, circular, box bend, blinded tee, etc.) and the angle of the bend (e.g., 30 degree, 45 degree, 90 degree). These also happen to be values the engineer is searching for in a typical query. For example, they may want to know what bend type is best for a given output size distribution of particles. According to Step 2 of the scenario in Section 2.5.1, this means we should interpolate to find the bend type from a set of cases. This requires interpolation over nominal values. In addition, the engineer will often want to express the fact that she/he only has certain bend types available. Such a constraint should be taken into account when it comes to interpolation. Chatterjee and Campbell (1993) propose a method for interpolation, which assumes natural ordering of nominal values. The ranking of nominal values is based on the linear distance. Wilson and Martinez (1997) extend value difference metric [Stanfill and Waltz, 1986] for defining metrics over nominal values. Their method however, is not capable of coping with solutions not in the case base. *k*-NN and Distance-Weighted Nearest Neighbour (DWNN) [Mitchell, 1997] uses a voting function for classification in nominal domains. However, these methods do not produce intermediate values (i.e., values not in the case base or not in the retrieval set) [Knight and Woon, 2004b].

2.5.2.4 Definition of Metrics on the Query Space

Standard CBR models assume that the problem space and the solution space of a CBR model are separate. In the direct problem (e.g., Fig. 2-8) these will correspond to the inputs and outputs of a numerical model. However, in the inverse problem (e.g., Fig. 2-9), the inputs of the numerical model are the problem space, and the outputs are the solution space.

Ideally, the database model should allow engineers to query with SQL by selecting any set of variables (i.e., inputs/outputs of the numerical models or both) to be the problem space. The solution (query) space can be any subset of the entire space of inputs and outputs of the numerical models. To apply this equally to the case-based model, a similarity metric between a target query and the cases in the case base must be defined over the entire space of inputs and outputs. One question, which arose in this study, is: *"How do we define the similarity metric in the solution space?"*

2.6 CONCLUDING REMARKS

The chapter describes typical engineering problems with reference to two examples: (1) the projectile model, and (2) the pneumatic conveying problem. The approach to this kind of engineering problems makes use of a general architecture presented in this chapter. The architecture has applications in engineering domains where numerical models are used. We term this domain as "Case Based Engineering" (CBE). The CBE architecture comprises two main agents: a CBR system and a numerical model, to collaborate in searching for a solution. The key elements of the architecture have been discussed. These include the constraint problem and interpolation, nominal values, and metrics on the query space. These problems will be explored in detail in the rest of this thesis.

Chapter 3

Literature Review

3.1 INTRODUCTION

This chapter presents an overview of literature relevant to this research. The objective is to ensure that this research is not repeating the work of others. It is intended that this chapter will provide readers with a broad view of the research problem and with useful background information.

First, a summary of existing research with respect to the primary question: "Can Case-Based Reasoning methods be used to improve the usability of numerical models?" (stated in Section 1.2.1), is reported in Section 3.2 and 3.3. Section 3.2 presents a number of inverse problems and describes a selection of approaches used to solve different types of inverse problems. Section 3.3 presents a brief overview of Case-Based Reasoning (CBR), and existing work of CBR applications for numerical models.

In what follows, this chapter reports the literature relating to the sub-problems defined in Section 1.2.2. Section 3.4 describes various case base retrieval methods for filtering database records, which allow flexible query modes in relational databases (see Section 1.2.2.1). Section 3.5 presents an overview of existing interpolation methods for nominal values (see Section 1.2.2.2) and in Section 3.6 definitions of metrics defined on the query space are investigated. In Section 3.7 we discuss a selection of existing case selection techniques (see Section 1.2.2.3).

However, the topics that are related to each particular technical aspect of this research will be discussed in more detail in later relevant chapters.

3.2 THE INVERSE PROBLEMS (IP)

Prof. Oleg Mikailivitch Alifanov, the great Russian proponent of Inverse Methods, states: "Solution of an inverse problem entails determining unknown *causes* based on observation of their *effects*." In contrast, a direct problem involves finding effects based on a complete description of their causes.

In most industries, scientists often attempt to solve problems (i.e., finding unknown *causes*), based on observation of its *effects*. A review of IP applications shows that IP methods can be applied to a wide range of scientific and industrial fields.. This includes biological tissues [Ji and McLaughlin, 2003], signal processing and image recognition [Byrne, 2003; *West et al.*, 2004], statistical estimation [Evans and Stark, 2002], damage detection [Banks *et al.*, 2002], estuarine system [Bertino et al., 2002], etc.

In engineering fields where numerical models are used, engineers often are caught up with inverse problems with respect to the model of interest. Although the numerical model was designed to represent a system in the real world, the primary concern of the engineer is in certain circumstances to search for the right inputs of the model for given outputs. The following presents some examples of inverse problems faced by engineers:

A scenario of this kind might most be illustrated in the simple projectile problem. In the projectile problem, the trajectory of a cannon ball shot over flat ground depends on the initial angle and velocity of the gun. The physical description of the process, f, and the corresponding initial conditions, I= (angle θ and velocity V) constitute the causes of the problem, where $f: I \rightarrow O$. If these causes are known, the trajectory of a cannon ball (i.e., O) can be found. Hence, O is the *effect* of I. Suppose we know the range of the projectile target and f and there is a constraint defined such that the trajectory must clear a wall at certain height. The searching for the initial conditions that satisfy the suppositions constitutes an IP problem.

Another inverse problem taken from the example provided by Palansuriya (2000) is the metal cutting problem. In the metal cutting problem, the temperature of the cutting blade has a great effect on its life and on the cutting quality. A model with the speed of cutting as an input can generate the temperature distribution within the plate being cut. We can measure these temperatures at various points, but not the temperature of the blade directly. Hence the problem is: given the temperatures at these points, can we find the temperature distribution within the blade? This is an IP with respect to the model, which works the other way: given the temperature of the blade, find the distribution in the metal sheet. If we can solve the IP here, we can work out a control scheme for the cutter speed.

In fire investigation, one of the key objectives is to establish the cause of a fire accident and the process of the fire development. Based on the post-fire evidence collected at the fire scene, fire investigators will try to find out where the fire started and how the fire spread during the course of the fire development. It is not uncommon in fire investigation that investigators rely on their experience of previous work and intuitively follow basic laws of fire dynamics. However, fire is complicated and complex phenomena that a wide range of factors influence its process. This working practice may lead to misinterpretation of the post-fire evidence and the drawing of wrong conclusions from them. With the advance of computer technology and fire science, fire modelling, particularly CFD (computational fluid dynamic) fire modelling, has increasingly been used as a useful tool to aid fire investigation [Simcox *et al.*, 1992; Jia *et al.*, 2003]. While the searching of the fire origin and its progress course of a fire based on the observations from the fire scene is a typical problem of determining unknown causes based on observation of their effects, it is unlikely that we can construct an inverse problem mathematically for a particular fire case due to the complexity of fire phenomena and CFD fire modelling in its own nature.

3.2.1 Methods for Solving the Inverse Problems

There are a multitude of inverse problems. Palansuriya (2000) cites several approaches to solve inverse problems such as domain decompositions [Kunisch and Tai, 1996], graphical [Stolz, 1960], polynomial [Frank, 1963], Laplace transform [Krzysztof *et al.*, 1981], dynamic programming [Trujillo, 1978], finite difference [D'Souza, 1975], finite elements [Krutz *et al.*, 1978], least square method [Chow *et al.*, 1999] and regularization methods [Beck and Murio, 1986; McMasters and Beck, 2000]. However, there is no one best inverse method that can solve all IPs. We present here some to serve only as a representative selection of some general methods used for solving the inverse problems.

3.2.1.1 The 'Trial and Error' method and the Heuristic Approach

According to Pearl (1984), "Heuristics are criteria, methods, or principles for deciding which among several alternative courses of action promises to be the most effective in order to achieve some goal. They represent compromises between two requirements: the need to make such criteria simple and, at the same time, the desire to see them discriminate correctly between good and bad choices." . A heuristic may be a rule of thumb, which can be used to guide one's decision.

The *'Trial and Error'* method requires running a numerical model in a repetitive manner: looking at the output, adjusting the input parameters accordingly until the expected output is attained. The degree of adjustment on the parameters may depend on the intuitive judgement of engineers. Sometimes, this can be achieved by using an evaluation function to estimate the prediction accuracy. The 'Trial and Error' method [Barrett *et al.*, 1994] is a form of heuristic approach.

These methods are usually effective however not guaranteeing a solution or correct judgement. The following shows a possible approach to search for the solution of an inverse problem:

1. Set one of the setup parameters to maximum value and the others to minimum value
2. Gradually increase or decrease either one of the parameters
3. The judgment of the parameter variation should be based on the known effect of the parameter.

4. The judgment of the parameter variation should be based on the relative importance of the parameter.

The 'Trial and Error' method [Mahnken, 2004] has the following advantages: (i) it does not require additional programming for solving inverse problems; (ii) it can be used for all types of inverse problems; (iii) it can effectively exploit the knowledge of experts. The disadvantages are: (i) it is time-consuming; (ii) it does not guarantee accurate solutions; (iii) it does not have a specific objective criterion.

3.2.1.2 Neural Networks

A neural network is built out of a densely interconnected set of biologic nerves, the neurons. It simulates the information flow between neurons. Neurons are arranged in layers. A neural network may comprise an input layer, an output layer and others such as intermediate layer for hidden units (outputs of hidden units only available within the network). Each node (i.e., neuron) in the network has weights, which are determined by the training process. It uses the back propagation algorithm that relies on an error function to optimize the prediction of material parameters. The Neural Networks method [Mahnken, 2004] has the following advantages: (i) it can predict the material parameters simultaneously; (ii) it can be used for direct and inverse problems.

3.2.1.3 The Least-squares method

According to the least squares method, the best-fit curve has the property such that it has the minimal sum of the deviations from a given set of data. By minimizing the discrepancies between the simulated data and the experimental data, the best-fit curve can be determined and subsequently an estimate of parameters for a given problem. If the objective function involves dimensions that are less reliable or less relevant, a weighted least-squares estimate [Norton, 1986] can be used. The least-squares method has the advantage [Mahnken, 2004] that it can be used to determine input parameters of a given inverse problem simultaneously.

3.3 CASE-BASED REASONING (CBR) AND NUMERICAL MODELS

3.3.1 What is CBR and how does CBR Work?

"CBR is the essence of how human reasoning works. People reason from experience. They use their own experiences if they have a relevant one, or they make use of the experience of others to the extent that they can obtain information about such experiences. An individual's knowledge is the collection of experiences that he had has or that he has heard about..."
"The basic idea in case-based reasoning is simple: A case-based reasoner solves new problems by adapting solutions that were used to solve old problems. ...case-based reasoning means reasoning from prior examples"

Riesbeck and Schank (1989)

CBR assumes the world remains regular, consistent and predictable so that we can apply what we have learned in the past in similar circumstances in the future [Kolodner, 1996]:

- **Regularity.** The world is essentially a regular and predictable place. The same actions performed under the same conditions will normally have the same (or very similar) outcomes.
- **Typicality.** Events tend to repeat. Thus, a CBR system's experiences are likely to be useful in the future
- **Consistency.** Small changes in the world only require small changes to our reasoning and need correspondingly small changes to our solutions.

A typical CBR system consists of four phases: (1) Retrieve, (2) Reuse, (3) Adapt, and (4) Retain. When a new problem (Target case) is presented to a CBR system, the target case is then matched against source cases in the case base to retrieve the most similar case(s). Once the best match (i.e., the most similar case(s)) has been identified, it would be used/adapted (if necessary) to solve the new problem. If adaptation is required, the new case will be stored in the case base for future use. How much adaptation needs to be done varies depending on the differences between the best match and target case. The CBR cycle is illustrated in Fig. 3-1.

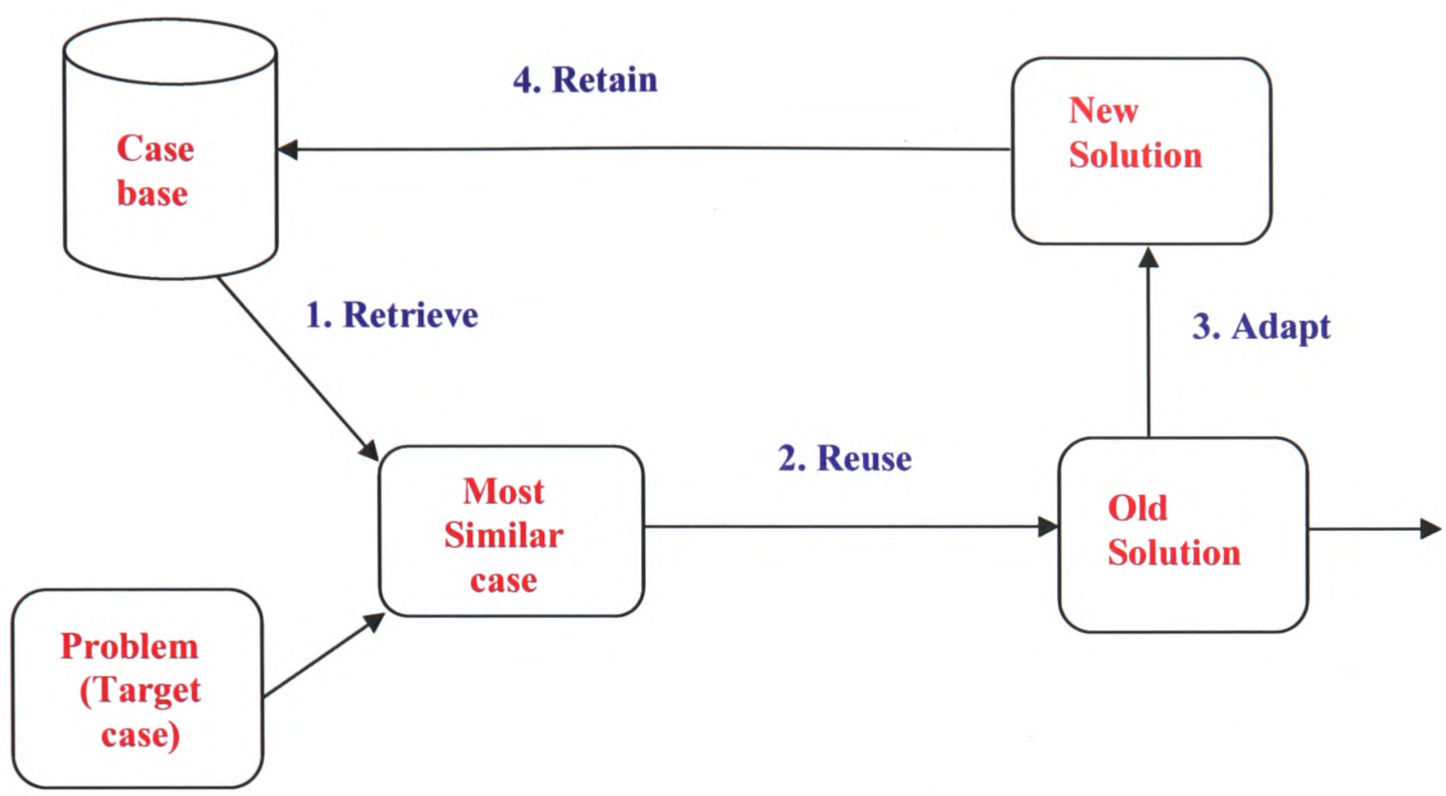


Figure 3-1. The Case-Based Reasoning cycle

3.3.2 Using CBR for Numerical Models

In this section, a selection of examples where CBR is used in specific numerical domain is presented. This includes the prediction of air quality problem, the colour matching problem, optimization setup of engineering designs and model generation in numerical solution. These examples show various approaches of using CBR for numerical models. They are discussed as follows:

3.3.2.1 The Air Quality Prediction Problem

Kalapanidas and Avouris (2001) give an account of a prototype, NEMO, built using a CBR approach combining heuristic and statistical techniques to support short-term prediction of NO_2 maximum concentration levels in Athens, Greece. The prototype consists of three main modules: (i) Retrieving cases, (ii) Filtering cases and (iii) Adaptation. In their model all the attributes involved are continuous and they have used the standard weighted sum method [Kolodner, 1993] for retrieval. Besides, they use heuristic knowledge to reduce the search space. Adaptation is achieved by parameter adjustment (similar to the one used in PERSUADER [Sycara, 1988]) corresponding to differences between a target problem and a retrieved case. In their approach, they use a statistical formula. They claim that the NEMO classifier can give fast prediction for the likelihood of an occurrence. It is robust to noisy data. The prototype is also shown to have an advantage over Artificial Neural Network (ANN) and Decision Tree (DT) in that it can be adapted to time-evolving problems without the need of frequent changes in the prototype itself.

3.3.2.2 The Colour Matching Problem

Cheetham and Graf (1997) describe a CBR tool called FormTool to help users to select a subset of the allowable colourants for colour matching in the General Electric(GE) Plastics. In the case base, each case is represented by a reflectance curve and a list of pigments and loadings used to create that color. During the colour matching process, FormTool searches for the best match and adapts it to get a closer match to the customer's colour standard when necessary. In the case selection process, a fuzzy preference function is used to calculate the similarity for each attribute of the case. Adaptation is done by iterative change of loading of the colourants and the evaluation of the new similarity. In this approach, Kubelka-Munk theory is used. The Kubelka-Munk theory provides a formula for predicting the colour change from modifying the loadings of the colourants. FormTool [Cheetham, 2001] has proved to be cost savings and increase of colour matcher productivity. The technology of FormTool is later used to create a web-based colour selection tool, ColorXpress Select. ColorXpress Select is one of the first web-based customer service tools implemented by GE Plastics. It has been around since 1999.

3.3.2.3 Numerical Optimization Setup of Engineering Designs

Schwabacher *et al.* (1998) have constructed a case-based system based on induction learning for the numerical optimization setup of engineering designs. They state that when setting up a numerical optimization, there are several factors that may affect the reliability of numerical optimization of complex engineering designs. These factors include the selection of a starting prototype, the selection of a formulation search space, etc. They argue that machine learning techniques can help in setting up an optimization based on the results of previous optimizations. It has been shown in their study that inductive learning such as C4.5 [Quinlan, 1993] and CART [Breiman, 1984] improves the speed and the reliability of design optimization. When designing a new artifact, the inductive learner generates or learns a decision tree from the design library. The decision tree is then used to map the new design goal into a starting prototype selected from the design library or selected formulation of search space. They have evaluated the various prototype-selection methods efficiency based on the capability of using the starting prototype to produce optimal design. The focus in their study is how inductive learning can be used in the retrieval phase (by comparing various prototype-selection methods with inductive learning) and that the Rutgers Hill-climber [Schwabacher *et al.*, 1998] is used as optimizer.

3.3.2.4 Model Generation in Numerical Simulation

Solution reuse for model generation in numerical simulation via the CBR approach has been studied extensively by Finn and Cunningham (1998). In their study, they propose an interactive knowledge-based system called CoBRA (CBR Assistant) that assists engineers in the task of mathematical modelling for heat transfer simulation problems. The CoBRA system is designed to assist engineers in formulating and evaluating spatial modelling decisions. In CoBRA, there are two key CBR processes: (1) case retrieval, and (2) case adaptation. Prior to the retrieval phase, the target case is decomposed into a number of smaller target cases, where each case is based on a single feature of the physical system. Retrieval phase involve two stages: (1) Matching of cases to the target case and (2) One-to-one mapping between the features of the target case and source cases in the case base. The adaptation phase is carried out using the derivational analogy method.

3.4 CASE RETRIEVAL TECHNIQUES IN RELATIONAL DATABASES

One of the objectives in this research is to build a CBR model that allows engineers to query the case base in a flexible manner. This section presents a study of a selection of case retrieval techniques used by researchers in filtering database records.

3.4.1 Fuzzy-based Retrieval with SQL

Portinale and Montani (2002) propose a fuzzy case retrieval approach based on SQL as a flexible product search engine for implementing an electronic catalogue. They focus on classification and prediction via fuzzy weighting. In their study, they address the problem of equality constraints in querying database records and argue that imprecise information can be more adequately modeled via a standard fuzzy-based approach. In their approach, the usual SQL query is extended to a much wider use by setting a similarity threshold so that those cases having similarity degree greater than the specified threshold will be retrieved. Each case is defined as a collection of <feature, value> pairs where each feature can only take values defined for that attribute domain. However, a target query can be defined by using linguistic (fuzzy) abstractions on such attribute values. This means each feature defined in the query can be described in one of these expressions: (1) linguistic value, (2) crisp value (i.e., is) and (3) a fuzzified crisp value.

3.4.2 Similarity-based Retrieval with SQL

A case retrieval engine built on top of a relational database in the context of intelligent product recommendation agents is presented by Schumacher and Bergmann (2000). First, by "retrieval on top of the database" they mean that the retrieval engine is a separate module that interfaces with the database. This approach differs from the "retrieval inside the database" in that the latter refers to integrating the retrieval function into the database itself. They argue that the need for building a retrieval engine as a separate module is that there is no standardized language for a similarity-base query formulation that is widely accepted by the database community. In general their idea is to approximate a similarity-based retrieval with SQL-queries. A user can assign weights (i.e., importance of an attribute), filters (i.e., hard constraints for the attribute values of the case) and *k*, the number of cases to retrieve. In their approach a query relaxation technique is used for retrieving cases when more cases are needed to determine the retrieval result. The method is implemented in a commercial CBR toolbox, ORENGE. It is shown in their study that speed of the relaxation technique is crucial in determining the number of cases that will be retrieved in each cycle. However, the retrieval efficiency suffers from the distribution of the cases on the representation space. If distribution of the cases is not uniform the retrieval efficiency will decrease resulting too many cases will be retrieved and too many SQL queries are generated.

3.4.3 Similarity between SQL Specifications

Shimazu *et al.* (1993) state that a relational database was essential for CBR systems to be used as part of a corporate-wide information system. They argue that computing similarity for individual cases is time-consuming, hence, in their approach, database query form similarities are computed instead. They use Neighbour Value Set (NVS) combinations to generate SQL specifications. NVS is a set of values neighbouring the target value specified by the user. Each NVS combination is translated into a SQL specification to calculate the similarity between a target specification and SQL specifications. The method has been implemented in the Case Retrieval Tool (CARET) and evaluated by a corporate-wide case-based system for a software quality control domain. It was shown in their study that the method provides retrieval results equivalent to those of non-relational database implementation at fast response time.

3.4.4 Predictor Restrictions and Goal Restrictions

Stanfill and Waltz (1986) state that the aim of memory-based reasoning is to predict the goal fields of a target record by retrieving records from a database. They present two restriction techniques called predictor restriction and goal restriction for restricting database records. Predictor restriction operates by finding the most important predictor and restricting that same predictor field of other database records containing the same value as the target record. Goal restriction is accomplished by finding plausible values for the goal field, and then filtering the database record so that their goal fields contain one of those values. Once the reduced set of records is determined, one can then apply dissimilarity measure to this subset. When retrieving the database records that most closely match the target, one can either set a dissimilarity threshold to retrieve records with smaller dissimilarity ratings or retrieve *n* closest matches where *n* is an integer. The prediction of a target's goal fields is given by the goal fields' values of retrieved database records, weighted according to their dissimilarity.

3.5 INTERPOLATION METHODS

Interpolative methods are well studied in the real domain, and can give good results from relatively sparse datasets. For many CBR systems, case bases are often sparse, being hard or expensive to obtain. When the solution space for such systems is real, interpolative methods can be used to increase the accuracy of solution. This section discusses several approaches to interpolation.

3.5.1 Chatterjee and Campbell

Chatterjee and Campbell (1993) advocate an interpolative approach that can serve as a domain-independent basis for quick and efficient time-critical CBR. They treat nominal values as linearly ordered (i.e., with a restricted class of similarity metrics). Their approach assumes a prior ordering of any nominal domains being present in the application so that there exists a mapping from solution points onto the real line. They cite several existing systems that have come close to the idea of interpolation, though without mentioning it explicitly. These include the model based meal planner CHEF [Hammond, 1987; 1990], the route planning systems TRUCKER and RUNNER [Converse *et al.*, 1989; Hammond *et al.*, 1993; 1988], the problem mediator system PERSUADER [Sycara, 1988], and the menu designer JULIA [Hinrichs and Kolodner, 1991]. They give the example of CHEF, which *"uses artificial numeric values for ordering objects according to their properties: 'taste of broccoli is savoury with intensity 5' and 'taste of beef is savoury with intensity 9'... Any* *adaptive steps taken in these systems, on the basis of these ordered features, fall within the purview of interpolation.* ". However, their methods cannot cope with nominal values where there is no natural ordering.

3.5.2 Instance-Based Learning Retrieval Algorithms

3.5.2.1 k-Nearest Neighbours (k-NN)

The *k*-Nearest Neighbours (*k*-NN) algorithm [Cover and Hart, 1967; Mitchell, 1997] has been a primary method used in classification. It predicts the solution for a given target query based on the output classification of its *k*-nearest neighbours assuming that the output class of the target query be most similar to the output class of its nearby instances in (possibly weighted) Euclidean distance. This algorithm can be used for approximating both discrete-valued target functions and continuous-valued target functions. In discrete-valued domain, the output class that has the most common vote among its *k*-nearest neighbours determines the output classification of the target query (see E(3-1).). For continuous-valued target functions, the mean value of its *k* nearest neighbours is taken as the solution rather than their most common class. The discrete-valued target functions and continuous-valued target functions are given as follows:

***k* - Nearest Neighbour Algorithm (In discrete-valued domain)**
$\hat{f}(x_q) \leftarrow \arg\max_{y \in Y} \sum_{i=1}^{k} \delta(y, f(x_i))$. E(3-1) where $\delta(y, y') = 1$ if $y = y'$ and where $\delta(y, y') = 0$ otherwise.

***k* - Nearest Neighbour Algorithm (In real-valued domain)**
$\hat{f}(x_q) \leftarrow \sum_{i=1}^{k} \frac{f(x_i)}{k}$. E(3-2)

3.5.2.2 *Distance Weighted Nearest Neighbours (DWNN)*

An alternative to *k*-NN is the Distance-Weighted Nearest Neighbour method (DWNN) [Mitchell, 1997]. In this algorithm, the contribution of each of the *k*-nearest neighbours is weighted according to its distance to the target query. When this algorithm is used in real-valued target functions the weighted mean value of the *k* nearest neighbours is taken (see E(3-4).), and it is thus a localized version of Shepard's method [Shepard, 1968]. However, for discrete solution spaces, a voting mechanism replaces the weighted mean. The Distance-Weighted Nearest Neighbour Algorithm for discrete solution spaces takes the solution value with the highest vote, where the vote for a particular solution is weighted by the inverse square of its distance from the target (see E(3-3).). The advantage of using this method is that it is robust to noisy training data. By weighting the contribution of each neighbour to the prediction of a target, it can smooth out the impact of distant noisy case. However for discrete-valued target functions, both the *k*-NN and DWNN methods assume that the predicted solution must exist in one of the nearest neighbours. It is not possible to produce intermediate values not present in the case base. The discrete-valued target functions and continuous-valued target functions are given as follows:

Distance Weighted Nearest Neighbour Algorithm (In discrete-valued domain)
$\hat{f}(x_q) \leftarrow \underset{y \in Y}{\arg\max} \sum_{i=1}^{k} w_i \delta(y, f(x_i))$. E(3-3)
where $w_i = \dfrac{1}{d(x_q, x_i)^2}$; $\delta(y, y') = 1$ if $y = y'$ and where $\delta(y, y') = 0$ otherwise.

Distance Weighted Nearest Neighbour Algorithm (In real-valued domain)	
$\hat{f}(x_q) \leftarrow \sum_{i=1}^{k} w_i f(x_i) / \sum_{i=1}^{k} w_i$.	E(3-4)
where $w_i \equiv \dfrac{1}{d(x_q, x_i)^2}$.	

3.5.3 Shepard's Method

Shepard's interpolation method [Shepard, 1968] is one of a variety of well-known algorithms available for multivariate scattered data interpolation [Franke and Nielson, 1980]). It is a global method, which means that it requires all points in the data set to estimate a function *f(x)* at the point *x*. Since Shepard's method is the main method that has been generalized to answer one of the subsidiary questions (see Section 1.2.2.2) in this research, we dedicate a special section in Chapter 5 to discuss the technical details of Shepard's method.

3.6 METRICS DEFINED ON THE QUERY SPACE

In Chapter 4, we propose to integrate the input space and output space as one unified space so that engineers are able to query inputs and outputs in the unified space. In many engineering applications, the unified space often involves nominal values and continuous values. One question arisen is that: "How do we define the metric on the query space particularly when the solution space is involved?" For the metric in the problem space, there is no problem. One approach is to use the standard weighted sum method [Kolodner, 1993]. For the unified space, a number of existing approaches to this problem are discussed below:

3.6.1 Value Difference Metrics (VDM)

Stanfill and Waltz (1986) introduce a value difference metric for defining distance over nominal attribute values. The original VDM algorithm uses feature weights that reflect the strength (i.e., importance) of each feature to constrain the values of goal fields. Wilson and Martinez (1997) give an account of a simplified version of VDM, which does not make use of the weighting schemes. They state that the distance measure between two attribute values depends upon how similar their correlations are with the output class. Wilson and Martinez (1997) point out that there is a problem with the VDM algorithm that is: it will not be able to cope with a new instance not yet present in the case base. For nominal values (i.e., no inherent ordering), it is not clear what needs to be done to define the distance measure for the new instance, because there is no way to determine the probability for such a value. If VDM is used on continuous attributes, a new instance is likely to cause the same problem, resulting in useless distance measures. One approach of using VDM on nominal values is to perform discretization to map the nominal values into continuous values e.g., [Cost & Salzberg, 1993]. However, discretization can cause information loss when two values at the opposite ends of the band are considered equal.

3.6.2 Heterogeneous Distance Functions

Wilson and Martinez (1997) propose three heterogeneous distance functions as alternatives to VDM. These include the Heterogeneous Value Difference Metric (HVDM), the Interpolated Value Difference Metric (IVDM) and the Windowed Value Difference Metric (WVDM). Each of these functions has been tested on the 48 datasets from the UCI machine learning databases using 10-fold cross validation [Witten & Frank, 2000]. HVDM uses two functions: the Euclidean distance function for continuous attributes and the VDM function for nominal values. They point out that the use of normalization scheme has an effect on the generalization accuracy due to the different type of measurements used for the continuous values and nominal values. Results show that HVDM achieves higher generalization accuracy than the Euclidean distance function. In the IVDM distance function, VDM can be applied directly to continuous attributes without the need for a normalization scheme.

Continuous values are discretized into a number of regular-spaced intervals. The IVDM approach is to use interpolation between the midpoints of two consecutive discretized ranges to alleviates the information loss problem caused by discretization (i.e., when two values at the opposite end of the range are considered equal). The WVDM interpolates probabilities from adjacent values rather than from range midpoints. Here adjacent values mean those values in the range of the attribute value x +/- $w_a/2$ where w_a is the window width determined by the discretized range. Overall results show that IVDM achieves highest average classification accuracy followed by WVDM and subsequently HVDM and Euclidean distance.

3.6.3 Distance between Cluster Centres

The K-means method is an iterative process that assigns instances to a specified number of non-overlapping clusters according to their closest cluster centre in Euclidean distance. This process is repeated through the data until it successfully clusters all instances in the data set. [Woon *et al.*, 2003a] uses cluster centre distance to represent the distances between three distinct Iris class: {Iris-setosa, Iris-versicolour, Iris-virginica} in the Iris data set [Fisher, 1936]. There are four attributes in the Iris data set: {sepal width, sepal length, petal width, petal length}. In their approach, the distance in the problem space between 2 instances is computed using the weighted sum method [Kolodner, 1993] with equal weights. The solution space metric is defined using the cluster centre distance.

3.6.4 Material Functions

Mejasson *et al.* (2001) describe an intelligent design assistant (IDA) CBR system to assist engineers for materials selection in the submarine cable industry. They define the distance between two values of a component material search index based on the following methods: comparison on qualitative scale, comparison on quantitative scale and comparison based on placement in an abstraction hierarchy. In their project, comparison on qualitative scales uses a three-point or five-point scale (e.g., very low, low, medium, high and very high). Quantitative scales may be defined using a distance function over a two-dimensional space.

Comparison based on placement in an abstraction hierarchy is for indices that inherit a natural hierarchy such as component material indices, which uses standard materials classification and component types (e.g., mechanical and electrical).

3.7 CASE SELECTION TECHNIQUES

Case selection techniques have been investigated in the guise of both case-base maintenance and instance-based learning (IBL). Mitchell (1997) states that IBL and CBR share some common properties: (1) they implement "lazy" learning, (2) they classify new instances by nearest neighbour methods.

3.7.1 Instance-Based Learning (IBL) Reduction Methods

A wide range of such algorithms have been developed for Instance-Based Learning: for example, CNN [Hart, 1968], RNN [Gates, 1972], Shrink [Kibler and Aha, 1987], IBL [Aha, 1992; Aha *et al.* 1991], DROP [Wilson and Martinez, 2000], Explore [Cameron-Jones, 1995], etc. Generally, these reduction algorithms can be divided into three main categories: Incremental, Decremental and Batch [Wilson and Martinez, 2000]; and other growing and reducing reduction methods such as Explore, ELGROW [Cameron-Jones, 1995], etc. In this thesis, we concentrate on decremental methods to investigate the feasibility of using Generalized Shepard Nearest Neighbour (GSNN) (please see Chapter 5 for the detail of GSNN) in substitution for other nearest neighbour methods in case base reduction.

3.7.2 The Shrink algorithm

The Shrink algorithm [Kibler and Aha, 1987] is a decremental reduction method. The method finds an optimum sparse case base in a decremental manner. In the beginning, the case base is set to be the initial set of cases. A reduced case base is produced with one less case iteratively. The algorithm removing cases based on the contribution of each case to the predictive power of the case base. Whether a case is to be removed is decided upon by first obtaining a candidate set from the current reduced case base. The reduction mechanism can be implemented by using a nearest neighbour algorithm such as k-NN and DWNN to obtain an estimated solution for each case in the reduced set. Cases that are correctly predicted will be removed. In the case where more than one case in the reduced set is correctly predicted, a case is removed at random for that iteration. The technical details of how the Shrink algorithm works is discussed in Chapter 7.

3.7.3 A Reduction Method based on Rough Set Theory

Salamo and Golobardes (2002) have invented a competence model that uses deletion and building techniques based on rough set theory in classification tasks. In their work, a set of equivalence relations are used to determine the core and reduct of knowledge (i.e. a set of cases) that can then be used for filtering unwanted cases. They propose a set of frameworks that work on different domains to examine whether to keep the boundary cases or create new cases in order to produce an optimum case base. Many results have been published in their work in comparison with other reduction methods such as the IBL2-4 algorithms [Aha, 1992; Aha *et al.* 1991], CNN [Hart, 1968], DROP1-5 [Wilson and Martinez, 2000], etc. on some data sets.

3.7.4 Case Base Maintenance Methods

Yang and Zhu (2001) address the problem by proposing a method that selects an optimal case base from a given set of cases, based on nearest neighbour retrieval. In their paper, no adaptation is considered and hence the nearest neighbour solution is applied to the target problem. Smyth *et al.* [Smyth and Keane, 1995; Smyth and McKenna, 1998, 1999] have also addressed similar problems by means of an algorithm based on a calculation of 'relative competence' of cases. Relative competence is used to measure the importance of cases, and consequently provides a means to optimize case base performance. Smyth and Keane point out that the relative competence of cases depends upon both retrieval and adaptation.

PART II

CORE IDEA OF THE RESEARCH

Chapter 4

Unified Problem:Solution Space in Case-Based Reasoning

Chapter 5

Interpolation over Nominal Values

Chapter 6

Interpolation on a Diverse Retrieval Set

Chapter 7

Case Selection Techniques

Chapter 4

Unified Problem:Solution Space in Case-Based Reasoning

4.1 INTRODUCTION

This chapter aims to answer the following subsidiary research question (see Section 1.2.2.1):

Can the standard CBR models be generalized to a unified (problem:solution) space to allow flexible query modes?

In the standard Case-Based Reasoning (CBR) model, a case is represented as a <problem, solution> pair. The problem space and solution space are treated as separate, and nearest neighbours are retrieved using a metric defined on the problem space. The standard method applies to domains where the similarity assumption is valid: that cases which are near in the problem space are also near in the solution space. This presumes that a metric is also defined in the solution space although it is not explicitly stated, and that the problem:solution space mapping is singled-valued. However, this assumption is not necessarily true for inverse problems, in which single-valued solution cannot be assumed. We have discussed in Section 2.5.2 that the constraint problem and the multi-valued solutions mapping problem can cause poor prediction when adaptation is performed on retrieved cases that are not close in the unified problem:solution space. Cases far apart are likely to give poor prediction in interpolation. To fully exploit the interpolation scheme we require a nearest neighbour method that retrieves cases that are close in the unified space. (Note: this is different to the idea conveyed in Chapter 6, which claims that a diversity algorithm can improve interpolation systems. In Chapter 6, the diversity algorithm is implemented to avoid the extrapolation trap in which retrieved cases are close together and yet the target is outside the interpolation points. Extrapolation is less likely to give good prediction as interpolation.)

In this chapter a generalization of the standard CBR retrieval method, which integrates solution space and problem space into a single query space is proposed. The retrieval method is proposed by means of the concept of nearest neighbours to a constraint region, which can be any subset of the unified space. Under this retrieval scheme, cases are treated as data points in the unified space so that a query can take the general flexible form of a database predicate. Engineers can query on any subset of the unified space and add constraints dynamically to the predicate. Query predicate on the unified space corresponds to a constraint region. Nearest neighbours to the predicated region are defined as those with the minimum distance to the region.

In fact, the standard CBR retrieval is a special case of this more general model. The advantages of the general model are associated with the nature of its more general query modes. In contrast to the standard model, which can only be queried by specification of problem "inputs", the general model is capable of retrieving general queries on the unified problem-solution space. The advantages of this flexible query form are explained in this chapter, by means of a variety of illustrative examples.

We have introduced the main idea of the method in this section. In Section 4.2 we present the theoretical basis for the method and show that the standard nearest neighbour method is a special case of the unified general method. Section 4.3 provides a short discussion on definition of metrics on the query space. Section 4.4 we give an illustrative example of the method as applied to a simple "inverse" problem. In Section 4.5.1 we continue to examine the method further with constraints over both inputs and outputs and in Section 4.5.2 we examine the method with additional constraints such as derived attributes added by engineers. In Section 4.6 we discuss the importance of such a general retrieval method and that it would benefits the adaptation scheme discussed in the CBE architecture (see Section 2.5.1). We conclude in Section 4.7 with a summary of the advantages of the generalisation of the nearest neighbour retrieval developed in this chapter.

4.2 GENERALIZATION OF NEAREST NEIGHBOUR RETRIEVAL (GNNR) OVER THE N-DIMENSIONAL UNIFIED SPACE

4.2.1 Generalization of Nearest Neighbour Retrieval over the Unified Space

In this section, we present the basis for retrieval of nearest neighbours to a constrained region over a unified n-dimensional space. We take the space to be formed from n one-dimensional domains: D_i , $i = 1, \ldots, n$, each with a distance function defined. If a_i and $b_i \in D_i$, we take $|a_i - b_i|$ to be the distance between elements a_i and b_i. We let D denote the n-dimensional space $D_1 \otimes D_2 \otimes D_3 \otimes \ldots D_n$, and let $(z_1, z_2, \ldots, z_n)$ represent a point in D, where $z_1 \in D_1$, $z_2 \in D_2$, ... , etc.

If $a, b \in D$ we assume $d(a,b)$ represents a distance function defined over D, such that

$$d(a,b) > 0, \ \forall \ a, b \ \ if\, a \neq b . \qquad E(4\text{-}1)$$

$$d(a,a) = 0, \ \forall \ a \ . \qquad E(4\text{-}2)$$

$$d(a,b) = d(b,a), \ \forall \ a, b \ . \qquad E(4\text{-}3)$$

$$d(a,b) + d(b,c) \geq d(a,c), \ \forall \ a, b, c \ . \qquad E(4\text{-}4)$$

The existence of a metric d on the unified space, which satisfies E(4-4) implies that the usual similarity assumption [Kolodner, 1996] holds true.
Normally we take d to be the pseudo-Euclidean metric:

$$d(a,b) = [(w_1|a_1 - b_1|^2 + w_2|a_2 - b_2|^2 + \ldots + w_n|a_n - b_n|^2) / \sum_{i=1,\ldots,n} w_i]^{1/2} \ .$$

where $w_i > 0$, $i = 1, 2, \ldots, n, > 0$. In many cases a user needs to specify a region R and we need to find its nearest neighbours within a given case base $C \subset D$. We first extend the definition of metric to include the distance between a point and a region. Let $R \subset D$ be a region in the n-dimensional space. We define:

$$d(a,R) = min_{\, r \in R} \, d(a,r) \qquad E(4\text{-}5)$$

Now we apply the definition E(4-5) to cases in C. For a given region R, we define the nearest neighbour in C as

$$Arg\ min_{\ c\ \in C}\ d(c,R) \qquad \text{E(4-6)}$$

(Notice there is symmetry here between R and C). Combining E(4-5) and E(4-6) we get:

$$Argmin_{\ c\ \in C}\ min_{\ r \in R}\ d(a,r) \qquad \text{E(4-7)}$$

for the nearest neighbour in C to the region R.

4.2.2 The Standard CBR Model as a Special Case

The above analysis treats the problem space and the solution space equally. However, in standard case base analysis, it is usual to divide the space into two parts: the problem space, which is completely known and specified as a target $(x_1, x_2, ..., x_k)$, and the solution space $(y_1, y_2, ..., y_m)$ which is unknown. To apply the general theory of the previous section, we consider the n dimensional space $(z_1, z_2, ..., z_n)$ where $n = k+m$ and

$$z_1 = x_1,\ ...,\ z_k = x_k,\ z_{k+1} = y_1,\ ...,\ z_n = y_m.$$

The user specifies the target region R by the k constraints (see Fig. 4-1):

$$r \in R \text{ iff } r_1 = x_1^{target},\ r_2 = x_2^{target},\ ...,\ r_k = x_k^{target}.$$

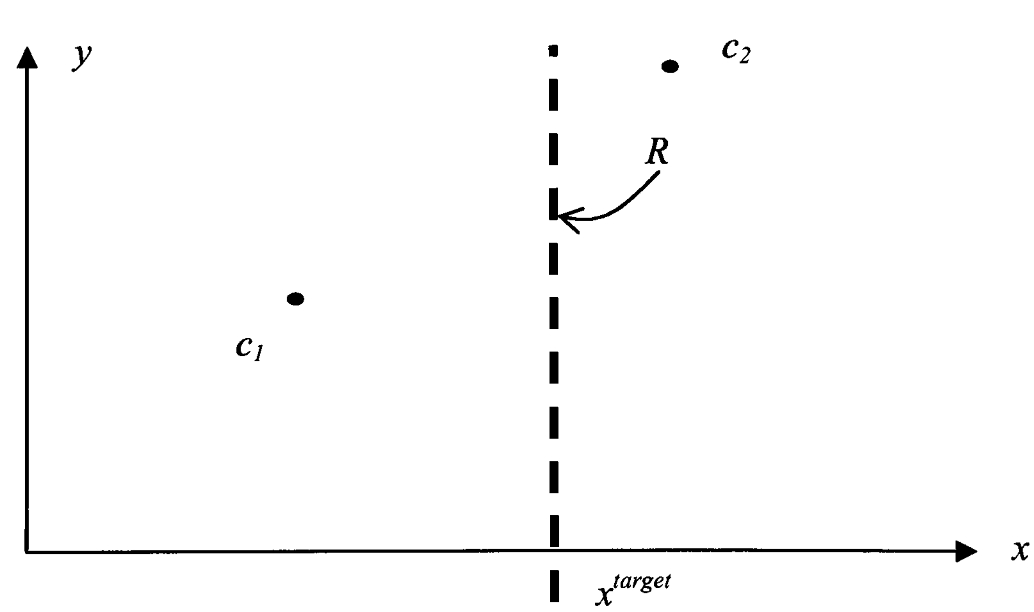


Figure 4-1. Visualization of a target case, x^{target} as a constrained region R

For a given point c in C and an element r in R, by taking $d()$ as the pseudo-Euclidean metric, we have

$$d(c,r) = [(w_1|c_1 - r_1|^2 + w_2|c_2 - r_2|^2 + \ldots + w_n|c_n - r_n|^2) / \sum_{i=1,\ldots,n} w_i]^{1/2}$$

For $i = k+1, \ldots, n$, there is no constraint on r_i, so we can choose $r_i = c_i$, whereupon $|c_i - r_i| = 0$ for a minimum $d(c,r)$. For $i = 1, \ldots, k$, $r_i = x_i^{target}$. E(4-7) gives

$$Argmin_{\ c \in C}\ [(w_1|c_1 - x_1^{target}|^2 + \ldots + w_k|c_k - x_k^{target}|^2) / \sum_{i=1,\ldots,k} w_i]^{1/2}$$

which is the standard nearest neighbour formula, relying on a distance metric defined only on the problem space.

4.3 DEFINITION OF METRICS ON THE QUERY SPACE

The similarity assumption applies to standard CBR models such that similar problems give similar solutions. Such assumption presumes that a metric is defined in the solution space as it is in the problem space although nobody has explicitly stated that a distance metric must exist. The old solution is retrieved based on the retrieved case that is close to the target query in the problem space. In order to use the general method proposed in this chapter, we need a definition for metric on the unified query space. One question arisen is that: "How do we define the metric on the query space particularly when the solution space is involved?". One approach is to use the standard weighted sum method [Kolodner, 1993]. However, it is not straight forward because often the solution space involves nominal values. Section 3.6 reports several approaches (e.g., [Stanfill and Waltz, 1986], [Wilson and Martinez, 1997] and [Woon *et al.*, 2003a]) to define metrics on the query space especially when nominal values are involved.

In this chapter, to demonstrate how GNNR works over the unified space, we use the weighted sum method to compute the distance between cases to a target query for the illustrative example: the projectile model in Section 4.4 and 4.5.

4.4 GNNR APPLICATION FOR AN INVERSE PROBLEM

The approach discussed in Section 4.2.1 is obviously symmetric between x and y, i.e., problem and solution domain have no significance in this generalized approach. If the user specifies the target region R by the m constraints:

$$r \in R \text{ iff } r_1 = y_1^{target},\ r_2 = y_2^{target},\ \ldots,\ r_m = y_m^{target}.$$

The problem reduces to the nearest neighbour problem with a metric defined over the y-domain. To illustrate how the method works for an inverse problem, we refer to the projectile problem described in Section 2.2.

Consider a gunner tries to hit the target at x=450m on the level ground (see Fig. 4-2). We want to decide on the right angle of gun θ and the initial velocity v to achieve that goal. Assume that there are various guns available, each with known muzzle velocity. Using GNNR, we can formulate the problem as follows: take R to be the space of points $r(r_1, r_2, r_3)$ where $r_1 = \theta$, $r_2 = v$, r_3 = the x value of the point where the cannon ball hits the level ground. R is the region subject to the constraint: $r_3 = 450m$.

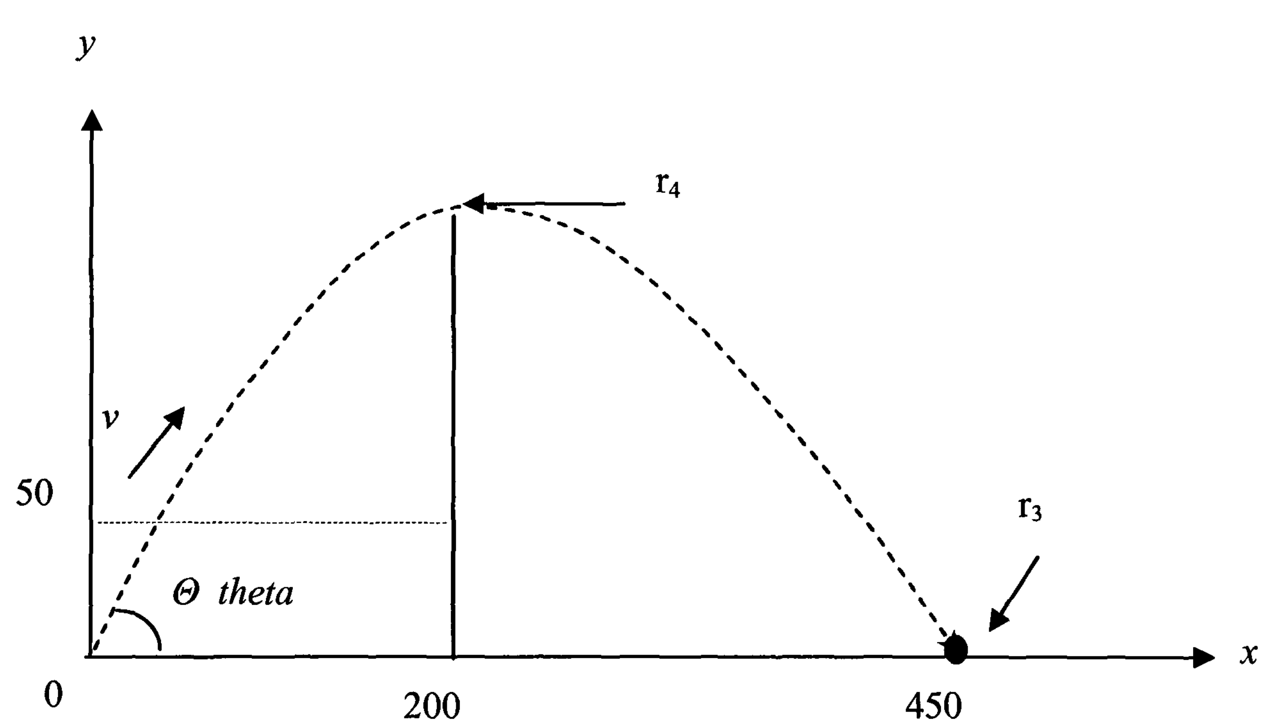


Figure 4-2. Visualization of the trajectory of a cannonball shot over flat ground

In the generalized approach, each case in the case base is represented by $c(c_1 = \theta, c_2 = v, c_3 = x)$, and the metric is:

$$d(c,r) = [(w_1|c_1 - r_1|^2 + w_2|c_2 - r_2|^2 + w_3|c_3 - r_3|^2) / \sum_{i=1,\dots,n} w_i]^{1/2} \ .$$

As there are no constraints on r_1 and r_2, *the minimum $d(c,r)$ will occur when $r_1 = c_1$*, so that $|c_1 - r_1| = 0$. Similarly, $|c_2 - r_2| = 0$. Hence E(4-7) reduces to

$$Arg\ min_{\,c \in C}\ [\, w_3 \mid c_3 - 450 \mid^2 / w_3]^{1/2} = Arg\ min_{\,c \in C}\ \mid c_3 - 450 \mid .$$

We are searching for cases with c_3 near to *450m.* Table 4-1 shows examples of cases, i.e., $c(c_1, c_2, c_3)$, generated in a case base with $\theta \in (0, \pi/2]$ and velocity $v \in [70, 100]$:

Table 4-1. Example cases presented in the case base and the distance calculated between each case and the target region *R*

Case ID	c_1	c_2 (m/s)	c_3 (m)	d (c,R)
29	1	70	445.556	0.00451
51	0.4	80	459.108	0.00924
19	1.2	80	432.296	0.018
38	0.8	70	489.791	0.0403
56	0.2	100	389.418	0.0614
35	0.8	60	359.847	0.0914

With this approach, the retrieved cases for k=3 nearest neighbours are:

Case 29: (1, 70, 445.556),

Case 51: (0.4, 80, 459.108), and

Case 19: (1.2, 80, 432.296).

As case 29 gives minimum $d(c,R) = 0.00451$, according to E(4-7) , the best match to the target region R is case 29. This analysis shows how the generalized approach can be used to retrieve solutions, which may then be adapted to solve an inverse problem. In fact, GNNR can also be used for a direct problem: given θ, and v, what is the value of x? Here we show that such a general query mode is indeed very powerful because it allows engineers to query the numerical model in any direction without concerning whether the variables are inputs or outputs. In addition, engineers can use the CBR model to explore alternative combinations of inputs that will give the same or similar output. In this example, the other alternative for the engineer to derive a near output of c_3*=450m* is Case 51. In the next section, we show examples of query constrained by both inputs and outputs, and query constrained by additional derived values.

4.5 GNNR APPLICATION FOR CONSTRAINT PROBLEMS

Following the inverse problem example in Section 4.4, we now examine how GNNR behaves by adding more constraints on inputs and outputs.

4.5.1 Constrained on Inputs and Outputs

Consider the gunner who tries to hit the target at x = 450m on the level ground may want to restrict the choice of gun with initial velocity = 80m/s. The problem is now represented by a target region R involving constraints on input v and output x: $r_2 = 80$ *m/s and,* $r_3 = 450m$. As there are no constraints on r_1, the minimum $d(c,r)$ will occur when $r_1 = c_1$, so that $|c_1 - r_1| = 0$. Hence E(4-7) reduces to

$$Argmin_{\ c \in C} [(w_1|c_1 - r_1|^2 + w_2|c_2 - 80|^2 + w_3|c_3 - 450|^2) / \sum_{i=1,\dots,3} w_i]^{1/2}$$

$$= Argmin_{\ c \in C} [w_2|c_2 - 80|^2 + w_3|c_3 - 450|^2) / \sum_{i=2,3} w_i]^{1/2}$$

Now we are searching for cases with c_2 near to *80m/s* and c_3 near to *450m.* In this problem we take equal weights for w_2 and w_3 assuming that r_2 and r_3 are equally important. Table 4-2 shows examples of cases, i.e., $c(c_1,c_2,c_3)$, generated in the case base with angle $\theta \in (0 , \pi/2]$ and velocity $v \in [70,100]$:

Table 4-2. Example cases presented in the case base and the distance calculated between each case and the target region *R* – constrained on inputs and outputs

Case ID	c_1	c_2 (m/s)	c_3 (m)	d (c,R)
51	0.4	80	459.108	0.00462
19	1.2	80	432.296	0.00898
29	1	70	445.556	0.0648
30	1	80	581.950	0.0669
46	0.6	80	596.505	0.0743
38	0.8	70	489.791	0.0827

Using GNNR, the best match to the constraint problem for k=3 nearest neighbours can be determined as follows:

Case 51: (0.4, 80, 459.108),

Case 19: (1.2, 80, 432.296), and

Case 29: (1, 70, 445.556).

As case 51 gives minimum $d(c,R) = 0.00462$, according to E(4-7), the best match to the target region R is case 51. Because GNNR is a general method that treats inputs as no difference to outputs, it allows engineers to query the projectile model with input and output constraints.

4.5.2 Constrained on Derived Attributes

We now introduce a new constraint to the example in Section 4.5.1. Suppose that there is an obstacle at distance 200m, which must be cleared by the cannon ball. The gunner tries to hit the target at x = 450m on level ground with no constraint on the initial velocity of the cannon ball and the new constraint of the obstacle with height 50m at distance 200m. To handle this newly added constraint, we augment the space to include a derived dimension, r_4 = the height of the cannon ball at the obstacle, when x = 200m.

By introducing this new constraint, the target region R is defined by: r_3 = 450m, and r_4 > 50m. We take the distance metric on the new dimension associated with the new constraint to be:

$$d(a_4,b_4) = \quad |a_4-b_4|$$

With the generalized approach we reformulate the problem so that the metric bears a penalty term $w_4\, d^2(c_4, r_4)$ for cases outside the constraint region:

$$d(c,r) = [(w_1|c_1 - r_1|^2 + w_2|c_2 - r_2|^2 + w_3|c_3 - r_3|^2 + w_4 d^2(c_4, r_4)) / \sum_{i=1,\dots,n} w_i]^{1/2}$$

For $c \in C$ and $c_4 <= 50$ and since no constraint is imposed on r_1 *and* r_2, $d(c, R) = min_{r \in R}\, d(c,r)$

$$= \quad [(w_3|c_3 - 450|^2 + w_4|c_4 - 50|^2) / \sum_{i=1,\dots,4} w_i]^{1/2}$$

For $c \in C$ and $c_4 > 50$ and since no constraint is imposed on r_1 *and* r_2, $d(c, R) = min_{r \in R}\, d(c,r)$

$$= \quad [(w_3|c_3 - 450|^2 / \sum_{i=1,\dots,4} w_i]^{1/2}$$

Hence for $c \in C$,

$$d(c, R) = min_{r \in R}\, d(c,r) = \quad [(w_3|c_3 - 450|^2 + w_4(50 - c_4)\max(0, 50 - c_4)) / \sum_{i=1,\dots,4} w_i]^{1/2}$$

Following E(4-7) we have $Argmin_{c \in C}\, d(c, R)$

$$= Argmin_{c \in C} [(w_3|c_3 - 450|^2 + w_4(50 - c_4)\max(0, 50 - c_4)) / \sum_{i=1,\dots,4} w_i]^{1/2}$$

Here, we are searching for cases with c_3 near to *450m* and c_4 greater than *50m.* For this problem, we take equal weights for all constraints. Table 4-3 shows examples of cases, i.e., $c\,(c_1, c_2, c_3, c_4)$, generated in the case base with $\theta \in (0, \pi/2]$ and velocity $v \in [70,100]$:

Table 4-3. Example cases presented in the case base and the distance calculated between each case and the target region *R* – constrained on derived attributes

Case ID	c_1	c_2(m/s)	c_3 (m)	c_4 (m)	d (c,R)
29	1	70	445.556	171.664	0.00225
51	0.4	80	459.108	47.72	0.00469
19	1.2	80	432.296	276.431	0.00898
38	0.8	70	489.791	121.840	0.0202
56	0.2	100	389.418	19.720	0.0316
35	0.8	60	359.846	91.475	0.0457

Using GNNR, the retrieved cases for k=3 nearest neighbours are:

Case 29: (1, 70, 445.6, 171.7),

Case 51: (0.4, 80, 459.108, 47.72), and

Case 19: (1.2, 80, 432.296, 276.431).

Although Case 29, Case 51 and Case 19 do not fully satisfy the constraints they may be useful for adaptation. The retrieved cases need not be an exact match to the target. It would be helpful for the case base to provide an indicator that guides the engineer to find an optimum solution. Using the GNNR method, engineers can add derived attributes to the query predicate easily. Without this flexible approach, it might take considerable time and effort for engineers to reconstruct the projectile model for the inclusion of new constraints. In the next section, we consider how GNNR benefits the adaptation scheme discussed in the CBE architecture (see Section 2.5.1).

4.6 THE MULTI-VALUED CASE MAPPING AND INTERPOLATION PROBLEM

Another motivation for the GNNR method actually comes from the multi-valued case mapping and interpolation problem discussed in Section 2.5.2.2. For inverse problems, there are often many solutions to the problem. Sometimes these solutions are far apart from each other in their input space. Cases far apart in a case base may produce poor prediction when these cases are used in an interpolation scheme. To make sure that we take advantage of the interpolation scheme, it is desirable to impose restriction over cases that are used for interpolation. For example, cases that are reasonably close together may preferably be chosen among the qualified candidates in the case base. We here call good candidate cases for interpolation conformable. Ideally when adaptation is required, the CBR model would provide useful advice in selecting good candidates for interpolation. Without taking precaution against poor candidates, interpolation methods may not be fully exploited for prediction accuracy.

To illustrate this further, we can take the inverse example in Section 4.4. There are two cases – Case 29 and Case 51 that produce c_3 near to the target (c_3 = 450m). Nevertheless, Case 29 and Case 51 are not close in the input space. We note that Case 38, which however is far apart from the target, could be a good interpolation candidate paired with Case 29. To highlight this interpolation dilemma and show how the unified space can benefit the adaptation scheme in the CBE architecture (see Section 2.5.1 Step 2), we provide two case study examples each in Chapter 8 – CBE-Projectile and Chapter 9 – CBE-Conveyor.

4.7 CONCLUDING REMARKS

In this chapter we have presented a generalized version of CBR model based on a unified (problem: solution) space. The nearest neighbour retrieval method has been re-cast as nearest neighbour to a constrained region. The region may be defined by means of a predicate defined on the unified space. This allows the user greater flexibility in specifying any query over the unified space. The first three illustrative examples in Section 4.4, 4.5.1 and 4.5.2 have demonstrated the benefit of the flexible approach of the unified space and GNNR to inverse and constraint problems. The inverse problem example has shown that the general method allows engineers to query the numerical model in any direction (direct and inverse). The first constraint example in Section 4.5.1 has illustrated that engineers can specify constraints both on inputs and outputs. The second constraint example in Section 4.5.2 has further shown the merit of the flexible generalized method that engineers can specify with little difficulties additional constraints such as derived attributes to the query predicate which otherwise might require reconstruction of the numerical model for the inclusion of such constraints.

The importance of the unified space concept is also discussed in relation with the adaptation scheme in the CBE architecture. The unified space can be a better measure than standard query space (i.e., problem) for finding a set of good candidate cases (conformable cases) for interpolation in engineering domains where multiple solutions are assumed. It will be shown in the chapters later on in this study that the GNNR method paired with the concept of conformability are of great help to fulfill the full potentials of interpolation systems such as GSNN.

Chapter 5

Interpolation over Nominal Values

5.1 INTRODUCTION

To answer the subsidiary question in Section 1.2.2.2:

"Can distance weighted interpolation be generalized to the unified problem:solution space, particularly with regard to nominal values?",

we present in this chapter a Case-Based Reasoning (CBR) adaptation method that utilizes a generalized interpolation method — Generalized Shepard Nearest Neighbour (GSNN) — applying equally to nominal and continuous values. The motivation for such a method is that we would like to extend a powerful interpolative method, already proven in the real domain, so that it is equally efficient in the domain of nominal values. Interpolative methods are well studied in the real domain, and can give good results from relatively sparse datasets. Case bases of many CBR systems are often sparse as they are difficult or expensive to obtain. When the solution space for such systems is continuous, interpolative methods can be used to increase the accuracy of the solutions. However, no general interpolative method exists for nominal (discrete) solution domains.

Chatterjee and Campbell (1993, 1999) have considered adaptation through interpolation in the context of the efficiency and timeliness of CBR. They consider three primary tasks for efficient implementation of case adaptation which arises when nominal (symbolic) values are present. These are:

- Interpolation in the domain of nominal values in the problem domain
- Ordering of nominal values in the solution domain
- Interpolation in the domain of nominal values in the solution domain

The second of these tasks is really connected with the approach to interpolation adopted by Chatterjee and Campbell which depends upon a prior ordering of any nominal domains present in the application. The assumption is that a distance metric $d_x(x, x_i)$ is defined on the problem domain X and $d_y(y, y_i)$ on the solution domain Y. Interpolation assumes a linear distance metric in the solution domain, by which there exists a mapping $D: y\text{->}R$ from solution points onto the real line, such that $d_y(y_1, y_2) = | D(y_1)-D(y_2) |$ (see Fig. 5-1). Fig. 5-1 shows each y value on the real line is mapped to a discrete band: {Band A, Band B, Band C, Band D}. The interpolation estimates $\hat{y}$ for a target value, and x^{target} is mapped to Band C. Chatterjee and Campbell give a variety of ways to calculate the mapping *D*. One such is to order the space somehow, and take *D* to be the mapping to the rank order. Although most real valued interpolation methods (such as the Shepard's method [Shepard, 1968]) also make the assumption of a linear distance metric in the solution domain when the solution domain is continuous, *D* is usually the identity mapping, so that $d_y(y_1, y_2) = | y_1\text{-}y_2 |$.

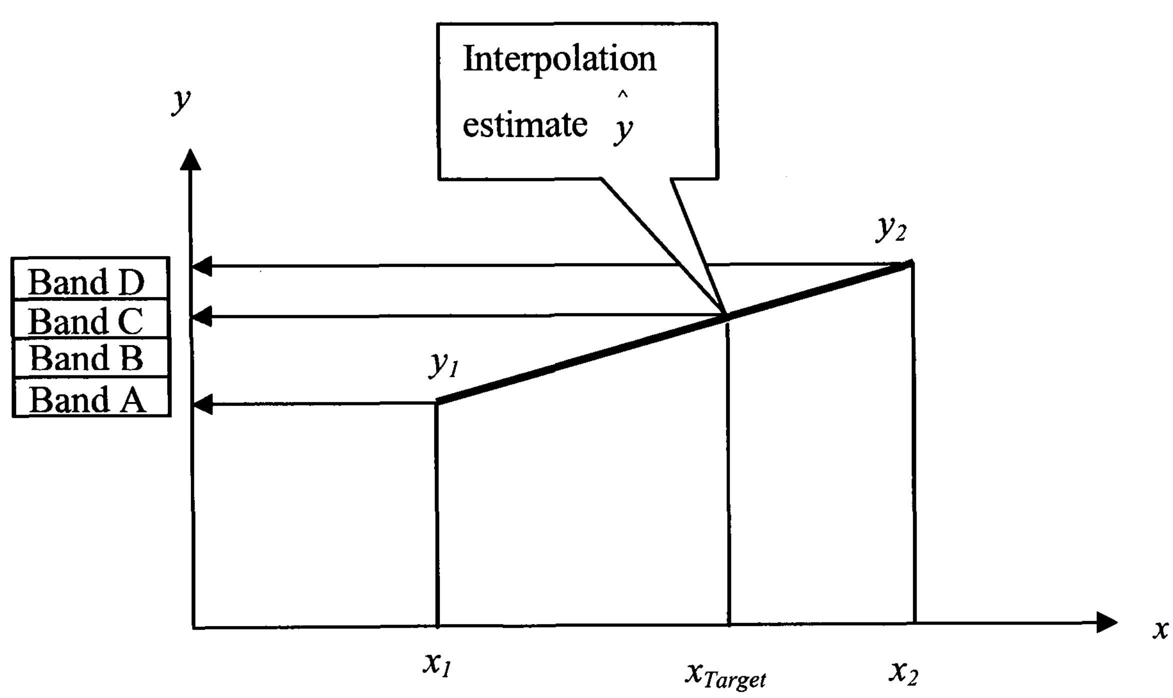


Figure 5-1. Chatterjee and Campbell's interpolation method

One problem with this approach is that it is not always easy to identify a natural ordering of the space. In many cases, CBR applications have more complex solution spaces, showing higher dimensionality, or indeed no embedding space at all. In some CBR systems, symbols are naturally mapped to taxonomic hierarchies. In other CBR systems, such as in the casting design system CASTAID [Knight *et al.*, 1995; 2000] solutions are represented by shape-graphs. Also there may be nominal solution values where there is no natural ordering of such space. These cases are not amenable to the methods proposed by Chatterjee and Campbell, but can be dealt with by the GSNN method proposed in this chapter.

The *k*-Nearest Neighbours (*k*-NN) algorithm [Cover and Hart, 1967; Mitchell, 1997] and Distance-Weighted Nearest Neighbour method (DWNN) have been the primary methods used in classification. For nominal domains, *k*-NN predicts the solution for a given target query based on the output classification of its *k*-nearest neighbours under the assumption that the output class of the target query is most similar to the output class of its nearby instances in Euclidean distance. Comparatively, DWNN weights the contribution of each of the *k*-nearest neighbours to the prediction according to its distance to the target query.

Both *k*-NN and DWNN operates on a voting mechanism (see Section 3.5.2.1 and Section 3.5.2.2). Under this mechanism, the predicted solution must exist in one of the nearest neighbours in the retrieval set. It is not possible for these methods to return intermediate values not present in the retrieved set Y. This property is not shared by most interpolation formulae over a continuous real domain, which typically return real values NOT in the retrieved set; for example, interpolating as the average of the retrieved set Y $=\{1,2\}$ produces the solution 1.5 which isn't in the set Y.

This characteristic of *k*-NN and DWNN over nominal solution domains has a number of disadvantages for CBR applications. The first of these relates to the performance of CBR systems in predictive classification. For many retrieval situations the retrieved set may not contain the target nominal value at all; particularly if the target is in a region of the case base where cases are sparse. This can in fact degrade the predictive performance of the retrieved set, as we see in the examples given in sections 5.5 and 5.7.1 of this chapter.

There is also a problem involved when new solutions are added to a case base. When a new solution is first added, there may be very few cases, or even no cases at all present in the case base. In this situation the new solution will rarely, if ever, be retrieved into the set for interpolation. In the example of the travel case base discussed in Section 5.7, a new hotel may be added for which there are no extant holidays. This hotel will never be retrieved by *k*-NN or DWNN as a solution for a target holiday.

The importance of interpolation over nominal domains has been discussed by Wilson and Martinez (1997). They provide an account of approaches to defining metrics over nominal domains, and propose the use of an extension to Stanfill and Waltz's Value Difference Metric (1986) as the basis for interpolation. There is a problem with their method too, that is: it is not capable of coping with solutions not in the case base.

In this chapter, we propose a generalization of Shepard's method [Shepard, 1968] which is somewhat similar to DWNN, but which depends upon the minimization of a distance weighted error function, rather than the usual maximization of a voting function. We show that this approach has advantages over existing interpolative methods such as k-Nearest Neighbours (*k*-NN) and Distance Weighted Nearest Neighbours (DWNN). We term the retrieval algorithm the Generalised Shepard Nearest Neighbour (GSNN) method. GSNN applies in exactly the same way to nominal and real value domains. As such, in contrast to *k*-NN and DWNN, it does not require the solution value to be contained in the retrieved set, or indeed within the case base at all.

In Section 5.2 we describe Shepard's interpolation method, and in Section 5.3 explain the generalization to nominal values. In Section 5.4 we discuss the general properties of GSNN, and in Section 5.5 illustrate the procedure of the use of GSNN with reference to the well-known Irises classification problem [Fisher, 1936]. The Irises problem is good for illustration purposes, since it demonstrates how 28/50 instances of iris-setosa can be correctly predicted by GSNN from a case base of just 2 cases: 1 of iris-virginica and 1 of iris-versicolor. This illustrates clearly how solutions not in the case base can be correctly interpolated. In Section 5.6 we give a simulated example, taken from a benchmark used by Ramos & Enright [Ramos and Enright, 2001]. This is used to measure the performance improvement by GSNN on random and regular case bases. In Section 5.7, we use the travel benchmark case base [Lenz *et al.*, 1996] to test both the improvement in performance effected by GSNN and its ability to deal with the addition of new solutions to a case base. We conclude in Section 5.8, with a summary and indications of future work.

5.2 SHEPARD'S METHOD

Shepard's interpolation method [Shepard, 1968] is one of a variety of well-known algorithms available for multivariate scattered data interpolation, where the independent variable $x \in R^d$, and the dependent variable $y \in R$ (see e.g., [Franke and Nielson, 1980]). We choose Shepard's method, since, as noted by Mitchell (1997), in the case of continuous domains it is the global form of the DWNN. Shepard's method is global, requiring all points in a dataset to estimate a function $f(x)$ at the point x. The interpolation function is given by:

$$f(x) = \Sigma_1^n \| x - x_i \|^{-p} f(x_i) \quad / \quad \Sigma_1^n \| x - x_i \|^{-p} . \quad \text{E(5-1)}$$

where the interpolation is over the set of n points $\{x_1, x_2, \ldots, x_n\}$, $p > 0$ and $\|x - x_i\|$ denotes the Euclidean distance in R^d.

Lazzaro and Montefusco (2002) have pointed out that this scheme has the advantage of full independence from the space dimension but disadvantages of low reproduction quality and high computational cost. By independence from the space dimension we mean that the d dimensional values of the point $x \in R^d$ do not enter into E(5-1) themselves, but only the distances between points x and x_i. This independence from the space dimension makes the scheme of interest in CBR applications and we show in Section 5.3 that we can generalize the method to operate only on distances. Often in CBR, we do not necessarily have embedding dimensions, as we shall show in Section 5.5.

The global nature of the function and its concomitant high computational cost make a local form of the method more suitable. Franke and Nielson (1980) have proposed a modification of the Shepard's method which presents improved reproduction quality and reduced complexity. In their method, known as the modified quadratic Shepard's method, the influence of each data point in the data set is confined to interpolation points within a radius of the data point.

However, for case based reasoning, the global nature of the Shepard's method does not present a problem since interpolation is only conducted over a small set of retrieved data points. The retrieval phase of the cycle has effected the localization of the method. In effect the retrieval phase ensures that only data points within a radius of the interpolation point will influence the interpolation value.

5.3 GENERALISED SHEPARD NEAREST NEIGHBOUR (GSNN) METHOD

The objective of this chapter is to present an adaptation method which will apply to a general class of both problem and solution domains. The assumption here is that a distance metric $d_x(x, x_i)$ is defined on the problem domain X and $d_y(y, y_i)$ on the solution domain Y.

As discussed earlier, Shepard's method applies to domains where $X = R^d$ and $Y = R^1$. We would like to apply it to more general domains, wherever a distance metric may be defined. Fortunately, the method is already independent of the space R^d, inheriting only the Euclidean distance $||x - x_i||$ from R^d. This is generalized to $d_x(x, x_i)$ over X.

We first need to express Shepard's interpolation in the problem space Y in terms of $d_y(y, y_i)$. Let $y=f(x)$ and $I(y) = \Sigma_1^n || y - y_i ||^2 || x - x_i ||^{-p} / \Sigma_1^n || x - x_i ||^{-p}$. It is noted that

$$\partial I / \partial y = 2\sum_1^n (y - y_i) \ \ || x - x_i ||^{-p} / \sum_1^n || x - x_i ||^{-p}$$

$$= 0 \quad \textit{iff} \quad y\sum_1^n || x - x_i ||^{-p} \ = \ \sum_1^n || x - x_i ||^{-p} \ y_i$$

And

$$\partial^2 I / \partial \ y^2 = 2\sum_1^n || x - x_i ||^{-p} / \sum_1^n || x - x_i ||^{-p}$$

is positive definite.

These indicate that $y = f(x)$ in E(5-1) is the value of y which minimizes the error function $I(y)$.

The function $I(y)$ depends only upon the Euclidean distance over $Y = R$ and $X = R^d$. In order to generalize the Shepard's interpolation method completely, we propose the error function:

$$I(y) = \Sigma_1^k d_y(y, y_i)^2 d_x(x, x_i)^{-p} \,/\, \Sigma_1^k d_x(x, x_i)^{-p} \,. \qquad \text{E(5-2)}$$

defined in terms of the generalized distance definitions over the problem and solution space. Here, the set $\{x_1, x_2, \ldots x_k\}$ are the k nearest neighbours in the problem space to the point x. $d_x(x, x_i)$ and $d_y(y, y_i)$ are distance (or dissimilarity coefficients) on domains $x \in X, y \in Y$. Then by the discussion above over Shepherd's interpolation, it is natural to propose that the interpolant value y is defined as the value $y \in Y$ which minimizes the error function I. We term this interpolation approach as the Generalised Shepard Nearest Neighbour (GSNN) method.

This method is different from the Distance-Weighted Nearest Neighbour method. Although both of them are local forms of Shepard's method in continuous solution domains, DWNN is not such a local form in discrete domains. In these domains DWNN relies on a voting function (i.e., $\delta(y, y') = 1$ if $y = y'$ and where $\delta(y, y') = 0$ otherwise). However, by introducing a distance metric, $d_y(y, y_i)$ defined on the solution domain Y, GSNN searches for the interpolation value $y \in Y$ that minimises the error function I.

The Generalised Shepard Nearest Neighbour Algorithm (GSNN) is given as follows:

Generalised Shepard Nearest Neighbour Algorithm
$\hat{f}(x_q) \leftarrow \arg\min_{y \in Y} \sum_{i=1}^{k} w_i d_y^{\,2}(y, f(x_i))$. E(5-3)   where $w_i \equiv \dfrac{1}{d_x(x_q, x_i)^p}$

Although E(5-3) is somewhat similar in appearance to the formula E(3-3) of DWNN (see Section 3.5.2.2), we should notice that the set Y in this algorithm is the set of all possible y-values, whereas in the formula for DWNN, Y is the set of y-values in the retrieved set.

5.4 GENERAL PROPERTIES OF GSNN

We can get an idea of how the method works by first examining a simple case $y = f(x)$, as illustrated in Fig. 5-1. Here $X = R$, $Y = R$ and $d_x(x, x_i)$, $d_y(y, y_i)$ are absolute distances in R. We take $p = 1$, $k = 2$ and consider two retrieved cases x_1, x_2 in the neighbourhood of point x.

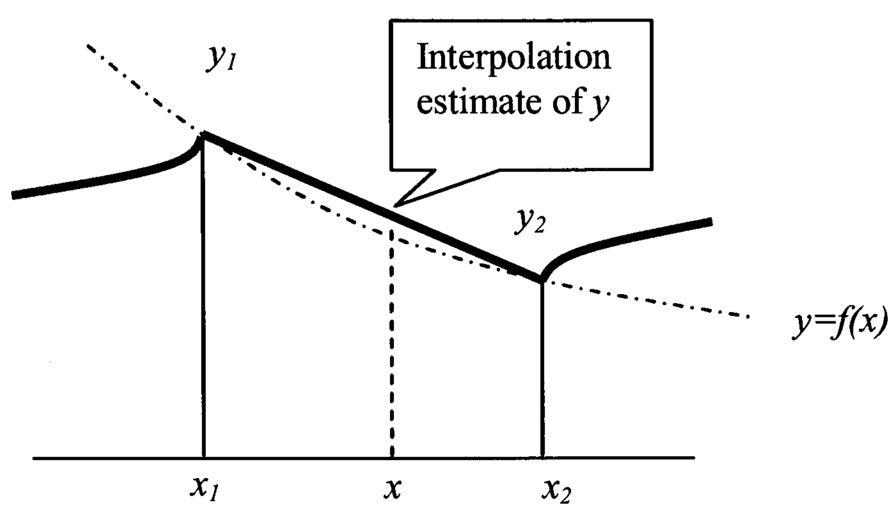


Figure 5-2. Interpolation using the function $I(y)$ for $y = f(x)$

For $x_1 \leq x \leq x_2$, we have:

$$I(y) = [(x_2 - x)(y - y_1)^2 + (x - x_1)(y - y_2)^2]/(x_2 - x_1).$$

The minimum value of I occurs when (x,y) lies on the straight line:

$$(x_2 - x_1)y = (y_2 - y_1)x + (x_2 y_1 - x_1 y_2)$$

Therefore, for this simple case the interpolation curve of GSNN is a straight line between the points (x_1, y_1) and (x_2, y_2). For smooth curves such as the one illustrated in Fig. 5-2, we expect good estimation by interpolation between the two retrieved cases.

In a more general case, where $X = R^d$ and $Y = R^1$, $1 \leq k$, we can also expect good estimation for x interpolated within a neighbourhood of the k retrieved cases, and rather worse for extrapolated values of x. In the numerical field, Shepard's method takes a weighted average of the retrieved y values, where weights are determined by the inverse distance from the cases. This gives good results near and within the retrieved set. However, it can never give a value outside the range of retrieved y values. For this reason alone, it is unlikely to be useful for extrapolation. The implications of this for CBR are that the method is likely to require boundary cases to be included in the case base, since it will not necessarily be capable of extrapolating them.

To compare the method with k-NN, we should first note that if $k = 1$, then GSNN actually reduces to the nearest neighbour method. If $k=2$, then for smooth functions it should be able to achieve the same performance as for $k = 1$, but with a smaller case base. We expect that the interpolation will do worst in bumpy chaotic domains, for instance where nominal values change rapidly as x changes. However, in these domains the k-NN will also do badly, and require a dense set of cases to cover the variation in y. For these dense case bases the interpolative method (GSNN or k-NN) with $k=2$ should give equivalent performance. This intuitive argument is confirmed by the tests in section 5.6.

The optimum value of k for a given problem is related to both the dimensionality, n, of the problem space and to the sparseness of the case base. The interpolation surface has dimension n in the $n+1$ dimensional (x,y) space. Ideally we would like $n+1$ data points to fix this surface. However, if we retrieve $n+1$ points from a sparse data set, some of them may be very remote from our target set, and may not be relevant to the local problem. The inverse distance weighting will compensate for the remoteness of retrieved cases to some extent, but how effective this is must be determined by experimentation.

The effect of the parameter p on the function $I(y)$ is generally to decrease the influence of more remote cases. The tests in Sections 5.6.1 and 5.6.2 will show that increasing p from 1 to 2 does not greatly influence the quality of estimation.

5.5 ILLUSTRATIVE EXAMPLE: INTERPOLATION OVER UNORDERED NOMINAL VALUES

In this example we illustrate how GSNN works in detail. We choose the well known irises dataset [Fisher, 1936]. Although the iris data set contains only continuous variables in the problem domain, the solution space is a set of 3 nominal values, which is sufficient to illustrate how the method works.

The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. We take the problem space X to be R^4, so that $x = (x_1, x_2, x_3, x_4)$ is a point in problem space, where x_1 = sepal width, x_2 = sepal length, x_3 = petal width, x_4 = petal length. The solution space $Y = \{setosa, versicolour, virginica\}$. For this problem, we need to define distance metric in both problem space X and solution space Y. For the problem space we define distance according to a weighted sum of attributes. For convenience, we assign equal weight = ¼ for each attribute, so that:

$$d_x(x, x') = \tfrac{1}{4}\,(|x_1 - x_1'| + |x_2 - x_2'| + |x_3 - x_3'| + |x_4 - x_4'|)$$

For the Y space, we need to construct $d_y(y,y')$. This is the equivalent to Chatterjee and Campbell's second primary task, discussed in Section 5.1. In this test, we have used the distances between cluster centres (see Fig. 5-3) to represent the distance between the classes. These distances are shown in the Table 5-1:

Table 5-1. Distance between Iris classes

	setosa	*versicolour*	*virginica*
setosa	*0*	*.35*	*.49*
versicolour	*.35*	*0*	*.18*
virginica	*.49*	*.18*	*0*

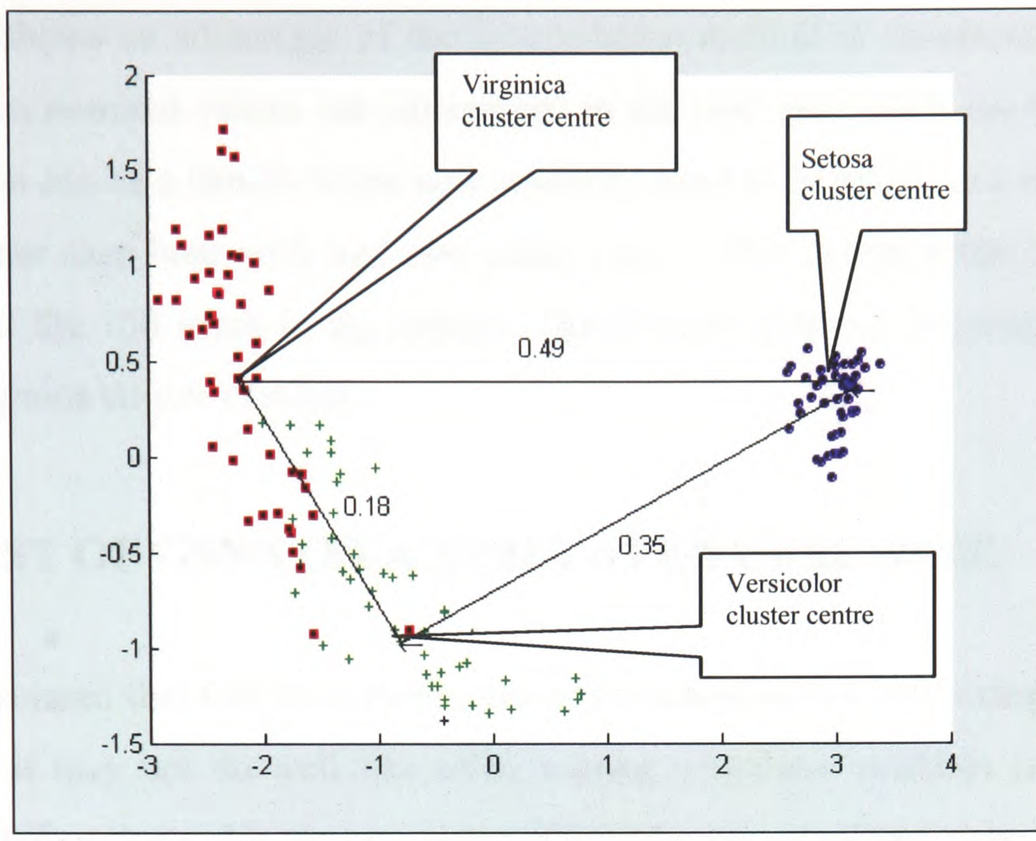


Figure 5-3. Principal component plot of the Iris dataset

To demonstrate how the method works, we take two cases, one from *setosa* and one from *virginica*:

$x_1 = (4.4, 2.9, 1.4, 0.2)$, $y_1 =$ *setosa*

$x_2 = (7.2, 3.2, 6, 1.8)$, $y_2 =$ *virginica*

We take as target the *versicolour* iris:

$x = (5.5, 2.3, 4, 1.3)$, $y = ?$

Taking p=1 and k = 2, the function $I(y)$ is:

$$I(y) = \Sigma_1^2 d_y(y, y_i)^2 d_x(x, x_i)^{-1} / \Sigma_1^2 d_x(x, x_i)^{-1}$$

$$= ((0.36)^{-1} d_y(y, setosa)^2 + (0.35)^{-1} d_y(y, virginica)^2) / ((0.36)^{-1} + (0.35)^{-1})$$

Using the values for d_y in Table 5-1, we have the following values for $I(y)$;

I(setosa) = 0.1217

I(versicolour) = 0.0768

I((virginica) = 0.1184

Since *I(versicolour)* is minimum, we take y = *versicolour* as the estimated value.

This example shows an advantage of the interpolation method in situations in which it can correctly predict nominal values not represented in the case base itself. As discussed later in Section 5.8, this can be a benefit when new solutions need to be added to a case base. In fact, from an extreme case base with just two cases used in this example the method correctly predicts 128 of the 150 irises in the dataset. The DWNN method can only predict the 100 setosa and virginica targets correctly.

5.6 TEST OF GSNN ON A SIMULATED CASE BASE

We have conjectured that GSNN performs better for interpolation than extrapolation and due to this nature it may not do well like other nearest neighbour methods in bumpy chaotic domains. To demonstrate this characteristic of GSNN, we simulated case bases of varying density and structure and used the method to estimate simulated target sets. As a basis for the simulation, we adapted the smoothly varying function:

$$F_3(x_1, x_2) = \sin\ 2\pi x_1 * \sin\ \pi x_2\ . \qquad \text{E(5-4)}$$

used by Ramos and Enright (2001) to test out Shepard's method for interpolation over scattered data. This is a good test function to take because it allows us to compare results with those given by Ramos and Enright. We adapted E(5-4) by discretising the function to give 21 nominal values, $y_1, \ldots, y_{21}$. These are the 21 integral values of the function:

$$y = Int\ (10\ sin\ 2\pi x_1 * sin\ 2\pi x_2\ ),$$

where *Int()* is the integer function that always return an integer value.

Although these values $y_1, \ldots y_{21}$ are in fact numeric, we have treated them as nominal throughout this experiment, and inherited a distance metric from the numeric values:

$$d_y(y_i, y_j) = |y_i - y_j|$$

In this way, we have treated the 21 values $y_1, \ldots, y_{21}$ as symbols, with no intrinsic order but with an externally imposed metric $d_y(y_i, y_j)$. Ramos and Enright tested Shepard's method on E(5-4) with both regularly spaced node sets and randomly spaced, and we have followed this example in two tests. Test I in Section 5.6.1 uses regularly spaced cases at various case densities. This might represent a well organized case base, where these cases had been selected from a large available pool. Test II in Section 5.6.2 uses randomly selected cases, and is intended to represent disorganized sparse case bases. Cases (x_1, x_2, y) are constructed as: $(x_1, x_2, y = Int(10\sin 2\pi x_1 * \sin 2\pi x_2)), \quad 0 \le x_1, x_2 \le 1$

5.6.1 Test I: Regularly Spaced Node Sets

In Test I, cases were constructed over a regular square lattice, with 7^2, 10^2, 20^2, 25^2, 30^2 points. Fig. 5-4, 5-5, 5-6 and 5-7 show the result of Test I with $k = 1, 2, 3$ and 4.

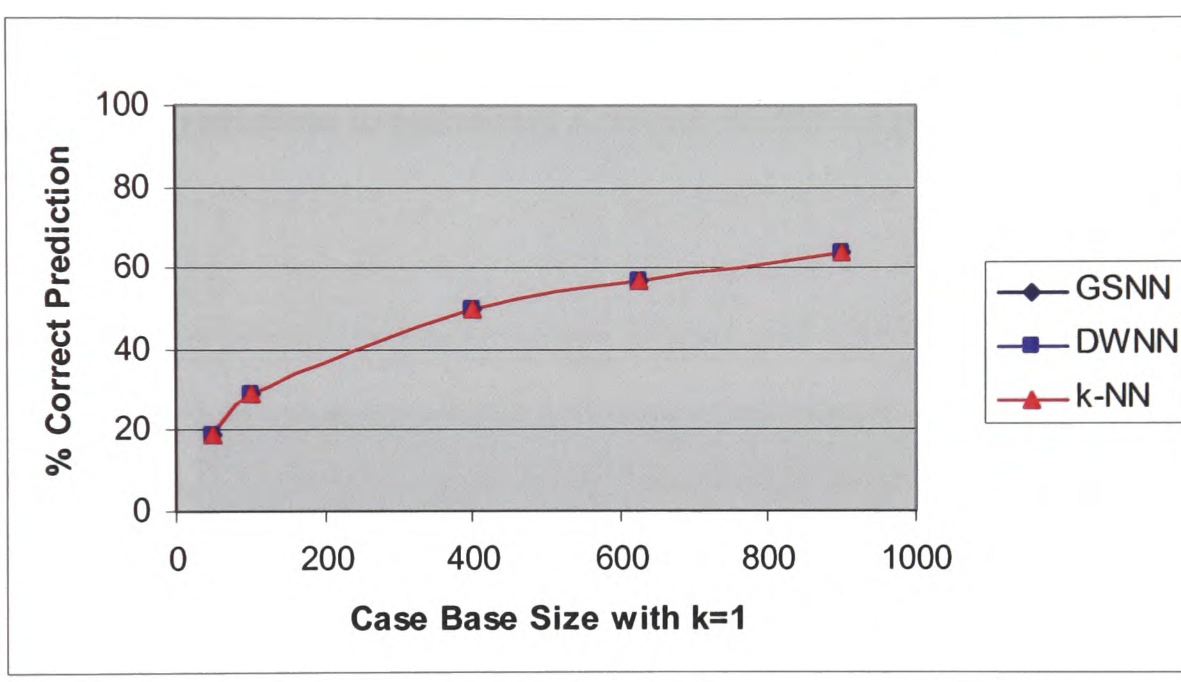


Figure 5-4. Correct Predictions in estimating a test set of 1000 targets, for regular case bases with k=1

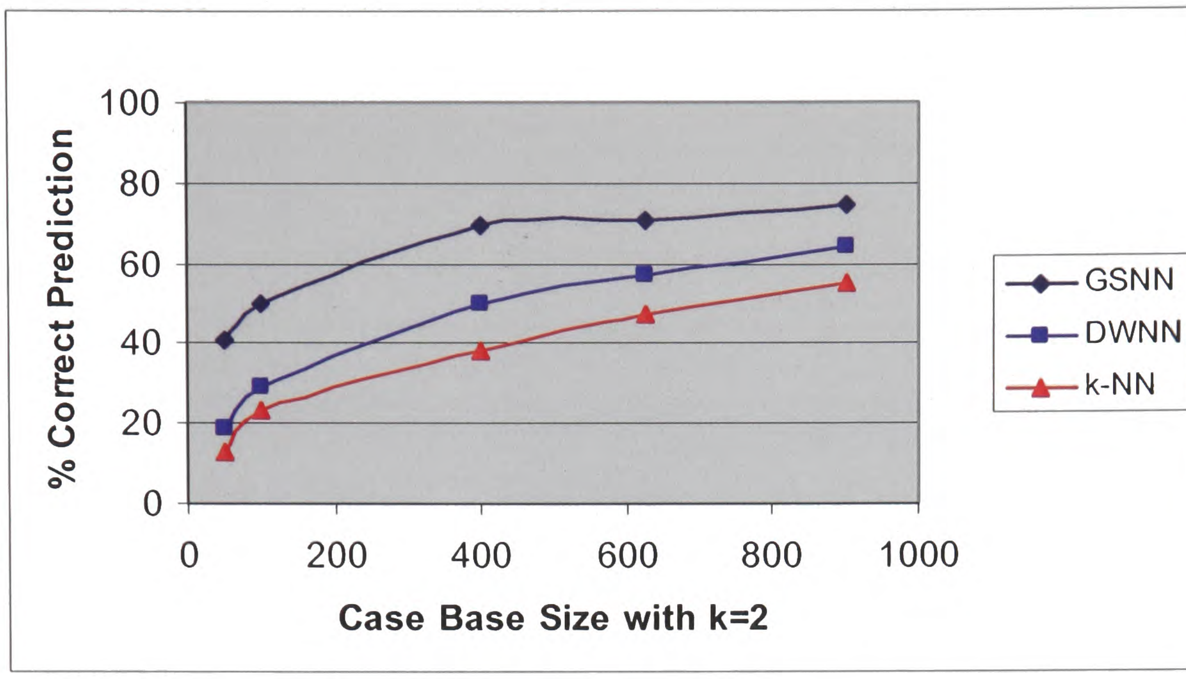


Figure 5-5. Correct Predictions in estimating a test set of 1000 targets, for regular case bases with k=2

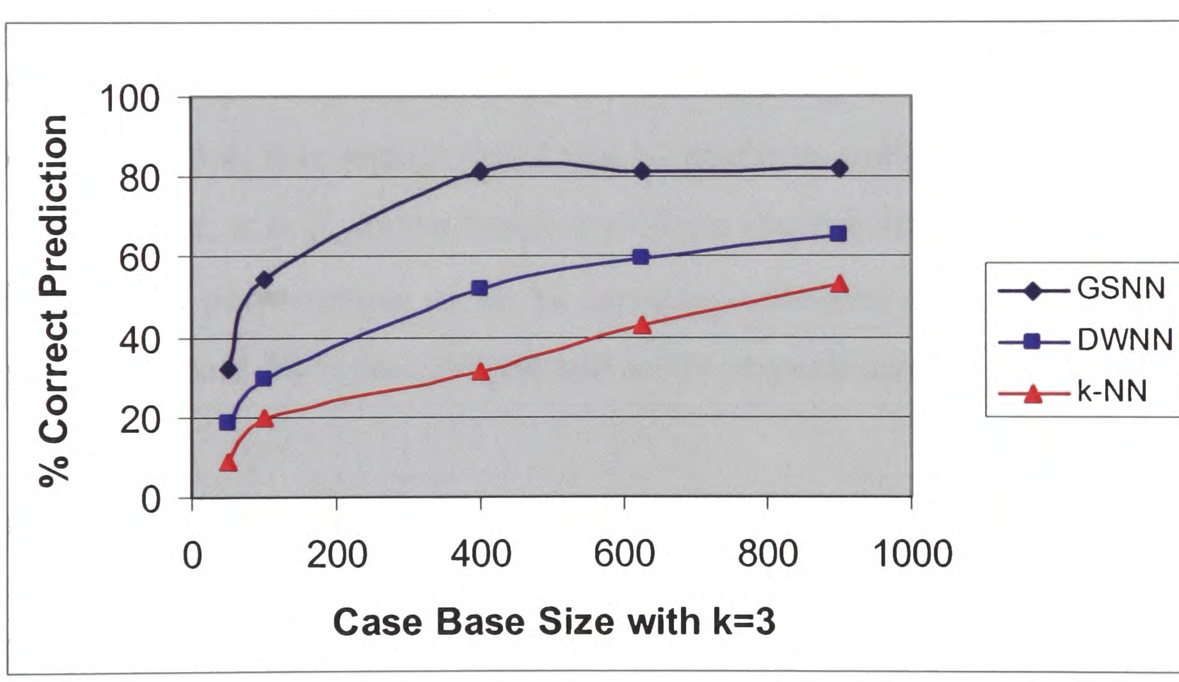


Figure 5-6. Correct Predictions in estimating a test set of 1000 targets, for regular case bases with k=3

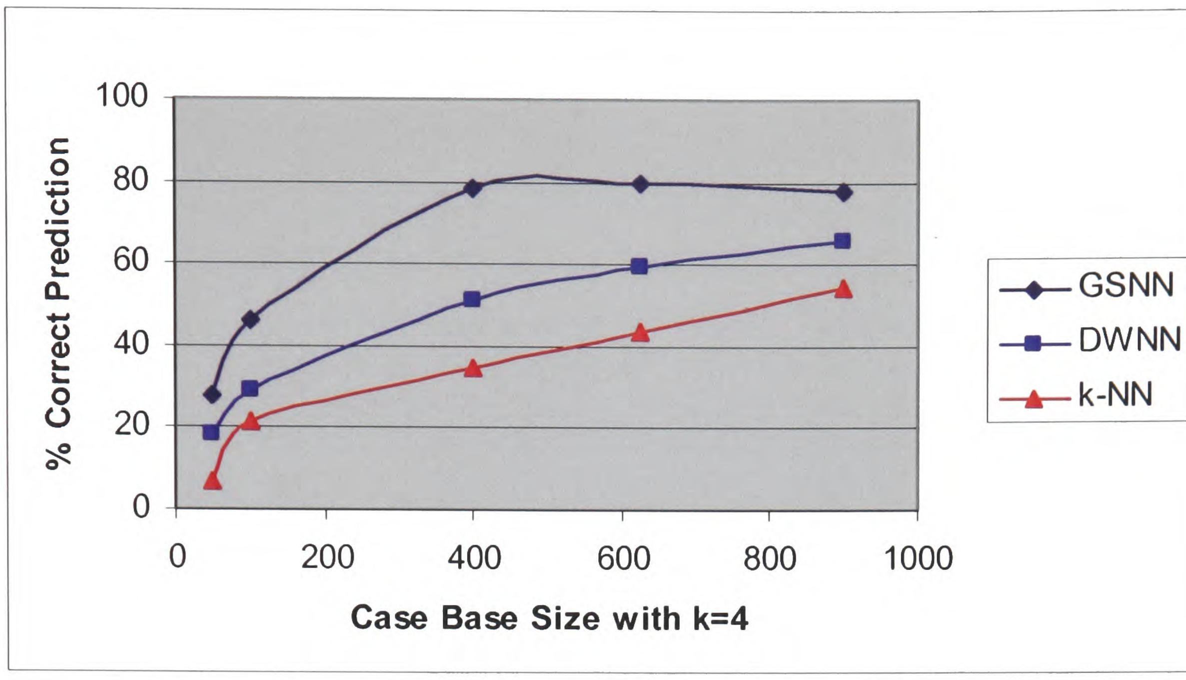


Figure 5-7. Correct Predictions in estimating a test set of 1000 targets, for regular case bases with k=4

These results confirm that GSNN can out-perform both k-NN and DWNN for case bases with regular structure. In fact, the optimum value of k is 3 (see Fig. 5-6) for GSNN. In the general discussion of Section 5.4, it is argued that $k = n+1$ might be optimal for an $n+1$ dimensional (x,y) space. In this test, $n = 2$, so the result $k=3$ does confirm the argument. We see that in fact GSNN reached a performance of 80 % correct predictions for a case base of only 400 cases, as against 50% and 30 % for DWNN and k-NN respectively for the same case base.

5.6.2 Test II: Randomly Spaced Node Sets

In Test II, we took case bases of the same size as those in Test I, but this time generated x_1, x_2 randomly within the unit square $0 \le x_1, x_2 \le 1$. Fig. 5-8, 5-9, 5-10 and 5-11 show the results of this test with $k = 1, 2, 3$ and 4.

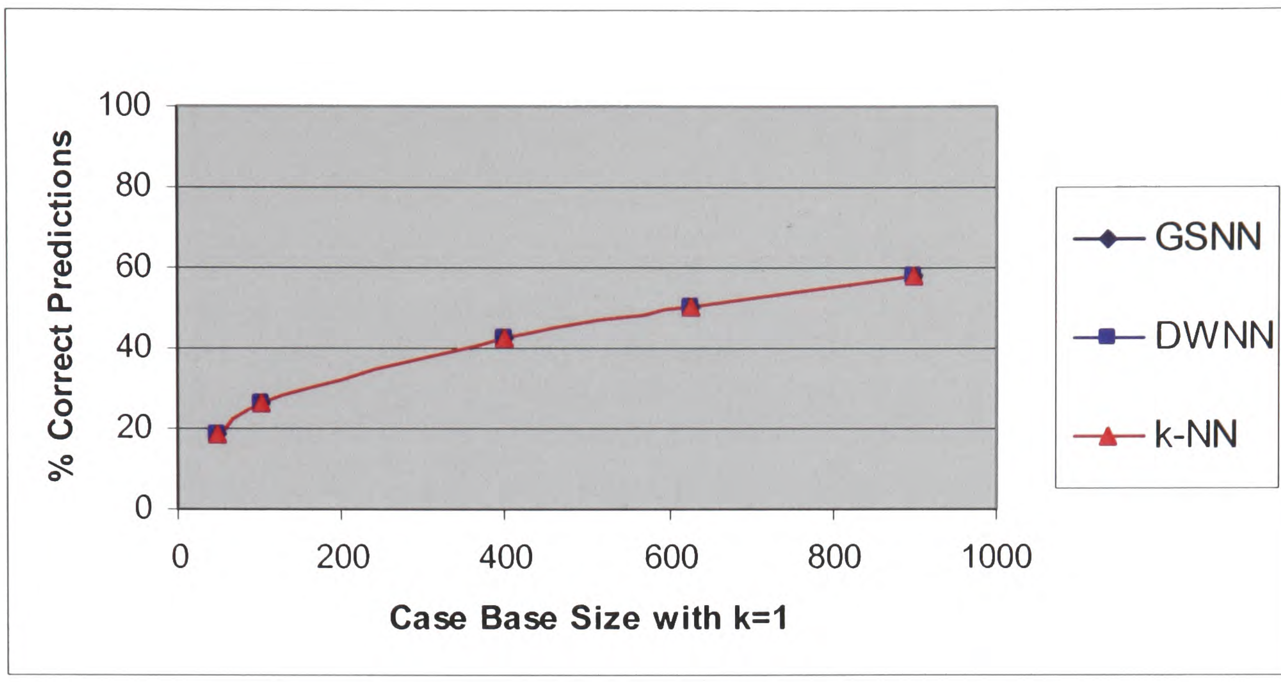


Figure 5-8. Correct Predictions in estimating a test set of 1000 targets, for random case bases with k=1

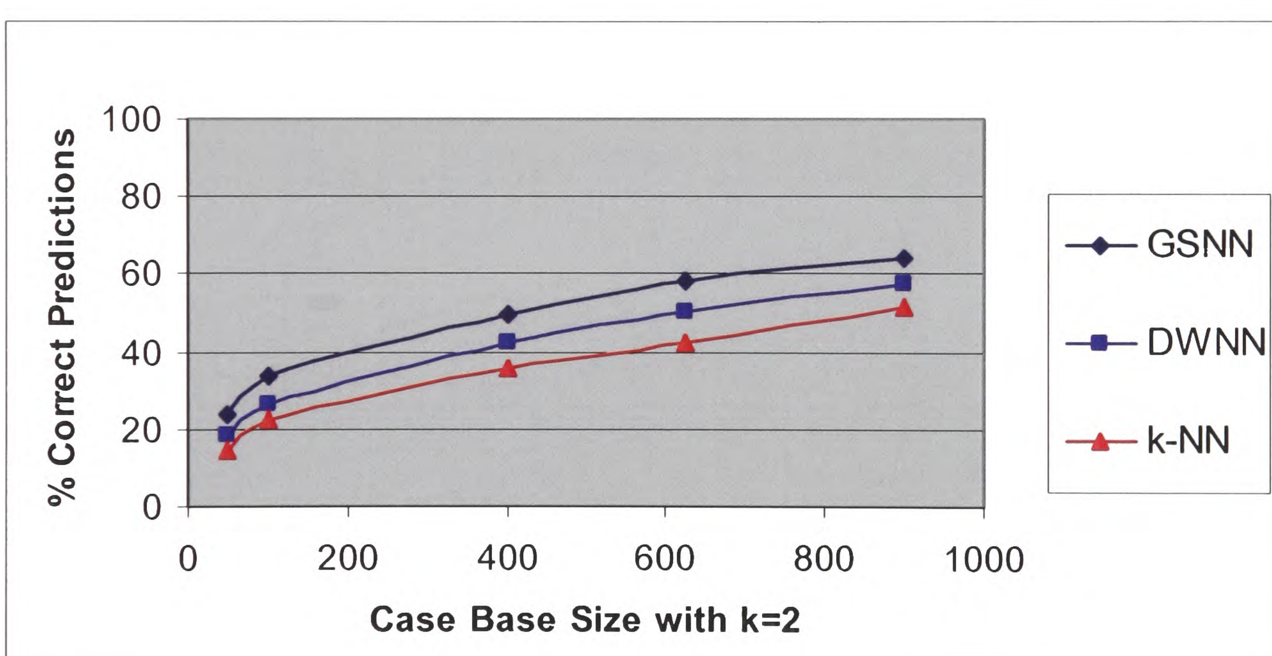


Figure 5-9. Correct Predictions in estimating a test set of 1000 targets, for random case bases with k=2

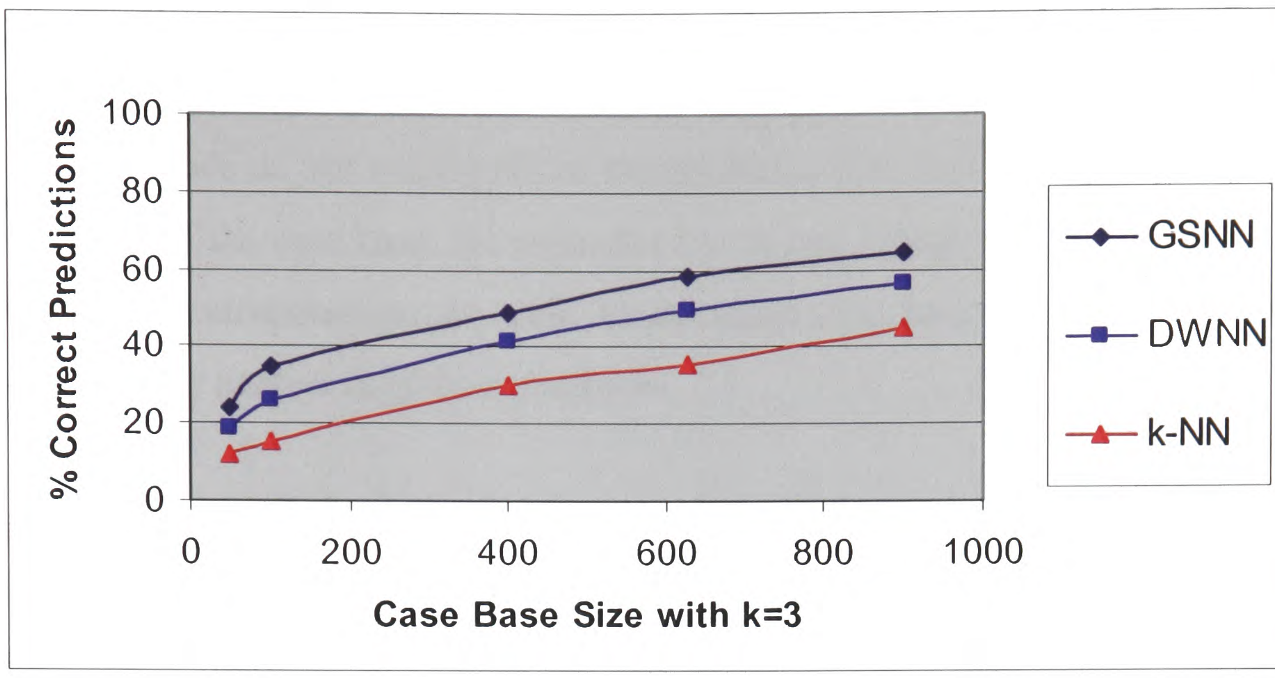


Figure 5-10. Correct Predictions in estimating a test set of 1000 targets, for random case bases with k=3

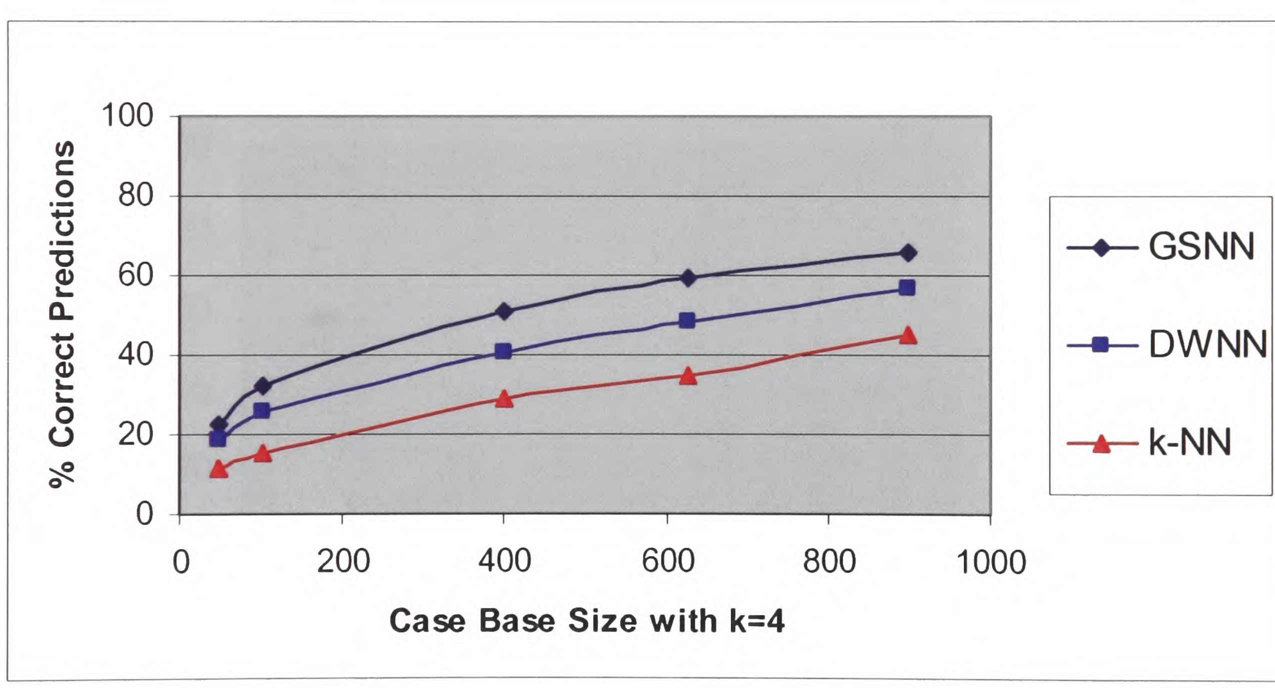


Figure 5-11 Correct Predictions in estimating a test set of 1000 targets, for random case bases with k=4

In Section 5.4, we mentioned that the interpolation will do worst in bumpy chaotic domains, and in these domains the nearest neighbour method requires a dense set of cases to cover the variation in *y*. Fig. 5-9 show that for these dense case bases the interpolative method with *k=2* give at least equivalent performance as *k=1*. The results show that more errors are recorded for random case bases than for regular case bases of equivalent size, whatever the value of *k*, although once again, *k=3* is optimal for GSNN. However, the improvement is not so marked as in the regular case base test in Section 5.6.1. The reason for this is that interpolation methods do not work well for extrapolation (see Section 5.4). In Test I, because of the regularity of the case base, all estimates are in fact interpolations, whereas in Test II, we expect some extrapolations as well. Once again, the results show that GSNN out-performed the other nearest neighbour methods.

As the case base size increases, the number of errors diminishes no matter what the value of *k* takes. For dense case bases, there is no particular advantage in choosing $k > 1$, in view of the extra computation involved in finding minima of the function $I(y)$. Finally, the same tests were repeated for $p = 2$. In all trials the number of errors for $p = 2$ are nearly the same as $p = 1$. Fig. 5-12 and 5-13 show the correct predictions of GSNN with $p=1$ and $p=2$ in estimating a test set of 1000 targets for both regular and random case bases. In these tests we use $k=3$.

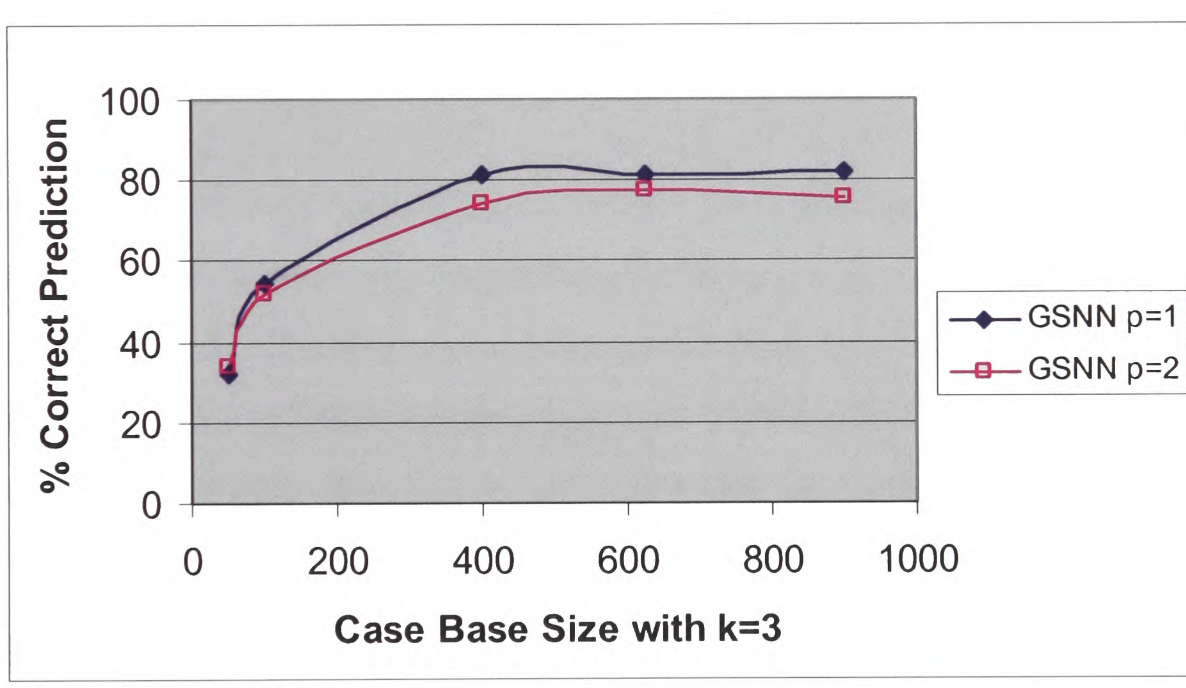


Figure 5-12 Correct Predictions in estimating a test set of 1000 targets, for regular case bases with k=3, p=1, 2

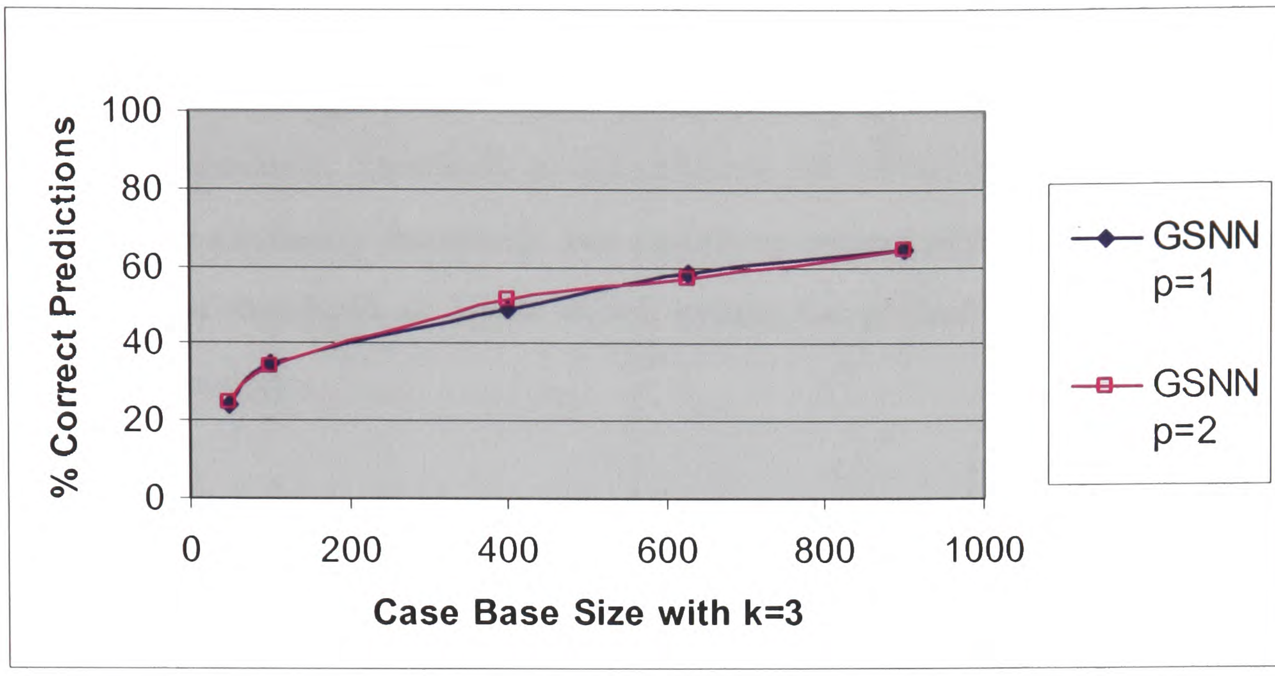


Figure 5-13 Correct Predictions in estimating a test set of 1000 targets, for random case bases with k=4, p=1, 2

5.7 TEST OF GSNN ON THE TRAVEL CASE BASE

We have pointed out that the capability of GSNN for predicting nominal values not in the retrieval set can be advantageous in improving performance. To examine how this applies in practice, in this section, GSNN is tested on a benchmark case base from the travel domain [Lenz et al., 1996]. The problem investigated here is that of predicting a hotel for a given package holiday. We divide the domain attributes into the problem domain: X = {holiday type, price, number of persons, duration, season, destination region, accommodation type, transportation}, and the solution domain Y = {Hotel}. The case base consists of 1024 package holidays. For the problem space we define distance according to a weighted sum of attributes with equal weight. For the Y space, we derive a metric on Y defined by its region and class of accommodation.

From the original case base we have chosen 1000 cases for experiments. 300 cases among the 1000 cases are chosen randomly as target problems. These cases are unseen target problems, not in the case base. The remaining 700 cases are used to form experimental case bases. We divide 700 cases into 7 independent case bases ranging with size from 100, 200, 300 to 700. This enables us to examine the predictive power of each retrieval method using various case base sizes on the 300 unseen target problems. Following Smyth & McKenna (1998), we use a similarity threshold as the criterion for correct prediction. If the predicted value is within the similarity threshold, that counts as correct prediction. In the experiments below, we take the threshold as 100% which means the predicted value has zero distance from the expected value.

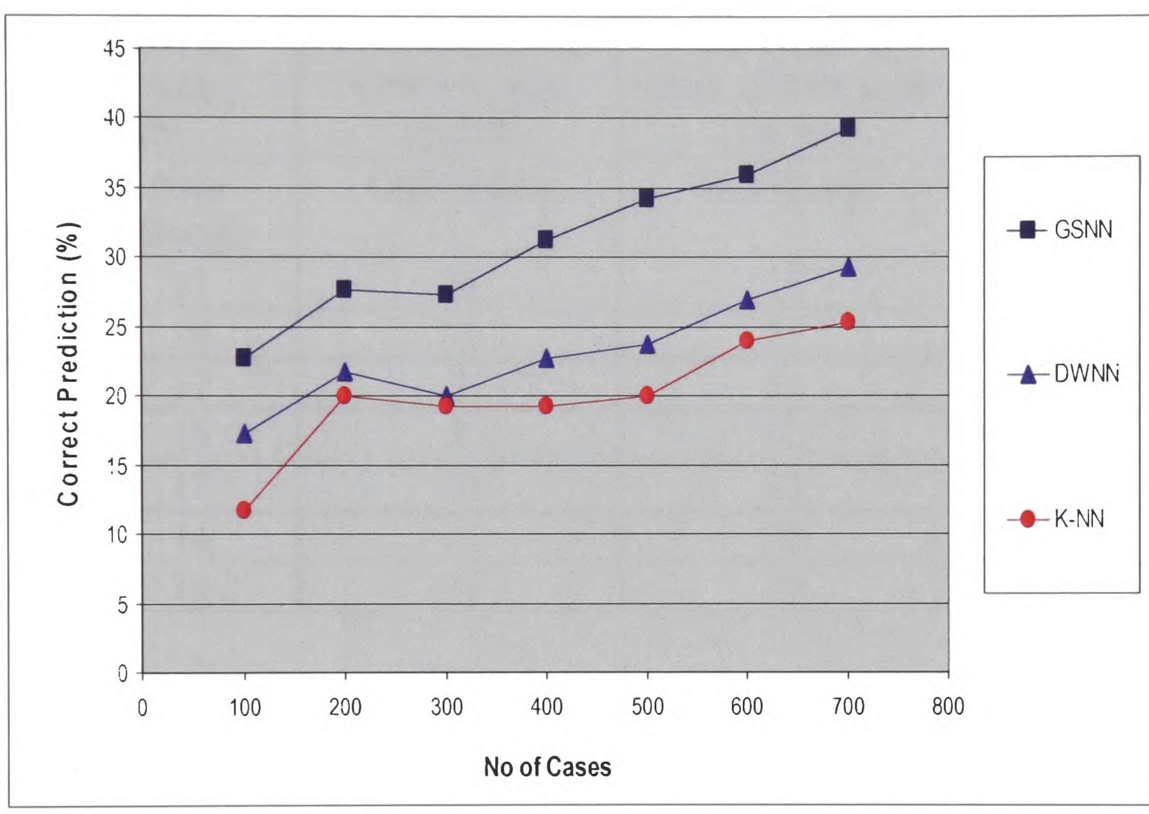


Figure 5-14 Comparing the correct prediction accuracy (%) of retrieval methods on 300 unseen target problems

Fig. 5-14 shows the comparison of correct prediction accuracy (%) of GSNN, DWNN and *k*-NN on the 300 unseen target problems. The experiment shows clearly that GSNN out-performs both DWNN and *k*-NN, due chiefly to its ability as shown in the Irises example in Section 5.5 to correctly predict nominal values not in the retrieval set, or even in the case base. This will be illustrated later in Sections 5.7.1 and 5.7.2.

Analysis of the results shows that GSNN has two advantages. The first is that it can predict solutions that are not in the case base. In this example, GSNN managed to predict correctly hotels for which there were no holiday packages in the case base. In column 3 (i.e., *New*

Hotels), Table 5-2, we see the statistics on new hotel predictions produced by GSNN. In contrast, DWNN cannot predict any of these correctly. Notice that these decrease as the case base size increases. This is because the number of new hotels in the test set decreases as the case base size increases.

Table 5-2. Comparing the number of correct predictions of old hotels and new hotels for GSNN and DWNN, on 300 unseen target problems

No of Cases	No of Correct Predictions by GSNN, not DWNN		No of Correct Predictions by DWNN, not GSNN	No of Correct Predictions by both GSNN and DWNN	GSNN	DWNN
	Old Hotels	New Hotels	Old Hotels	Old Hotels	Total	Total
100	7	23	14	38	68	52
200	14	14	10	55	83	65
300	12	17	7	53	82	60
400	19	15	8	60	94	68
500	25	17	10	61	103	71
600	24	14	11	70	108	81
700	26	14	10	78	118	88

The second advantage is that GSNN can correctly predict "old" hotels which are not in the retrieval set. Once again DWNN cannot predict these. Table 5-2, Column 2 (i.e., *Old Hotels*) shows the number of "old" hotels correctly predicted by GSNN which were incorrectly predicted by DWNN. Contrast this against column 4 shows the number of "old" hotels that are correctly predicted by DWNN but incorrectly by GSNN. Column 5 shows the number of "old" hotels that are correctly predicted by both GSNN and DWNN. Table 5-3 gives similar analysis on the correct prediction of GSNN and *k*-NN. The examples in Section 5.7.1 and 5.7.2 illustrate the advantages of GSNN over DWNN and *k*-NN.

Table 5-3. Comparing the number of correct predictions of old hotels and new hotels for GSNN and KNN, on 300 unseen target problems

No of Cases	No of Correct Predictions by GSNN, not KNN		No of Correct Predictions by KNN, not GSNN	No of Correct Predictions by both GSNN and KNN	GSNN	*k*-NN
	Old Hotels	New Hotels	Old Hotels	Old Hotels	Total	Total
100	19	29	15	20	68	35
200	25	25	27	33	83	60
300	25	23	24	34	82	58
400	32	22	18	40	94	58
500	39	23	19	41	103	60
600	41	19	24	48	108	72
700	50	18	26	50	118	76

5.7.1 Illustrative Problem I: Solution Not in the Retrieval Set

Given case definition:

(JourneyCode, HolidayType, Price, NumberOfPersons, Duration, Season, Region, Accommodation, Transportation)

We take as target the "Acces Hotel Mercure, Belgium." hotel:

x = (23, Recreation, 978, 3, 7, July, Belgium, Three stars, Car)

For 3 nearest neighbours, we get:

x_1 = (133, Bathing, 1047, 3, 7, July, BalticSea, Three stars, Car),
Hotel: "Hotel Baltic, Usedom.", distance = 0.2512.

x_2 = (781, Recreation, 849, 3, 7, February, Belgium, Holiday Flat, Car),
Hotel: "SunParks Groendyk, Belgium.", distance = 0.2523.

x_3 = (895, Active, 998, 3, 7, December, Bavaria, Three stars, Car),
Hotel: "Sporthotel Rosenberger, Bavaria.", distance =0.3753.

Prediction from GSNN gives: "Acces Hotel Mercure, Belgium."

Prediction from DWNN gives: "Hotel Baltic, Usedom."

Prediction from KNN gives: "Hotel Baltic, Usedom."

This example shows that GSNN can correctly predict the hotel for package holiday x (i.e., JourneyCode = 23). It is noted that the solution given by GSNN "Acces Hotel Mercure, Belgium." is an old hotel and that it is distinct from the solutions, which appear in the retrieval set (i.e., "Hotel Baltic, Usedom.", "SunParks Groendyk, Belgium.", "Sporthotel Rosenberger, Bavaria.").

5.7.2 Illustrative Problem II: Solution Not in the Case Base

We take as target the "Hotel Zur Post, Upper Bavaria." hotel:

x = (71, Recreation, 3135, 3, 14, August, Alps, FourStars, Car)

For 3 nearest neighbours, we get, we get:

x_1 = (462, Recreation, 2322, 3, 14, July, SalzbergerLand, FourStars, Car),
Hotel: "Hotel Eschbacher, Salzb. Land.", distance = 0.2642.

x_2 = (586, Recreation, 1958, 6, 14, August, Alps, HolidayFlat, Car),
Hotel: "H.Flat Switzerland.", distance = 0.2705.

x_3 = (495, Wandering,, 2322, 3, 14, August, Tyrol, TwoStars,, Car),
Hotel: "Pension Tannenhof, Tyrol.", distance =0.3892.

Prediction from GSNN gives: "Hotel Zur Post, Upper Bavaria."
Prediction from DWNN gives: "Hotel Eschbacher, Salzb. Land."
Prediction from KNN gives: "H.Flat Switzerland."

This example shows that GSNN can correctly predict the hotel for package holiday x (i.e., JourneyCode = 71). It is noted that the solution given by GSNN "Hotel Zur Post, Upper Bavaria." is distinct from the solutions, which appear in the retrieval set (i.e., "Hotel Eschbacher, Salzb. Land", "H.Flat Switzerland", "Pension Tannenhof, Tyrol."). In addition, "Hotel Zur Post, Upper Bavaria." is a new hotel, which is not yet present in the case base.

5.8 CONCLUDING REMARKS

In this chapter, we have proposed a method for interpolation over nominal values. The method assumes only that a distance or dissimilarity metric is defined over the problem domain and over the solution domain. The method has an advantage for CBR in that it is applicable to case bases with nominal values in the problem and solution domain where no natural ordering exists. The method generalizes Shepard's interpolation method by expressing it in terms of the minimization of an error function I(y). This function relies only on distance metrics defined over problem and solution spaces.

The method has two advantages over conventional nearest neighbour methods: the first is that it can predict "new" instances, not yet present in the case base; the second is that it can predict solutions not present in the retrieval set. In fact in nominal domain GSNN is the true generalization of Distance Weighted Nearest Neighbour (DWNN) except that it has been reformulated as the minimization of an error function defined in terms of distance metrics in the solution and problem spaces. In a real domain, GSNN is equivalent to DWNN which is a localized version of Shepard's method. A novel aspect of GSNN is that it provides a general method for interpolation over nominal solution domains.

We have tested the method on three test problems: the well studied Irises problem; a benchmark simulated nominal case base; and a benchmark CBR case base from the travel domain. The examples studied indicate that GSNN can be more efficient as an adaptation engine than conventional nearest neighbour methods.

Another area where GSNN could be useful is that of case-based model building, from experimental or numerical modeling exercises [Woon et al., 2003b; 2003c]. Here the problem is both a selection of cases and planning of case production. Investigations using numerical models indicate that GSNN would appear to be a promising approach for the construction of efficient case-based models.

Chapter 6

Interpolation on a Diverse Retrieval Set

6.1 INTRODUCTION

The experimental results presented in Chapter 5 are derived using nearest neighbours retrieval set for interpolation. We have shown that in Chapter 5, GSNN works better in simulated case bases with regularly spaced node sets than with randomly spaced node sets, and that this is connected with the general property that the algorithm performs better at interpolation than extrapolation. We suspect that the GSNN results of the Travel example in Section 5.7 may suffer from the same problem although the method out-performed the predictive performance of DWNN and k-NN. If this is the case, the requirement for a target to be always interpolated by a set of retrieved cases that surround it should improve the predictive performance of GSNN and other interpolation methods.

Analysis of false predictions from GSNN shows that it may not perform well when the members of the retrieved set are close together since it is highly likely that in this scenario it falls into the extrapolation trap (See Section 5.4 in Chapter 5) with the target being outside the interpolation points. This points to the importance of performing interpolation rather than extrapolation with the retrieval set.

In this chapter, we propose the use of a diversity algorithm to generate a diverse retrieval set used for interpolation. This will increase the chances that all targets are interpolated within a set of interpolation points. Smyth & McGinty (2003) and Smyth & McClave (2001) have shown that selecting a more diverse set can improve the efficiency of recommendation systems. Here, we show that use of a diversity algorithm can also improve the efficiency of the interpolation system. In this chapter, we use the Bounded-Greedy diversity technique in [Smyth & McGinty, 2003] to evaluate the performance of GSNN and other nearest neighbour methods. In Section 6.2 we describe the Bounded-Greedy diversity technique, and in Section 6.3 examine the predictive performance of GSNN, DWNN and k-NN on the travel case base using a diverse retrieval set. Section 6.4 and 6.5 present results obtained from tests on the simulated random case bases and regular case bases used in Section 5.6. We conclude in Section 6.6 on the findings and the advantages of using GSNN and the diversity technique together, with a summary and indications of future work.

6.2 BOUNDED-GREEDY DIVERSITY TECHNIQUE

The Bounded-Greedy diversity technique in [Smyth & McGinty, 2003] is one of a variety of well-known diversity techniques available for improving the efficiency of recommendation systems. Diversity of a set of cases is a measure of how far apart they are from one another. Diversity techniques [Shimazu, 2001; Bridge, 2001; McSherry, 2002] play an important role in recommendation systems. Smyth & McClave (2001) argue that often diversity can be as important as similarity particularly in case-based recommender systems. This is due chiefly to the fact that a case appearing to be similar to a target problem does not necessarily mean that it can be successfully adapted for this target. Comparatively, the same reason can cause GSNN to give poor prediction. This is because in nature GSNN is an interpolation method, which would produce good performance with a set of interpolation points that surround a target.

The bounded-greedy technique actually has two phases: (i) it selects the best *bk* (*b* is the bound (in this case it is a constant >1); *k* is the number of cases in a retrieval set) cases most similar to the target query; (ii) it then builds up the retrieval set (*R*) out of the selected cases incrementally. During each step the remaining cases are ordered according to their quality. The highest quality case is added to *R*. The position of a case *i* in the sequence of the remaining cases is proportional to the similarity between case *i* and the current query *q*, and to the diversity of case *i* relative to those cases that have been selected so far, $R = \{r_1, \ldots, r_m\}$. The equations of the quality (see E(6-1)) and diversity (see E(6-2)) of a case are given as follows:

$$Quality\,(q,i,R) = \alpha * Sim\,(q,i) + (1-\alpha) * Div(i,R) \quad . \qquad \text{E(6-1)}$$

$$Div\,(i,R) = 1 \quad if\, R=\{\};$$

$$= \frac{\sum_{j=1..m} (1 - Sim\,(i, r_j))}{m} \quad otherwise. \qquad \text{E(6-2)}$$

where $Sim(q,i) = \frac{\sum_{j=1..n} w_j * sim(q_j, c_j)}{\sum_{j=1..n} w_j}$ is a measure of the similarity between a case i and the target query q; α is a constant that is used for balancing the similarity and diversity.

The procedure of the Bounded Greedy Selection Technique is given as follows:

Bounded Greedy Selection Technique
Parameters: (q, CB, k, b) *CB'* := *bk* cases in *CB* that are most similar to *q* *R* := {} For *j*:= *1* to *k* Sort *CB'* by *Quality(q, i, R)* for each case *i* in *CB'* *R* := *R* + *First(CB')* *CB'* := *CB'* – *First(CB')* EndFor return *R*

Figure 6-1. Pseudo code for Bounded Greedy Selection technique

6.3 THE TRAVEL CASE BASE

In this experiment, we perform the same test as in Section 5.7 except that we use the bounded-greedy diversity technique to generate a diverse set of candidate cases. The diverse set is then used for interpolation, using DWNN and GSNN. The results for various case base sizes are given in Fig. 6-2, which contrasts the two methods using both diverse retrieval sets and nearest neighbour retrieval sets. We set the parameters in the bounded-greedy diversity technique as follows: α = 0.5, b=3 and k=3.

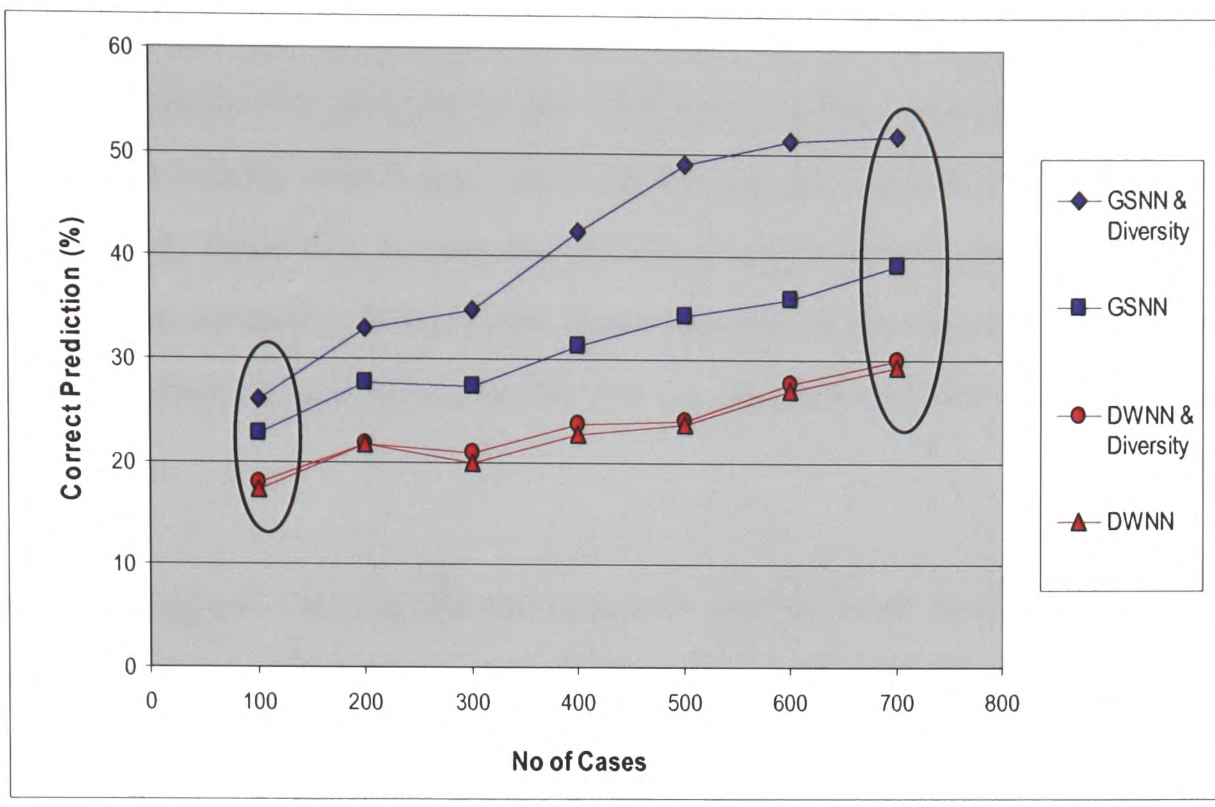


Figure 6-2. Comparing the correct prediction accuracy (%) of GSNN and DWNN using both diverse retrieval sets and nearest neighbour retrieval sets, on 300 unseen target problems

Here we see that GSNN benefits considerably from the use of a diverse retrieval algorithm. DWNN however improves only marginally. This is because DWNN operates using distance weighted voting that is usually dominated by the nearer neighbours. Diverse sets will naturally include more distant neighbours, which will play little part in the voting. GSNN on the other hand does not suffer from this disadvantage.

For a smaller case base (E.g. 300 cases) GSNN shows a better performance on a diverse set than ordinary DWNN. This is due chiefly to the fact that GSNN can predict new hotels not in the case base. For a larger case base (E.g. 700 cases) GSNN also gives good prediction mainly because it can predict old hotels not in the retrieval set. Results show significant improvement in GSNN prediction with the use of a diverse retrieval set. The joint use of GSNN and the diverse retrieval algorithm is nearly twice as good as DWNN prediction in a large case base (E.g. 700 cases). We also note that as the case base size increases, the higher density in the case base does not degrade GSNN prediction. This shows that GSNN is able to give good approximation on a diverse retrieval set.

Once again, the analysis of the results shows that GSNN has two advantages. The first is that it can predict solutions that are not in the case base. In this example, GSNN managed to predict correctly hotels for which there were no holiday packages in the case base. In column 3 (i.e., *New Hotels*), Table 6-1, we see the statistics on new hotel predictions. DWNN cannot predict any of these correctly. Notice that these decrease as the case base size increases. This is because the number of new hotels in the test set decreases as the case base size increases (see Fig. 6-2).

The second advantage is that GSNN can correctly predict "old" hotels, which are not in the retrieval set. Once again DWNN cannot predict these. Table 6-1, Column 2 (i.e., *Old Hotels*) shows the number of "old" hotels correctly predicted by GSNN, which were incorrectly predicted by DWNN. The number of old hotels in the test set increases as the case base size increases (see Fig. 6-2). Contrast this against column 4 shows the number of "old" hotels that are correctly predicted by DWNN but incorrectly by GSNN. Column 5 shows the number of "old" hotels that are correctly predicted by both GSNN and DWNN.

Table 6-1. Comparing the number of correct predictions of old hotels and new hotels for GSNN and DWNN using diverse retrieval sets, on 300 unseen target problems

No of Cases	No of Correct Predictions by GSNN, not DWNN		No of Correct Predictions by DWNN, not GSNN	No of Correct Predictions by both GSNN and DWNN	GSNN	DWNN
	Old Hotels	New Hotels	Old Hotels	Old Hotels	Total	Total
100	5	35	16	38	78	54
200	21	28	15	50	99	65
300	28	22	9	54	104	63
400	44	24	12	59	127	71
500	60	27	12	60	147	72
600	61	22	12	71	154	83
700	63	16	14	76	155	90

In addition, we also performed the same test as above for k-NN and use the same values for α , b and k. The results for various case base sizes are given in Fig. 6-3. Table 6-2 shows that GSNN out-performed k-NN for the smaller case base because it can predict new hotels not in the case base. For a larger case base GSNN also gives good prediction because it can predict old hotels not in the retrieval set. The combination of GSNN and the diverse retrieval set is nearly 1.5 to 2 times as good as all k-NNs prediction in all case bases. Results show that k-NN when interpolating on the diverse set gives better prediction than on the nearest neighbour retrieval set. The reason for this is that the contribution of its nearest neighbours in the retrieval set to the prediction of a new instance is not distance-weighted (see Section 5.1). For votes that are unevenly distributed, the winner is the classification that has the most common vote among the nearest neighbours. It is possible that the result obtained from k-NN on the diverse set is biased to the random selection of classification if votes are equally distributed for each distinct classification in the retrieval set. Also the nearest neighbours in the diverse retrieval set is likely to be different from those in the nearest neighbour retrieval set because they are selected from among the bk cases and are sorted according to its quality and diversity (see Section 6.2) to the new instance. Hence it is possible that cases, which are far apart from the target are treated as equal with other nearer cases to the target. Consequently, they may be selected at random as the classification of the new instance.

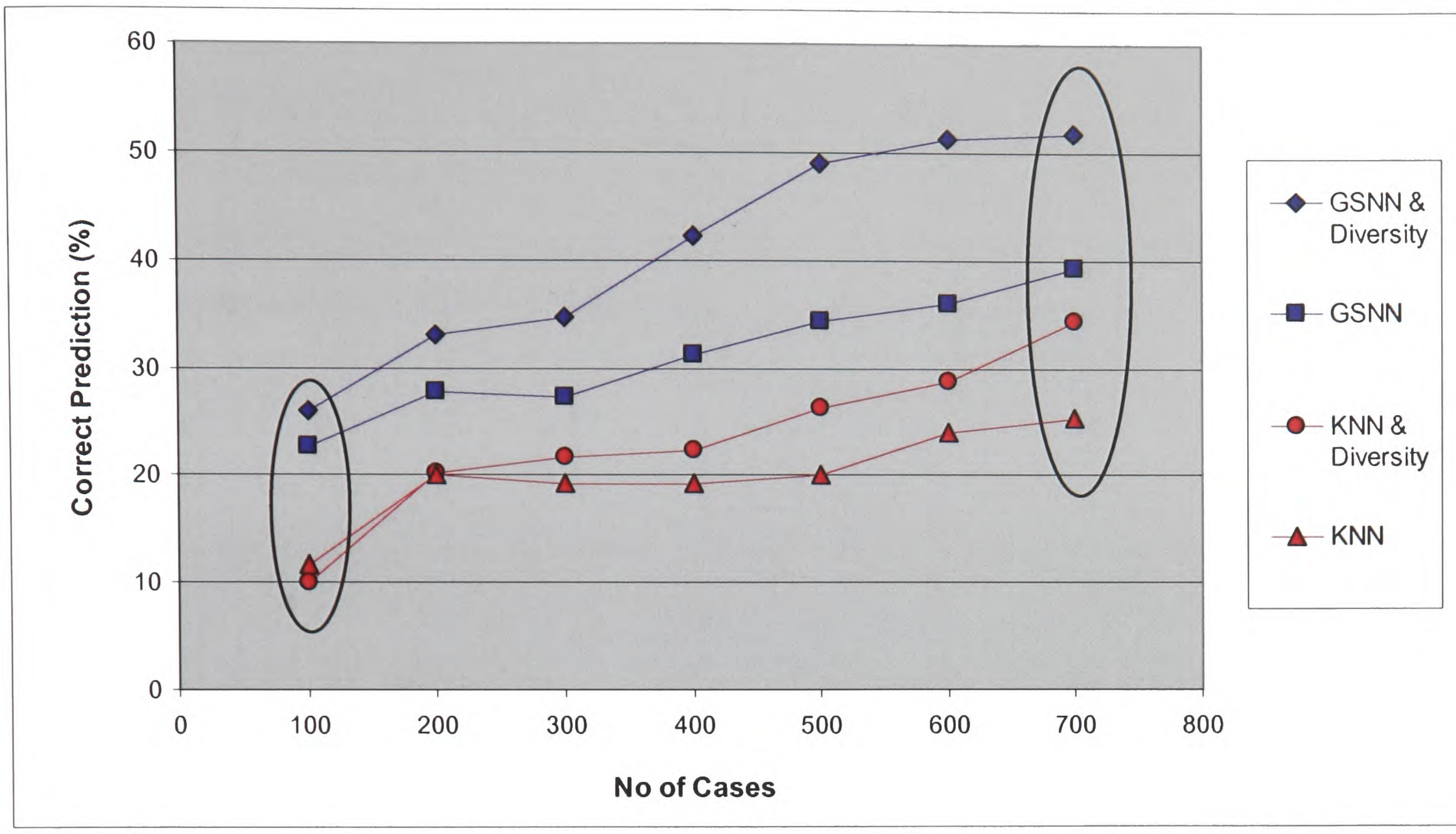


Figure 6-3. Comparing the correct prediction accuracy (%) of GSNN and *k*-NN using both diverse retrieval sets and nearest neighbour retrieval sets, on 300 unseen target problems

Table 6-2. Comparing the number of correct predictions of old hotels and new hotels for GSNN and *k*-NN using diverse retrieval sets, on 300 unseen target problems

No of Cases	No of Correct Predictions by GSNN, not *k*-NN		No of Correct Predictions by *k*-NN, not GSNN	No of Correct Predictions by both GSNN and *k*-NN	GSNN	*k*-NN
	Old Hotels	New Hotels	Old Hotels	Old Hotels	Total	Total
100	20	43	15	15	78	30
200	35	39	36	25	99	61
300	38	38	37	28	104	65
400	47	42	29	38	127	67
500	66	38	36	43	147	79
600	75	32	39	47	154	86
700	69	23	40	63	155	103

On the other hand, it is also noted in Fig. 6-4 that *k*-NN with the diversity technique in some cases (e.g., case base size = 500 or 700) performs better than DWNN. This is due to the fact that DWNN's prediction is distance-weighted. Consequently, its prediction is dominated by the nearest neighbour in the retrieval set (see Section 5.1). Also, because of this DWNN is regarded as more robust to noisy training data (see Section 3.5.2.2.).

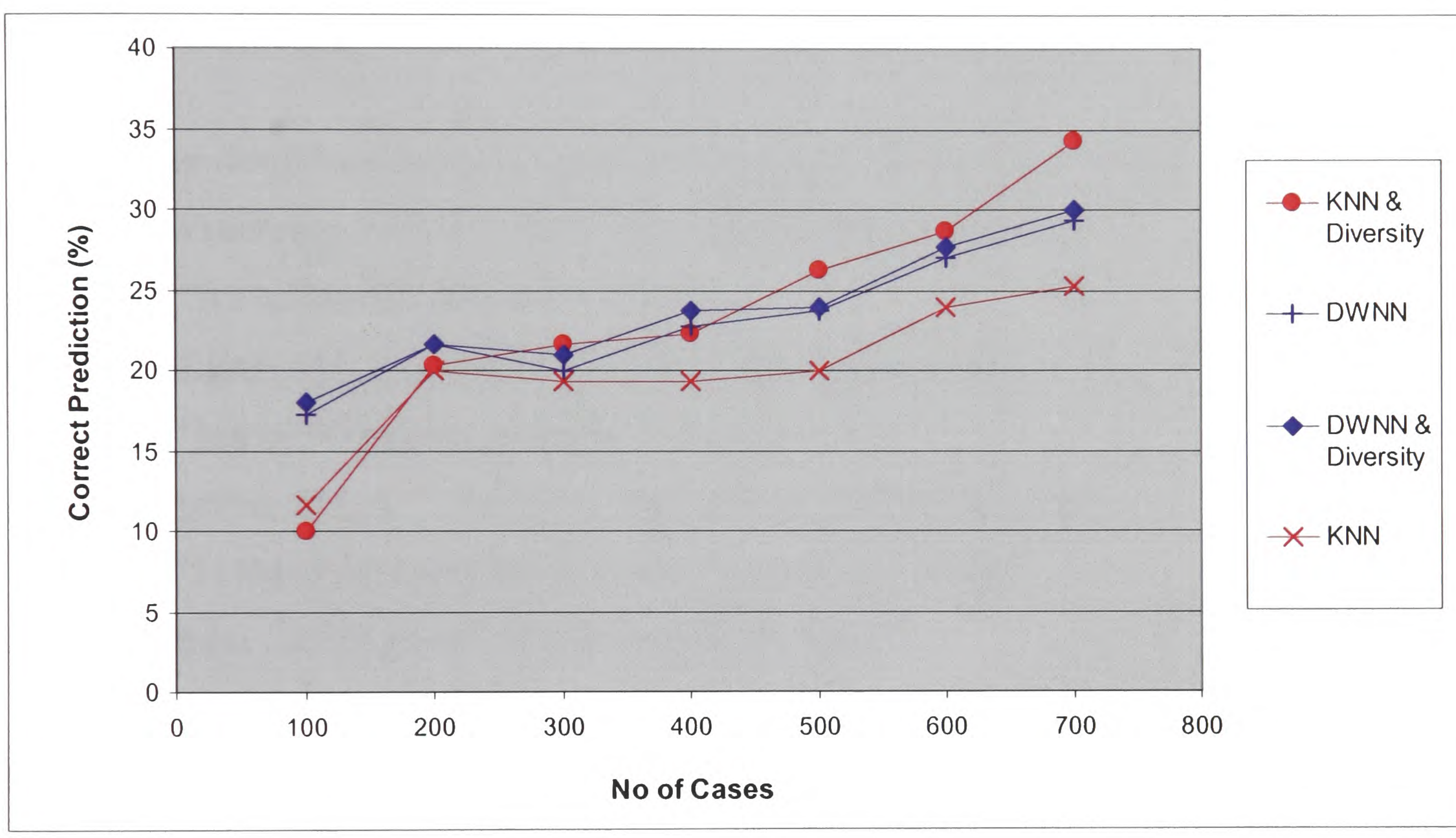


Figure 6-4. Comparing the correct prediction accuracy (%) of DWNN and *k*-NN using both diverse retrieval sets and nearest neighbour retrieval sets, on 300 unseen target problems

6.3.1 Illustrative Problem: Nearest Neighbour Retrieval Set versus Diverse Retrieval Set

Given the same case definition:

(JourneyCode, HolidayType, Price, NumberOfPersons, Duration, Season, Region, Accommodation, Transportation)

We take as target the "H.Flat Black Forest." hotel:

x = (152, Wandering, 449, 2, 7, May, BlackForest, HolidayFlat, Train)

For 3 nearest neighbours using the ***nearest neighbour retrieval set***, we get:

x_1 = (112, Wandering, 719, 4, 7, July, Harz, HolidayFlat, Train),

Hotel: "H.Flat Harz.", distance = 0.3797.

x_2 = (912, Skiing, 449, 2, 7, January, Bavaria, HolidayFlat, Car),

Hotel: "H.Flat Wildgatter, Bavaria.", distance = 0.5.

x_3 = (975, Active, 514, 6, 7, February, BlackForest, HolidayFlat, Car),

Hotel: "H.Flat Ferienpark Black Forest.", distance = 0.5011.

Prediction from GSNN gives: "H.Flat Ferienpark, Harz."

This example shows that GSNN failed to predict the hotel for package holiday x (i.e., JourneyCode = 152) correctly. The following shows that using a diverse retrieval set, GSNN can predict the same instance correctly.

For 3 nearest neighbours using the ***diverse retrieval set***, we get

x_1 = (112, Wandering,, 719, 4, 7, July, Harz,, HolidayFlat,, Train),

Hotel: "H.Flat Harz.", distance = 0.3797.

x_2 = (973, Wandering, 688, 2, 6, December, BlackForest,, Two stars, Car),

Hotel: "Landgasthof Schuetzen, Black Forest.", distance = 0.5042.

x_3 = (50, Recreation, 789, 5, 7, May, Fano,, HolidayFlat,, Car),

Hotel: "H.Flat Romo.", distance = 0.5059.

Prediction from GSNN gives: "H.Flat Black Forest." This example shows that GSNN can correctly predict the hotel for package holiday x (i.e., JourneyCode = 152). Note that the "H.Flat Black Forest." is a new hotel and is not in the retrieval set.

6.4 SIMULATED CASE BASE: RANDOMLY SPACED NODE SETS

According to the analysis in Section 5.6, we note that GSNN performs better in regularly spaced node sets than randomly spaced node sets because of the extrapolation trap. Here, we perform the same test as in Section 5.6.2 except that we use a diverse retrieval set for interpolation, using k-NN, DWNN and GSNN, to see if GSNN would give improved predictive performance. The results for various case base sizes are given in Fig. 6-5, which contrasts three methods using both diverse retrieval sets and nearest neighbour retrieval sets. We use α = 0.5, b=2 and k=3 to generate diverse retrieval sets. Here we see that GSNN when using in conjunction with a diverse retrieval algorithm shows improved predictive performance. DWNN however again improves only marginally because its prediction is dominated by the nearest neighbour. k-NN performs better than k-NN & Diversity because the latter's prediction is biased to the random selection of classification when votes are equally distributed for each distinct classification in the retrieval set (see Section 6.3).

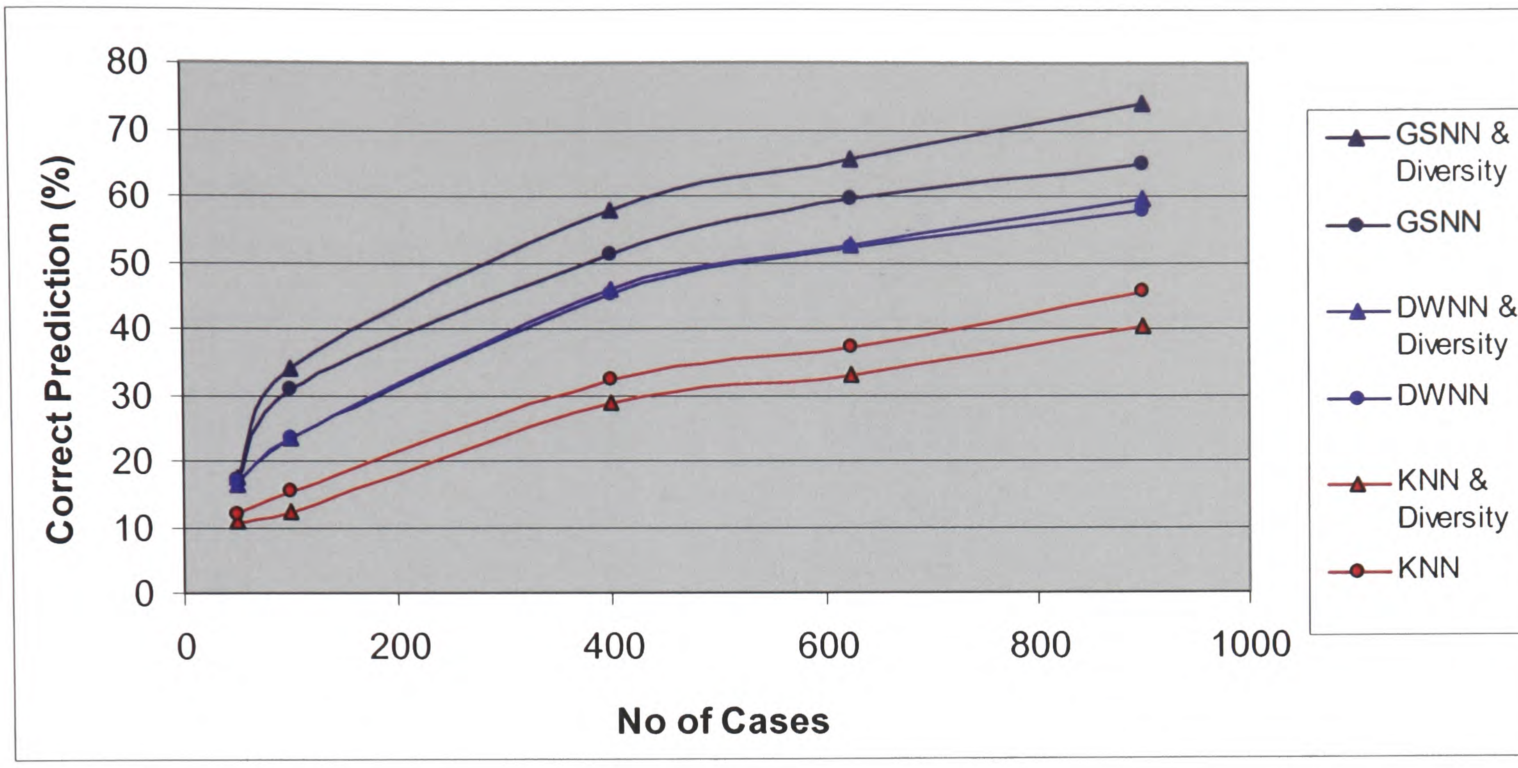


Figure 6-5. Comparing the correct prediction accuracy (%) of retrieval methods using both diverse retrieval sets and nearest neighbour retrieval sets, in estimating a test set of 1000 targets, for random case bases

6.5 SIMULATED CASE BASE: REGULARLY SPACED NODE SETS

We perform the similar test (see Section 6.4 with α = 0.5, b=2 and k=3) on the simulated regular case base. Since the case base is made up of regularly spaced nodes, the distribution of the cases is already quite diverse. We hence expect that GSNN performs better using a nearest neighbour retrieval set than a diverse retrieval set. Fig. 6-6 shows that GSNN when performed on a nearest neighbour retrieval set show better predictive performance compared to GSNN with the diversity method. Once again DWNN improves only marginally.

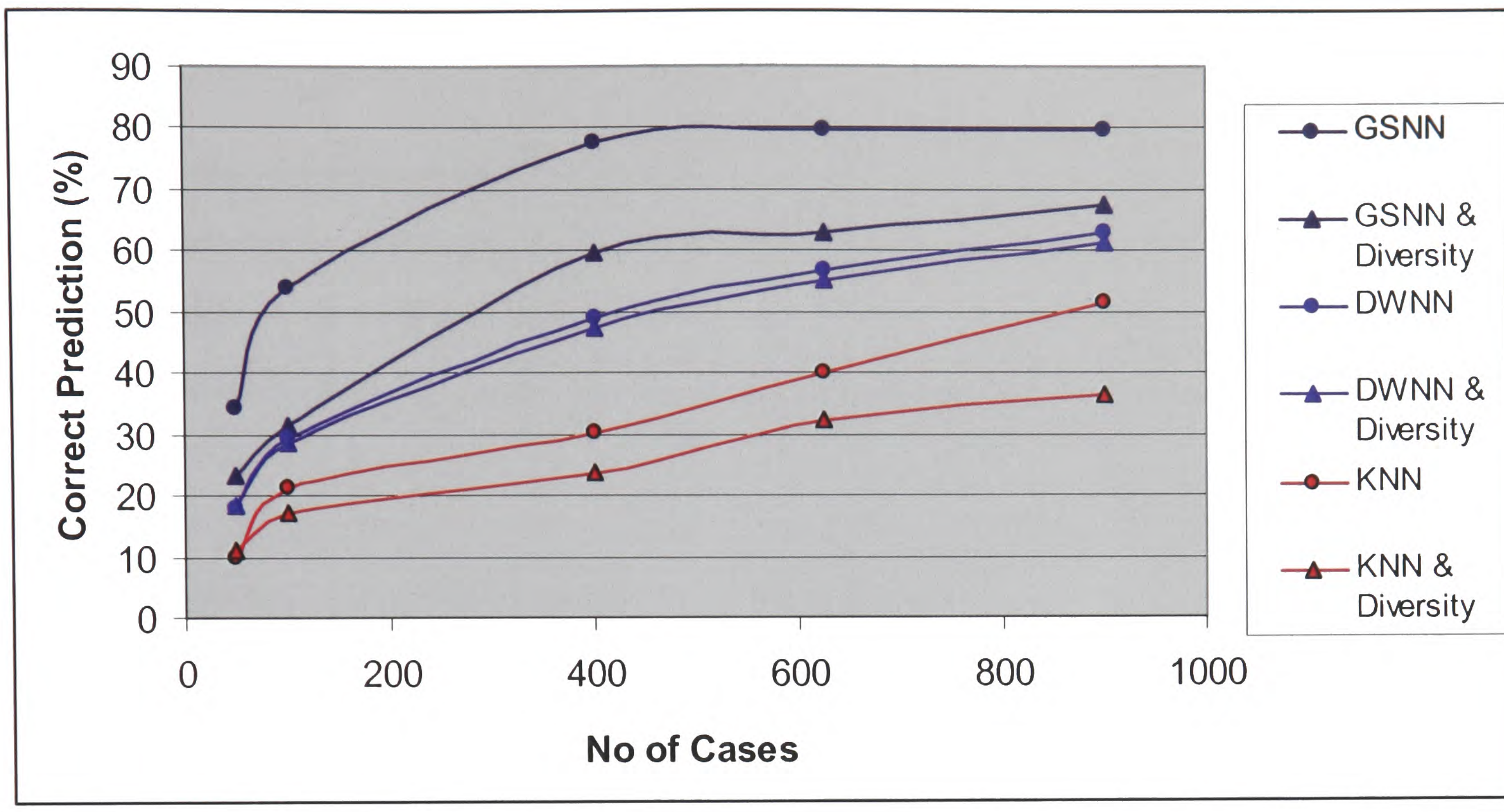


Figure 6-6. Comparing the correct prediction accuracy (%) of retrieval methods using both diverse retrieval sets and nearest neighbour retrieval sets, in estimating a test set of 1000 targets, for regular case bases

6.6 CONCLUDING REMARKS

In this chapter, we have proposed to use a diversity technique in [Smyth & McGinty, 2003] to generate a diverse retrieval set for interpolation. We have tested GSNN and DWNN on two test problems: a benchmark CBR case base from the travel domain and a benchmark simulated nominal case base. The examples studied indicate that GSNN can be more efficient as an adaptation engine than other nearest neighbour methods when used in conjunction with a diverse retrieval algorithm. This confirms the general property of GSNN that the method performs better at interpolation than extrapolation.

Once again, the results have shown that GSNN has two advantages over conventional nearest neighbour methods when interpolation is performed on diverse retrieval sets. First, it can predict solutions not in the case base. Second, it can predict solutions not in the retrieval set. Conventional nearest neighbour methods that use a voting function in the nominal domain prediction fail to predict any of these solutions.

Here we can see in fact that GSNN benefits from these advantages because it relies on distance metrics defined in the problem space and solution space. GSNN can give even better prediction when a diverse retrieval set is used for interpolation.

Chapter 7

Case Selection Techniques

7.1 INTRODUCTION

Instance-Based Learning (IBL) and CBR [Mitchell, 1997] share some common properties: (1) they implement "lazy" learning, (2) they classify new instances by nearest neighbour methods. IBL has historically focused on learning algorithms for case-based prediction. A thorough survey of IBL reduction methods can be found in [Wilson and Martinez, 2000]. On the other hand, the efficiency of the case selection process in CBR has been studied by several authors recently [e.g., Salamo and Golobardes, 2002; Yang and Zhu, 2001; Smyth and McKenna, 1998, 1999] (see Section 3.7).

For most case selection techniques, the importance of a case is often measured by its contribution to the predictive performance of a case base. For example, reduction in the Shrink algorithm [Kibler and Aha, 1987] is based on the contribution of cases to the predictive power of the case base; Smyth and McKenna (1999) use 'relative competence' to measure the importance of cases, and consequently provide a means to optimize case base performance. If a case can be predicted correctly by the case base, that case can be excluded from the case base. The reasoning behind this is that case base designers often assume that the case base is a representative sample of the problems in real world [e.g., Smyth & McKenna, 1998]. Consider the following: if a case can be correctly predicted by the case base, this also implies that even if it were taken away from the case base and then re-presented to the case base as a new problem, the case base is still capable to solve it. If a case cannot be correctly predicted by the case base, this implies that if it were taken away from the case base and re-presented to the case base as a new problem, the case base will not be able to give a correct prediction. Hence any case that has been wrongly predicted by the case base to some extent has certain contribution to the predictive performance of the case base. This indicates that the prediction accuracy of a case by its case base can somehow affect the retention of a case in the case base.

On the other hand, the prediction accuracy of each case in the case base could be influenced by the adaptation method used. The common adaptation methods used for prediction in case selection techniques are k-Nearest Neighbour (k-NN) and the Distance Weighted Nearest

Neighbours (DWNN) [Mitchell, 1997; Wilson and Martinez, 2000]. In Chapter 5, a variety of test problems have shown that the choice of an interpolation method used for adaptation affects the prediction accuracy of a case base. Analyses in Chapter 5 show that GSNN can out-perform these nearest neighbour methods on these test problems and it is due mainly to the fact that it assumes distance metrics defined in solution space. Conventional nearest neighbour methods however operate on a voting mechanism. The voting mechanism has a disadvantage such that it fails to produce any intermediate values not present in the retrieval set (see Section 5.1). Boolean expression of the voting mechanism has been shown to be less efficient as an adaptation engine.

From the above reasoning, we can conclude that there may be two possibilities, which will influence the case selection process:

First, the performance of nearest neighbour methods may result in different case bases due to their different predictions of a target query. For example, GSNN gives correct prediction for a given case and therefore suggest to the case base designer that the case can be excluded from the case base while k-NN and DWNN may not agree with this. This may result in different case bases produced by different nearest neighbour methods. It is possible that the use of GSNN may produce a smaller case base than k-NN and DWNN for similar prediction accuracy since GSNN is a more efficient adaptation engine.

Second, in most case selection techniques such as the decremental approach (see Section 3.7), cases are often removed based on a pre-defined removal criterion. In the Shrink algorithm, the removal criterion is defined as any case in the case base that can be correctly predicted by other cases in the case base should be removed. The voting mechanism gives away little information (i.e, Yes or No) as to which cases should be removed. On the contrary, with GSNN, the distance metric gives extra information since it is a measure of how close the predicted value is to the true solution.

In this chapter we propose a case selection technique, which uses a metric defined on the solution space used in the GSNN algorithm (see Chapter 5) to estimate the output classification (e.g., nominal or real-valued solutions) of a given target query, and then to estimate the contribution of each case to the predictive power of the case base. The case selection technique subsumes some existing decremental methods, such as the Shrink algorithm. A ten-fold cross validation test [Witten and Frank, 2000] is performed on two case bases of different kinds, with several metrics proposed in the solution space. This includes the Iris data set [Fisher, 1936] and on a pneumatic conveying particles problem to examine the predictive power of case bases at different stages in a decremental fashion. The former is a classification problem. The latter is a practical problem that exists in pneumatic conveying industry (see Section 2.3 and Chapter 9).

In the experiments described in this chapter, we have included reductions in the case base to very small sizes. These reductions have been at the expense of case base competence. In general we would not expect to reduce case bases to such a level that competence is reduced at all. There is sufficient evidence at the competence preserving levels, to show that GSNN redundancy elimination is advantageous whilst preserving case-base competence. For example, in Table 7-4, we see that GSNN can still perform at 93% competence, even with only 3 cases, against DWNN and k-NN with only 50% and 30% competence respectively.

The test results in Section 7.3 will show that such a reduction technique, which uses a solution space metric, performs well. The aims of the tests are to investigate the feasibility of using such a reduction technique to extract an optimum case base from data provided by the user, and to use the simulation results as a bench mark to evaluate the predictive power of the reduced case bases. The test results confirm that GSNN can out-perform k-NN and DWNN on these problems with several measures, depending upon metrics defined in the solution space. Furthermore, due to the inherent nature of GSNN, a case-removal order can be defined in terms of the GSNN error function. When implementing the case-removal order, good predictive power can still be achieved even with very sparse case bases.

7.2 USING GSNN AS CASE SELECTION CRITERION

The Generalized Shepard Nearest Neighbour algorithm is an adaptation method, which has two features of interest in case base reduction. It is an interpolative method capable of operating over nominal values, which shows good predictive power for sparse case bases (see Chapter 5). Also, it allows us to exploit knowledge of any distance metric defined over the solution space. We show in this section that this feature is of use in case base reduction, in that the solution space similarity can be used effectively in ordering case removal. The GSNN algorithm forms a basis for a reduction technique proposed in this section incorporating solution space metrics. The use of solution space metrics can achieve a sparse case base with good predictive power.

The algorithm assumes that there exists a distance metric $d_x(x,x_i)$ in the X (problem) space and also a metric $d_y(y,y_i)$ in the Y (solution) space (see Chapter 5). GSNN employs an error function $I(y)$ constructed from k nearest neighbours

$$I(y) = \Sigma_1^k d_y(y,y_i)^2 d_x(x,x_i)^{-p} / \Sigma_1^k d_x(x,x_i)^{-p} \ . \qquad \text{E(5-2)}$$

$I(y)$ can be interpreted as a measure of the deviation of the value y from its prediction based on its k nearest neighbour. $I(y)$ is defined entirely in terms of distance metrics $d_x(x, x_i)$ and $d_y(y, y_i)$, taking the form of E(5-2) (see Section 5.3). $I(y)$ may be seen as the distance weighted mean square deviation of y from the neighbouring solutions y_i . We take $p = 1$ for all tests in this chapter. By GSNN, $\hat{y}$ is $y \in Y$, which minimizes the error function $I(y)$.

$$\hat{y} = \arg \underset{y \in Y}{Min}\, I(y) \ . \qquad \text{E(7-1)}$$

The GSNN algorithm has more general applicability than k-NN and DWNN, allowing general solution space similarity metrics. We can in fact show that a special case of GSNN is in fact DWNN. Take the case that the solution space Y is the real number domain, and take $d_y(y,y_i) = |y - y_i|$. In this case, we can calculate the minimum value of $I(y)$ from the condition $\partial I / \partial y = 0$. This gives: $y \sum_1^n \| x - x_i \|^{-p} \; = \; \sum_1^n \| x - x_i \|^{-p} \, y_i$

and we see that y is given by a distance weighted average value, identical with DWNN. However, GSNN differs markedly from DWNN when the solution space is discrete, whereupon DWNN resorts to a voting mechanism. GSNN will also allow more general solution domains, such as vector domains with continuous or mixed elements, as long as a metric is defined on Y.

In order to use GSNN in case base reduction methods, and to compare it against other nearest neighbour methods, we first need to re-cast such a technique, being cognizant of the special feature of GSNN, viz. that it can exploit the existence of a solution based metric. In this section, we show how one such method, the decremental method can be generalized so that GSNN can be used in substitution for other nearest neighbour methods in case base reduction.

First we review the general reduction method. If we take D be the domain of cases, C the initial set of cases, R the Reduced set, S to be a candidate set, and $S \subseteq R, c \in D, C \subseteq D, R \subseteq C$, we can formulate the general reduction algorithms as :

```
Function R(C):
Begin
        R = C
        While CandidateSet(R) ≠  Null
          R ← R - Select(CandidateSet(R))
        loop
End
```

Reduction techniques such as the Shrink algorithm reduce the case base according to this general algorithm, so that each iteration of the algorithm produces a case base with one less case. To achieve this goal, a removal candidate set from the current R is obtained. The function CandidateSet (R) returns a removal candidate set of cases. The function Select (S) returns a case: Select (S) $\in S$, to be removed. In all the reduction methods discussed in Section 7.1, this is done by using a nearest neighbour algorithm to obtain an estimated solution $\hat{y}$ for each case $c \in R$. Inclusion in the candidate set is then based on the difference between $\hat{y}$ and the true solution y_t *for each case* $c \in R$. The candidate set is usually taken to be the set of cases such that

$$\left| y_t - \hat{y} \right| = 0 \qquad \text{E(7-2)}$$

Since these cases satisfy E(7-2) y_t is perfectly predicted by its nearest neighbours. These reduction methods add little predictive power to the case base.

The second phase of the removal strategy is represented by the Select() function. Usually this is a random selection of a single case from the candidate set. If the candidate set is empty, the algorithm will halt. E(7-2) plays a central role in the reduction technique. It sets a condition to define the candidate set. We here generalize this rule by the condition:

$$\left| y_t - \hat{y} \right| = Min_{\ y \in R} \left| y_t - y \right| \ . \qquad \text{E(7-3)}$$

On the right of this equation is the minimum value of the difference between $\hat{y}$ and the true solution y_t for cases in R. This difference could be 0 or positive. Equation E(7-3) will be satisfied by all cases with a difference equal to this minimum. This set is therefore the candidate set, which must always contain at least one member while R itself is not empty. We can therefore continue the reduction to produce case bases of any size q, as long as $|R| > q$. Since when $|R| < k$ (k is the number of the nearest neighbours in the retrieval set) there are not enough interpolation points to produce estimates $\hat{y}$ for any case in R, q is supposed to be greater than k. The general reduction algorithm now takes a form of the following:

```
Function R(C):
Begin
        R = C
        While CandidateSet(R)> q
         R ← R - Select(CandidateSet(R))
        loop
End
```

In the further generalization for the use of GSNN, the simple differences in E(7-3) is replaced with a generalized solution space metric $d_y(y, y')$:

$$d_y(y_t, \hat{y}) = Min_{\ y \in R}\, d_y(y_t, y) \quad . \qquad \text{E(7-4)}$$

7.3 EXPERIMENTAL RESULTS

To examine how the GSNN algorithm might work on practical applications, and to measure its performance when it is embedded in a reduction method, we performed a trial test on the Iris dataset [Fisher, 1936] and on a pneumatic conveying particles degradation dataset (see Section 2.3 and Chapter 9). The reduction method is used to find a minimum case base.

We have chosen the Shrink algorithm from among reduction methods available due to its decremental fashion of eliminating cases. The Shrink algorithm begins by placing all training instances into a case base and then prunes the case base by removing instances that are correctly classified by the remaining subset. It is somewhat similar to RNN [Gates, 1972] except that RNN considers whether the classification of other instances would be affected by the instance's removal. Like some other reduction methods, the Shrink algorithm retains boundary cases but it is sensitive to ordering effects.

The fact that GSNN takes advantage of interpolative mechanism, and makes use of solution-space metric, will be further highlighted later on in this section. GSNN will prove itself to be a complement to the Shrink algorithm. With GSNN, a removal order can be implemented using formula E(7-4) in the Shrink algorithm to further remove the cases to find a minimum case base.

The benefits of the algorithms proposed here, in comparison with the many existing case reduction algorithms are: (i) the algorithm can exploit the existence of a similarity metric in the solution space. This applies particularly for nominal values, for instance, with the Iris classification problem shown in Section 7.3.1, (ii) this algorithm can achieve sparse case bases with good predictive power.

To verify the above argument, the standard 10-fold cross validation test has been performed for analysis with different k values where k is the number of nearest neighbours retrieved.

In the 10-fold cross validation test, cases in the dataset are randomized and then partitioned into 10 sets. Each set is in turn used as a test set while the other sets are used as training set. We repeat this 10 times since there are 10 partitions. The test results presented here are the average results of the 10 test sets. Also we used the same partitions in all tests. This test examines two aspects of the method: (i) Case base storage size, and (ii) Prediction accuracy. We compare the performance of GSNN using two different solution space metrics (GSNN-1 and GSNN-2) with both the standard k-NN and DWNN method.

7.3.1 Test I: The Iris Classification Problem

Consider the Irises problem discussed in Section 5.5. We here use the same problem space distance metric defined in Section 5.5. In GSNN-1, the solution space distance metric is defined as $d_y(y, y') = 0$ if $y = y'$, and $d_y(y, y') = 1$ otherwise. This metric corresponds to the view that the three Iris classes are each totally dissimilar to each other. It is interesting to notice that GSNN-1 behaves like DWNN in all tests. This suggests that DWNN (and k-NN) also implicitly treat solution classes as mutually dissimilar.

In GSNN-2 we take the Iris classes to bear some similarity to each other. Fig. 5-3 (see Section 5.5) shows a principal component plot of the dataset. Although this is only a 2-dimensional representation, it indicates that class Versicolour lies somewhat "between" classes Setosa and Virginica (although not *linearly* between). To reflect this view of the classes bearing some similarity to each other, we use the same distance metric defined in Section 5.5 for the solution space, Y.

Table 7-1. Average Correct Predictions in estimating the Iris dataset using 10-fold cross validation

Embedded methods in the Shrink algorithm	Case base size	k-NN (%)	DWNN (%)	GSNN-1 (%)	GSNN-2 (%)
k=1	135	92.67	92.67	92.67	92.67
	19	87.33	87.33	84.67	87.33
	3	52.67	52.67	64.67	83.33
k=2	135	92.67	92.67	92.67	92.67
	19	84.0	87.33	87.33	87.33
	3	34.0	52.67	60.67	84.0
k=3	135	94.67	94.0	94.67	94.67
	19	92.0	88.0	89.33	90.67
	4	50	42.0	61.33	66.0
k=4	135	94.67	94.67	94.67	94.67
	19	92.0	89.33	89.33	89.33
	5	52	56.0	70.67	67.33
k=5	135	94.0	94.67	94	94
	19	91.3	90.0	94	92.67
	6	36.37	40.67	59.33	58.67

Table 7-1 shows the results of these tests. The Shrink algorithm was used to find reduced case bases (R). For k=1 and 2, we use a minimum case base size of 3. For k=3, 4 and 5, since we need $|R| > k$ interpolation points to produce estimates $\hat{y}$ for any case in R, the minimum case base sizes are chosen as 4, 5 and 6 respectively. The table shows how effective these reduced case bases were in classifying a target query, using 10-fold cross validation, in comparison with the complete case base and various retrieval methods. These results confirm that GSNN, which uses solution-space metrics, can out-perform the two other nearest neighbour methods for case bases using the Shrink algorithm. For all k values (i.e., 1 to 5), GSNN demonstrates better classifying power than either k-NN or DWNN for the smallest reduced set. In particular, the GSNN-2 sparse case bases with only 3 cases for k= 2 achieves 84% average correct predictions compare to 34% for k-NN or 52.67% for DWNN.

7.3.2 Test II: Pneumatic Conveying Particles Degradation

CBR models are currently under investigation for estimation of conveyor design quality (The setup of this problem here is slightly different from the one discussed in Chapter 9 in that these experiments were run prior to the results obtained in Chapter 9. At the time, limited information was available.), which may affect particle degradation and erosion of pipe bends. However, experimental cases are expensive to produce, and suitable sparse case-bases are desirable. Therefore, a conveyor simulation model, which simulates pneumatic conveying of sugar, instead of real experimental tests has been constructed and 50 cases in the form of <problem, solution> pairs have been obtained from the simulation model. These cases, each with a unique solution to each problem, are generated at random by a user.

The problem domain consists of the attributes: velocity of air and velocity of particles at the outlet, distribution of particles class size at the outlet. The solution domain consists of attributes: velocity of air and suspension density at the inlet and the angle of a bend. This problem is formulated based on the user's interest in finding appropriate parameters for a desired distribution of particles in different size class. The distance metric in both the problem domain and solution domain are computed by using the weighted sum of attributes with equal weight on all attributes.

The solution domain is a mixed domain with a vector of y = (angle of a bend, air velocity, suspension density), where the angle of the bend is discrete. The angle can only be 45 degree and 90 degree according to the user. A simple mean value for k-NN and the weighted mean for DWNN in prediction of the target query would not be appropriate in this problem. Also, it is important to note that the setup of the present case base assume single-valued solution, with only one unique solution corresponding to each problem presented in the case base. This might affect the predictive power of the case base when using a cross validation test because both the k-NN and the DWNN methods will never give the exact solution because the test set is separated from the training set. Therefore, we use a classifier with a given tolerance, which can be used in predicting solutions of the test set. We classify a case with error outside the tolerance as incorrect, and otherwise it is classified as correct. The tolerance used in this problem $\varepsilon = 0.163$ is arbitrary selected.

Table 7-2. Average Correct Predictions obtained from several different retrieval methods using a classifier based on the average error with a tolerance, $\varepsilon = 0.163$

Embedded methods in the Shrink algorithm	Case base size	k-NN (%)	DWNN (%)	GSNN-1 (%)	GSNN-2 (%)
k=1	45	60	60	60	60
	30	58	58	58	58
	5	32	32	32	32
k=2	45	50	60	60	66
	30	44	58	58	64
	5	24	32	32	36

In GSNN-1, we use a distance metric $d_y(y, y') = 0$ if $y = y'$ (i.e., $y - y' <= \varepsilon = 0.163$), $d_y(y, y') = 1$ (i.e., $y - y' > \varepsilon = 0.163$) otherwise. In GSNN-2, the solution space distance metric is computed using the standard weighted sum method with equal weight for all attributes. The predictive quality of the case bases generated by the reduction method with k-NN, DWNN and GSNN respectively is listed in Table 7-2. As mentioned in Section 7.3.1, GSNN-1 behaves like DWNN in all tests. This is due to the inherent nature of the data set collected from the user and the characteristic of the solution space distance metric used in GSNN-1. The data set consists of 50 unique <problem, solution> pairs in which we would expect both DWNN and GSNN-1 give the same result regardless of the value of k since they both weighted the contribution of each of the k nearest neighbours by the problem space distance and the solution space distance of $\{0,1\}$. Thus, we would expect the nearest case to the target query will always be the estimated value of the target solution. However, for $k > 1$, GSNN-2 with the case base size 30 and 5 seems to out-perform k-NN by approximately 50%.

7.4 THE CANDIDATE SET CONDITION

In this section we examine the effect of the candidate set condition on the reduction algorithm. The candidate set conditions E(7-3), E(7-4) have the general form:

$$M(y_t, \hat{y}) = Min_{y \in R} M(y_t, y) \qquad \text{E(7-5)}$$

where $M(y_t, \hat{y})$ is a measure of how important y_t is to the case base's predictive power. In Section 7.2, the measure M was taken in the form of $M = d(y_t, y)$ which is a distance metric in the solution space. However a better gauge of how important y_t is to the case base is a measure of how close y_t is to being selected.

By GSNN, We know from E(7-1) and E(5-2) that:

$$\hat{y} = \arg \underset{y \in Y}{Min}\ I(y) \ .$$

Hence, $I(\hat{y}) = Min\,(I(y))$ is a measure of the deviation between the estimated value and its k-neighbour predictions. Similarly $I(y_t)$ is a measure of the deviation between the true value and its k-neighbour predictions. Hence the expression: $I(y_t) - I(\hat{y})$ can be a measure of how near y_t is to being selected. If $I(y_t) - I(\hat{y}) = 0$, then y_t will definitely be selected . If $|\ I(y_t) - I(\hat{y})\ |$ >> 0, then y_t is a long way from being selected.

We hence suggest that the measure

$$M = \left| I(y_t) - I(\hat{y}) \right| \ . \qquad \text{E(7-6)}$$

is such a measure to describe how close y_t is to being selected. Substituting E(7-6) into E(7-5), we have

$$\left| I(y_t) - I(\hat{y}) \right| = Min_{y \in R} \left| I(y_t) - I(y) \right| \qquad \text{E(7-7)}$$

To investigate the effectiveness of this newly proposed candidate set criterion, we re-ran the Shrink tests of the previous section by taking $M = \left| I(y_t) - I(\hat{y}) \right|$ in the candidate set criterion.

7.4.1 Test I with Alternative Candidate Set Conditions

The results of the Iris classification problem are given in Table 7-3. They show a marked improvement on the predictive power of a case base produced with $M = \left| I(y_t) - I(\hat{y}) \right|$. Note that the results of GSNN shown in Table 7-3 with implementing the candidate set condition E(7-7) in the removal of cases are not only better than using E(7-4) where a distance metric in the solution space was used, but also far better than the prediction of using *k*-NN and DWNN retrieval. As shown in Table 7-4, when *k*=1, the use of E(7-7) is 1.8 times better than *k*-NN and DWNN and 2.7 times better than *k*-NN and 1.8 times better than DWNN when k=2.

Table 7-3. Average Correct Predictions in estimating the Iris dataset for GSNN-2 –Shrink algorithm with different removal strategies using 10-fold cross validation

$M(y_t, \hat{y})$	Case base size	*k*=1	*k*=2
$d(y_t, \hat{y})$	3	83.33%	84%
$\left| I(y_t) - I(\hat{y}) \right|$	3	96%	93.33%

Table 7-4. Average Correct Predictions in estimating the Iris dataset for GSNN-2 –Shrink algorithm with removal strategy: E(7-6) and different retrieval methods using 10-fold cross validation

Retrieval methods	Case base size	*k*=1	*k*=2
GSNN-2 with removal strategy: $\lvert I(y_t) - I(\hat{y}) \rvert$	3	96.0 %	93.33%
k-NN	3	52.67%	34.0 %
DWNN	3	52.67%	52.67%

7.4.2 Test II with Alternative Candidate Set Conditions

We here repeat the same test on the pneumatic conveying particles problem. Note that again the results shown in Table 7-5 by implementing the candidate set condition E(7-7) in the removal of cases are better than using E(7-4), and gives much better prediction than using *k*-NN and DWNN retrieval as shown in Table 7-6.

Table 7-5. Average Correct Predictions in estimating the Pneumatic conveying particles dataset for GSNN-2 –Shrink algorithm with different removal strategies using 10-fold cross validation

$M(y_t, \hat{y})$	Case base size	*k*=1	*k*=2
$d(y_t, \hat{y})$	5	32.0 %	36.0 %
$\lvert I(y_t) - I(\hat{y}) \rvert$	5	42.0 %	42.0 %

Table 7-6. Average Correct Predictions in estimating the Pneumatic conveying particles dataset for GSNN-2 –Shrink algorithm with removal strategy: E(7-6) and different retrieval methods using 10-fold cross validation

Retrieval methods	Case base size	k=1	k=2
GSNN-2 with removal strategy: $\left\| I(y_t) - I(\hat{y}) \right\|$	5	42.0 %	42.0 %
k-NN	5	32.0 %	24.0 %
DWNN	5	32.0 %	32.0 %

7.5 CONCLUDING REMARKS

In this chapter we have set out a case base reduction technique, which uses a metric defined on the solution space. Firstly, the capability of the GSNN algorithm embedded in the reduction technique was investigated and through two test cases, the classification of the Iris case base and pneumatic conveying particles dataset, it has been demonstrated that GSNN out-performs standard nearest neighbour techniques. Secondly, a generalized reduction criterion has been proposed in terms of $M(y_t, \hat{y})$, *which represents a measure of how close a potential candidate is to being selected. A realization of the technique is to define M in terms of the GSNN error function, I(y) and it has been demonstrated that this version of the technique improves the performance of the reduction method and the improvement over the standard nearest neighbour techniques is significant.*

PART III

CASE STUDIES

Chapter 8

Case Study I: CBE-Projectile

Chapter 9

Case Study II: CBE-Conveyor

Chapter 8

Case Study I: CBE-Projectile

8.1 INTRODUCTION

The projectile problem has been discussed in Chapter 2 as part of the motivation of this research. In Section 4.4 it is again used to illustrate how the Generalized Nearest Neighbour Retrieval (GNNR) method works. Here we use the same example to illustrate other aspects of CBE, such as interpolation. We repeat here some of the tables and results of Chapter 4, for ease of readability.

In this chapter, we describe the development of a CBE (Case-Based Engineering) application for the projectile design problem. We termed the CBE system for the projectile design problem as CBE-Projectile. The CBE-Projectile system is implemented using the CBE architecture discussed in Section 2.5.1. There are three agents in the CBE-Projectile system: a gunner, a CBR model and a projectile model. These agents collaborate together by sharing their expertise to solve the projectile design problem. For example the gunner can use the CBR system as a flexible query engine to search for a solution. Using the CBR system, the gunner can specify design problems by defining constraints on the unified input and output space. The CBR system will return the best match to the target problem. If the best match requires adaptation, the CBR system can perform interpolation using retrieved cases to give a prediction. The gunner can then cross check the CBR system's prediction by running the projectile model.

In Section 8.2, we give an overview of the projectile model. In Section 8.3 we present the findings of the development of the CBE-Projectile application and in Section 8.4 we show an example of how the CBE-Projectile solves an inverse problem versus a human expert. Results show that the CBE-Projectile system is faster and more efficient than the human expert who uses the projectile model directly to search for a solution. In Section 8.5, we discuss the advantages of the CBE-Projectile system over the traditional approach used by the gunner. We conclude in Section 8.6, with a summary of the chapter.

8.2 THE PROJECTILE MODEL

Fig. 4-2 presents the projectile model that simulates the flight of a cannonball shot over flat ground in a given time interval, T. There are two inputs: v = initial velocity and θ = angle of gun for the model. The output of the model is a trajectory of the shot, which is represented by a set of points (x, y) (see Section 2.2 and Section 4.4). A typical engineering problem faced by the gunner is this: Can the right angle of gun and initial velocity be determined so that the cannonball will hit a target on the ground at x = 450m? Given a range of possible velocity values and angle of gun with $v \in [70,100]$ and $\theta \in [0.2,1.4]$, the gunner wants to find out what is the best set-up to use.

In Chapter 2, we have shown that indeed such a problem can be formulated in the form of a predicate: $P\ (I_1,\ I_2,\ O_1)$, where $I_1 = \theta$ is the angle of the gun, $I_2 = v$ the initial velocity, and $O_1 = x$ the projectile target. As referred to the above example, the predicate is:

$$P\ (\ ?\ I_1 = \theta,\ ?\ I_2 = v,\ O_1 = 450m)\ .$$

In this chapter, we want to show how the CBE architecture can be used to help the gunner to solve such a problem.

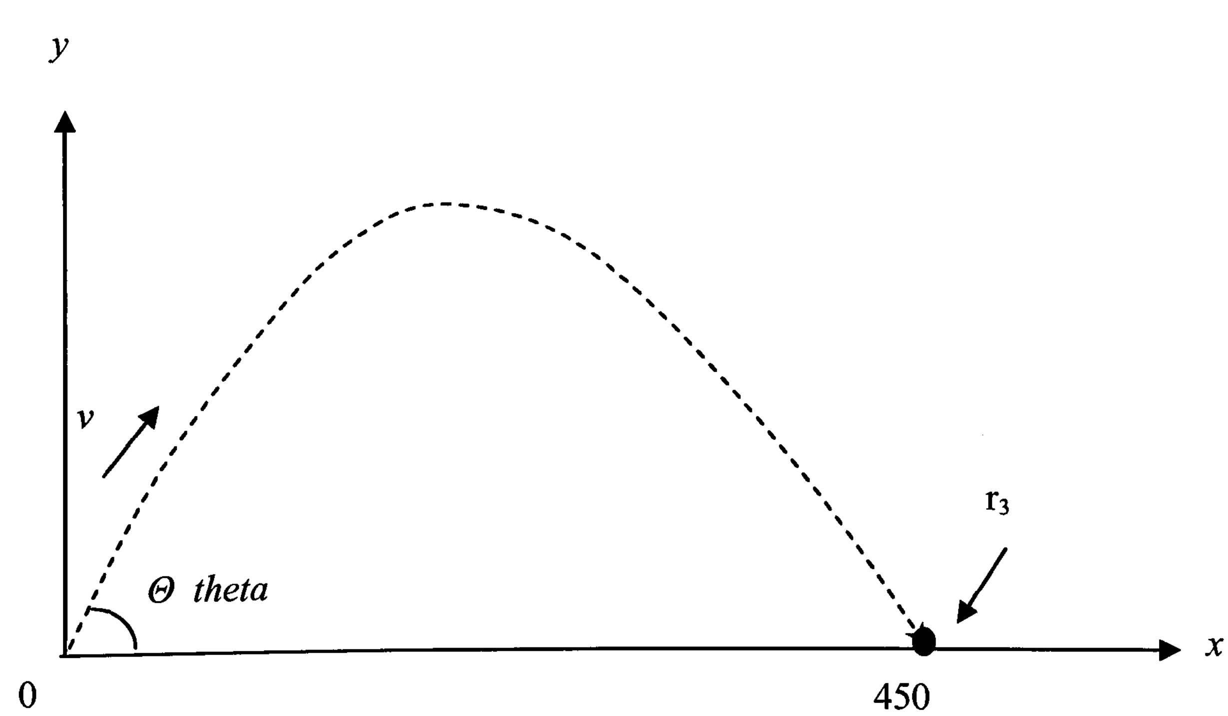


Figure 8-1. Visualization of the trajectory of a cannonball shot over flat ground

8.3 DEVELOPMENT OF CBE-PROJECTILE

In CBE-Projectile, cases are represented by the predicate in the unified problem:solution space (see Section 4.4). Let c be a case such that $c \in C$ the case base. We notate each case attribute as $c_1, c_2, \ldots, c_n$; where n is an integer. Following the discussion in Section 8.2, we now re-present the problem with new notations such that: $c(c_1, c_2, c_3) = P(\theta, v, x)$. For the case base, we have generated fifty six cases with $\theta \in (0, \pi/2]$ and velocity $v \in [70, 100]$. In what follows, we show approaches used in the development of the CBE-Projectile system. This requires us, first of all, to examine the similarity metric on the query space.

8.3.1 Metrics Defined on the Unified Query Space

Since each case $c \in C$ is defined on the unified problem:solution space, we need to define a similarity metric over the whole unified space. The similarity between a target case and cases in the case base is computed using the standard weighted sum method [Kolodner, 1993] for all case attributes. For this problem, all case attributes (i.e., $c_1 = \theta$, $c_2 = v$, $c_3 = x$) are real-valued. Nearest neighbours to a target case is approximated using the Generalized Nearest Neighbor Retrieval (GNNR) method (see Chapter 4) over the whole unified space. Distance between two points for each attribute is normalized by the range value for the attribute.

8.3.2 The Gunner's Problem: *?θ, ?v, x=450m*

Following the above discussion, we now examine the gunner's problem. We have a target constraint such that $c_3 = x = 450$m. Indeed the gunner's problem is an inverse problem. The gunner wants to know what inputs combination of θ and v will hit the projectile target at $x = 450m$. Table 8-1 shows a list of cases retrieved ordered by minimum distance (i.e., between the gunner's problem and the retrieved cases in the case base).

Table 8-1. List of cases retrieved ordered by minimum distance for the gunner's problem

Case ID	$c_1 = \theta$	$c_2 = v$	$c_3 = x$	Distance
29	1	70	445.556	0.00451
51	0.4	80	459.108	0.00924
19	1.2	80	432.296	0.018
38	0.8	70	489.791	0.0403
56	0.2	100	389.418	0.0614
35	0.8	60	359.847	0.0914

From Table 8-1, we can see that the best match to the target is Case 29. The second best match is Case 51. Here, we have two possible set of solutions that are near to the target. Now the gunner can use these two cases for interpolation since they are the two nearest neighbours to the target.

However, we argued in Section 4.6 that to make sure that we fully take advantage of interpolation schemes such as GSNN, it is important, first of all, to examine whether these candidate cases which are not only close in the output but also in their inputs, i.e., whether they are close in the unified space, so that a better prediction can be achieved after interpolation. Note that although the interpolation with Case 29 and Case 51 produce a third case with c_3 near to the target (= 450m), the two cases are far apart in their inputs. On the other hand, there is a more relevant case (= Case 38) which can be used for interpolation. We see that Case 38 produce an output c_3 = 489.791, which is further apart from the target compared to both Case 29 (i.e., c_3 = 445.556) and Case 51 (i.e., c_3 = 459.108). Nevertheless, Case 38 is closer to Case 29 than Case 51 in the unified space. The distance between Case 29 and Case 38 is 0.0705 in the unified space. The distance between Case 29 and Case 51 is 0.213 in the unified space. (Note: The distances are calculated uses the weighted sum method with equal weight for all case attributes and normalized by the range value for each attribute as specified in Section 8.3.1) Hence, we expect that the pair (Case 29, Case 38) gives better prediction compared to the pair of (Case 29, Case 51).

To examine the results given by these cases we perform two tests: one for the conforming set (Case 29 and 38) and the other for the non-conforming set (Case 29 and 51). Here we refer a conforming set as cases that are conformable for interpolation which means 'near cases in the problem space are also near in the solution space'.

8.3.2.1 Interpolation over a Non-Conforming Set

First, we perform interpolation using Case 29 and 51. Table 8-2 show the interpolated result based on these two cases using the GSNN method (see Chapter 5). We see that the modeled result with c_3 = 507.166 shows the interpolation has failed to fulfill its goal, i.e., to produce a better prediction for the target $c_3 = 450$.

Table 8-2. Interpolated value and projectile model solution obtained from a non-conforming set: Case 29 and Case 51

Non-conforming set			
Case Id	$c_1 = \theta$	$c_2 = v$	$c_3 = x$
29	1	70	445.556
51	0.4	80	459.108
Interpolated value			
-	0.8846	71.92	448.163
Model solution			
-	0.8846	71.92	507.166

8.3.2.2 Interpolation over a Conforming Set

Table 8-3 shows that the interpolation based on Case 29 and Case 38, is far better than the non-conforming set. As expected, the conforming set gives a much better interpolation result - ($c_3 = 446.559$) than the non-conformable set ($c_3 = 507.166$).

Table 8-3. Interpolated value and projectile model solution obtained from a conforming set: Case 29 and Case 38

Conforming set			
Case Id	$c_1 = \theta$	$c_2 = v$	$c_3 = x$
29	1	70	445.556
38	0.8	70	489.791
Interpolated value			
-	0.9975	70	446.103
Model solution			
-	0.9975	70	446.559

These two examples illustrate the merits of conformable candidate cases for interpolation. The metric over the unified problem:solution space is a better measure than standard query space (i.e., problem) for finding a set of good candidate cases for interpolation in problems where multi-valued solutions are assumed. Without it, it would be difficult to spot the hidden better pair of candidates for interpolation when the top two are obvious candidates. This will guard against the poor selection of interpolation candidates from among the qualified candidates in the case base. Using the unified space, we can rely on the adaptation scheme to give a better prediction.

Ideally when adaptation is required, the CBE-Projectile system model would provide useful advice in selecting good candidates for interpolation. Without taking precaution against poor candidates, interpolation methods may not be fully exploited for prediction accuracy.

8.4 CBE-PROJECTILE VERSUS HUMAN EXPERT

In this section, we measure the performance of the CBE-Projectile system versus a human expert in solving inverse problems and constraint problems in the projectile design problem. The human expert is a numerical modeler who has more than 15 years of modeling experiences. The following test case presented demonstrates a typical approach used by the expert to solve inverse problems.

8.4.1 The Gunner's Problem

Consider the gunner's problem:

$c_3 = 450\text{m}$

Table 8-4 shows attempts made by the human expert to achieve the target constraint. The expert first ran a simulation (i.e., i=1) assuming that c_1 = 0.2 and c_2 = 100m/s. The strategy taken by the expert to search for the required solution is to start the search with the low end of the given range of the angle and the top end of the given range of the initial velocity of the canon ball. Then the target may be achieved by increasing the angle bit by bit or adjusting the initial speed gradually according to the results shown in column 5 of Table 8-4.

Table 8-4. Results of iterative runs by expert to solve the gunner's problem

Target: c_3	i=No. of simulation	c_1	c_2 (m/s)	c_3 (m)
450m	1	0.2	100	389.418
	2	0.4	100	717.356
	3	0.4	70	351.505
	4	0.4	85	518.290
	5	0.4	80	459.108
	6	0.4	78	436.439
	7	0.4	79	447.702
	8	0.4	79.5	453.387
	9	0.4	79.25	450.540

In the first simulation, the expert obtained c_3 = 389.418m that is short of the target. Since c_2 = 100m/s is already set to the maximum value, the expert increased c_1 from 0.2 to 0.4 in the second attempt. The second simulation result shows that c_3 = 717.356m that is too far away from the target (= 450m). Thus, this prompted the expert to reduce the velocity from 100m/s to 70m/s in the third run. The expert then adjusted c_2 accordingly in subsequent simulations and eventually concluded that the 9th run is a fairly good solution to the problem. In this case, it takes 9 runs for the expert to achieve the target constraint. This conclusion does not alleviate the possibility that there may be other combinations of c_1 and c_2 values that will produce nearer values to the target. However, it will take additional runs to prove it and there is no guarantee of a definitely better solution.

Comparatively, the CBE-Projectile system gives a quick solution and alleviates repetitive runs of the projectile model. Table 8-1 shows 6 nearest neighbours retrieved using the CBE-Projectile system.

Table 8-1. List of cases retrieved ordered by minimum distance for the gunner's problem

Case ID	$c_1 = \theta$	$c_2 = v$	$c_3 = x$	Distance
29	1	70	445.556	0.00451
51	0.4	80	459.108	0.00924
19	1.2	80	432.296	0.018
38	0.8	70	489.791	0.0403
56	0.2	100	389.418	0.0614
35	0.8	60	359.847	0.0914

First, the engineer does not need to run the model repeatedly by looking at the outputs (= c_3) and changing the inputs accordingly. Section 8.3.2.2 has shown that the CBE-Projectile system is able to adapt the old solutions (i.e., Case 29 and Case 38) to give a better prediction.

Table 8-3. Interpolated value and projectile model solution obtained from a conforming set: Case 29 and Case 38

Conforming set			
Case Id	$c_1 = \theta$	$c_2 = v$	$c_3 = x$
29	1	70	445.556
38	0.8	70	489.791
Interpolated value			
-	0.9975	70	446.103
Model solution			
-	0.9975	70	446.559

Now compare the CBE-Projectile's prediction and the human expert's result, for the gunner's problem (see Table 8-5), we note that the human expert's result is much better than the prediction from the CBE-Projectile system. But, don't forget that it takes the expert to run the projectile model 9 times to achieve this.

Table 8-5. CBE-Projectile's model solution and human expert's solution for Test Case I

CBE-Projectile's Model solution		
c_1	**c_2 (m/s)**	**c_3 (m)**
0.9975	70	446.559
Human expert's solution		
0.4	79.25	450.540

Now if we try to do the same with the CBE-Projectile system to see if we can obtain a better result, it would only be possible by the gunner's intervention. Based on the result obtained by the CBE-system (in Table 8-5), i.e., the adaptation result from the interpolation of Case 29 and 38, we adjusted c_1 hoping to get a better result. It took two additional simulations (First $c_1 = 0.995$, Second $c_1 = 0.99$) to achieve the following results:

Table 8-6. (CBE-Projectile model solution + Human Intervention) and human expert's solution for Test Case I

CBE-Projectile's Model solution + Human Intervention		
c_1	**c_2 (m/s)**	**c_3 (m)**
0.99	70	449.545
Human expert's solution		
0.4	79.25	450.540

The question we want to ask is: "Has the CBE-Projectile system failed?" The analysis is this: the CBE-Projectile system has three advantages over the human expert's approach: First, it gives quick solution. Second, the gunner does not need to start from scratch. Using the CBE-Projectile system, the gunner can begin the search from the nearest neighbour (Case 29) as the starting point to search for a better solution.

Third, the results in Table 8-6 shows that in fact Case 51 and Case 29 are good starting points to search for an optimum solution. The CBE-Projectile system can provide useful advice that has come close to solving the gunner's problem. The astute reader would find that the human's expert solution is very close to Case 51. The only difference between the two is 0.75 m/s for c_2.

In addition, we also see that such useful advice may also help the gunner to explore alternatives solution. For example, if a case with velocity c_1 = 70 m/s is not available in the case base, the gunner can consider to use a case with c_1 = 80 m/s and c_2 = 0.4 instead, to get similar c_3

8.5 DISCUSSION ON THE EXPERIMENTAL RESULTS

From the above experimental results, the CBE-Projectile system is shown to have the following advantages:

- The CBE-Projectile system provides fast response time and gives quick solution.
- It provides useful information that points engineers to the right direction in searching for an optimum solution.
- When adaptation is required, engineers can rely on the CBE-Projectile system to produce an improved solution and fine-tune the result with the use of the projectile model.
- This is only a simple system. We can expect that the CBE systems in connection with more complex models are even more effective and efficient.

8.6 CONCLUDING REMARKS

In this chapter, we have described an application of CBE, CBE-Projectile that has been developed using the CBE architecture for the collaboration of a CBR system and a projectile model, so as to solve the gunner's problem. We have first demonstrated that GSNN paired with conformable candidate set for interpolation can benefit to produce a better prediction. The merit of the CBE-Projectile system has then been further demonstrated through its capability of producing quick solutions and providing useful information to guide engineers to search for an optimum solution that otherwise could take up a great amount of time.

Although there is no guarantee that the system always produces an exact solution [Richter, 2001] or a better solution than a human expert who uses a 'trial and error' approach, it is true that the system can be a more efficient means of searching solutions for inverse and *constraint problems. In circumstances where the CBE-Projectile system fails to produce an optimum solution, engineers can intervene. New cases can be added into the case base to improve the case base performance. Indeed, this is the beauty of the CBE-architecture that it allows the collaboration of a CBR system, a numerical model and engineers to solve design problems.*

Chapter 9

Case Study II: CBE-Conveyor

9.1 INTRODUCTION

The pneumatic conveyor problem has been discussed in Section 2.3 as one of the showcases of the motivation of this research. In this example, one of the engineering problems faced by engineers is to predict what inputs they should use for a given distribution of particles sizes at the receiver. Design engineers of a pneumatic conveyor are often concerned with the weight fraction of the maximum particle size class and the weight fraction of the minimum particle size class at the receiver (E.g., Section 2.3.1). In this chapter we examine the conveyor model in the context provided by Chapelle *at el.* (2004). The simulation model is a combination of a degradation model, based on the experimental determination of breakage matrices, and a physical model of solids and gas flow in the pipeline. A pneumatic conveyor may consist of at least two straight pipes and at least one bend. The model is useful for prediction of particle size distributions at the receiver of various kinds of materials such as sugar, tea leaf, rice, and etc. Input parameters of the model include the velocity of air, suspension density, number of bends, number of straight pipes, the type of each bend , the angle of each bend, the diameter of the bend, diameters of the straight pipes, and breakage matrices of particle. For this research, a single bend conveyor line is used to examine the feasibility of using CBR in such an application domain.

In this chapter, we study how the CBE (Case-Based Engineering) architecture (see Section 2.5.1) can be implemented to solve such an engineering problem. We termed the completed practical development of the CBE system as CBE-Conveyor. The system consists of the engineer, a CBR model and the pneumatic conveyor model. In the CBE-Conveyor system, the engineer can interact with the CBR model and the conveyor model to search for a solution for inverse and constraint problems.

In Section 9.2 we give an overview of the pneumatic conveyor model and the assumptions that have been made. In Section 9.3 we present the findings of the CBE-Conveyor system. In addition, we also discuss solutions for this particular engineering application to the questions raised in Section 2.5.2. In Section 9.4 we present a variety of examples to illustrate how the CBE-Conveyor system solves a derived constraint problem and compare it with the trial and error approach normally taken by a human expert. Results show that the CBE-Conveyor system is more efficient than the human expert in using the conveyor model directly to search for a solution. In Section 9.5, we discuss the advantages of the CBE-Conveyor over traditional approaches of solving inverse and constraint problems in this pneumatic conveyor application. We conclude in Section 9.6, with a summary of the chapter.

9.2 THE CONVEYOR MODEL

The pneumatic conveyor design problem is part of the Quality in Particulate Manufacturing (QPM) initiative funded by the UK EPSRC Innovative Manufacturing Initiative for Process Industries. In this problem, there are three input parameters: velocity of air, bend type and bend angle; the output is the particle size distribution at the outlet (see Fig. 2-2 in Section 2.3). The engineer's task is to determine suitable input parameters so that there will not be too much dust formed particles. The present model [Chapelle *et al.*, 2004] considers only the damage caused by the collisions of particles at the conveyor bends and assumes the following:

- All particles travel at the same speed regardless of their size and density.
- Inter-particle collisions are not considered.
- Damage of a particle is represented by a single impact. Additional impacts in the same bend are not considered.
- Each impact of a particle is considered independent of the next; fatigue phenomena (i.e., a particle breaks after a number of collisions) is ignored.

The main factors of the conveyor model are described below:

- **The air velocity factor.** By increasing the air velocity, the particle conveying velocity will increase. Consequently, the attrition rate increases accordingly because the particle velocity before collision defines the momentum that turns into impact load during the collision. Impact loading is the force (strength) to which the particle is subjected during the collision and that will cause its degradation. It is directly related to the particle velocity before the impact.

- **The bend angle factor.** The collision angle (i.e., the angle between the particle path and the tangent to the wall at the collision point) greatly affects the impact loading during the collision. The collision angle is at maximum (i.e., 90 degree) when the particle collides in perpendicular with the wall. The angle of collision decreases as the radius of a bend increases. The lower the bend angle, the lower the attrition rate.
- **The bend type factor.** Each bend type has different collision point for the same bend angle. There are four bend types in this model: long radius, short radius, blinded tee and turbulence drum. The amount of degradation depends on the bend type used.

9.3 DEVELOPMENT OF THE CBE-CONVEYOR SYSTEM

In the CBE-Conveyor system, cases are represented by the predicate in a way conducive to the use of the generalized nearest neighbour retrieval(GNNR) approach(see Section 4.2): $c(c_1, c_2, \ldots, c_{15}) = (Out\_1, \ldots, Out\_6; VairIn, BendType, BendAngle, In\_1, \ldots, In\_6)$ where *Out_i* means the weight fraction of particles at the outlet in size range *i*, *In_i* means the fraction of particles at the inlet in size range *i*, where i=1, 2, …, 6. The size class ranges from 1 to 6 (see Table 9-1). *VairIn* is the air velocity. *BendType* and *BendAngle* are nominal values. For convenience of exposition, we leave out the values of *In_1*, …, *In_6*, which are kept constant in the following examples (see Table 9-2). One hundred experimental cases have been generated from the conveyor model for degradation of granulated sugar in a single bend pneumatic conveyor line. In what follows, we show approaches to the problems outlined in Section 2.5.2 in the development of the CBE-Conveyor system.

Table 9-1. Particles size class

Class	**1**	**2**	**3**	**4**	**5**	**6**
UpperBound (μm)	850	600	500	425	355	250
LowerBound (μm)	600	500	425	355	250	0

Table 9-2. Initial input parameters used for the experimental setup

Class range for particles at the Inlet	In_1	In_2	In_3	In_4	In_5	In_6
Weight Fraction (%)	29	27	15	18	9	2

9.3.1 Metrics Defined on the Unified Query Space

The similarity between a target case and cases in the case base is computed using the standard weighted sum method [Kolodner, 1993]. Nearest neighbours to a target case is approximated using the Generalized Nearest Neighbor Retrieval (GNNR) method over the unified problem:solution (see Chapter 4) space. For example, let R be the target region by n constraints, where n is an integer. We take R to be the space of points $r(r_1,\dots, r_n)$, and the distance metric, $d(c,r)$ is given as follow:

$$d(c,r) = [(w_1 |c_1 - r_1|^2 + w_2 |c_2 - r_2|^2 + \dots + w_n |c_n - r_n|^2))/ \sum_{i=1,\dots,n} w_i]^{1/2}.$$

w_i is the weight assigned for i^{th} attribute. For more information about the metrics on the unified space please refer to Section 4.2. For continuous domains such as the air velocity, the distance between two points for each attribute is normalized by the range value. For the nominal domains such as the bend type and the bend angle, the distance metric is provided by a human expert, experienced in the construction and use of the various bend types. Table 9-3 gives a list of the domain type for each attribute.

Table 9-3. Attribute analysis for the pneumatic conveyor problem

Attribute	Type	Range
In_1,..., In_6	Continuous	[0, 100] %
Out_1,...,Out_6	Continuous	[0, 100] %
VairIn	Continuous	7 - 21 (m/s)
BendType	Nominal	{LongRadius, ShortRadius, BlindTee, TurbulenceDrum}
BendAngle	Nominal	{30, 45, 65, 70, 80, 90}

9.3.2 Constraints and Interpolation: The Inverse Problem

Consider target outputs are *Out_1* = 24 and *Out_6* = 3.85. The distance between a target and each case in the case base is calculated based on the constraints defined on *Out_1* and *Out_6*. In this case, *R* is the target region where r_1: *Out_1 =24*, r_2: *Out_6 = 3.85*. Table 9-4 shows two cases selected for interpolation to find the bend type and the bend angle for the given target.

Table 9-4. Two cases selected for interpolation for target: *Out_1* = 24 and *Out_6* = 3.85

Two cases selected for interpolation for target: *Out_1* = 24 and *Out_6* = 3.85						
Distance	**Case Id**	**Out_1**	**Out_6**	**VairIn**	**BendType**	**BendAngle**
0.042	12	24.54	3.71	14.28	BlindedTee	65 deg
0.0481	65	23.77	4.1	15.96	LongRadius	65 deg
Interpolated value						
0.01819	-	24.2	3.88	15	ShortRadius	65 deg
Model solution						
0.01848	-	24.25	3.87	15	ShortRadius	65 deg

Table 9-4 shows the interpolated solution given by GSNN. Notice that the interpolated value for the nominal bend types BlindedTee and LongRadius is ShortRadius, which is neither of those in the interpolation set. Finally the modelled case confirms the accuracy of the interpolation. The modelled case gives a distance of 0.01848, which is far better than the two retrieved cases (note that the distance for Case 12 is 0.042 and for Case 65 is 0.0481).

9.3.3 Constraints and Interpolation: Nominal values

The inverse problem in Section 9.3.2 shows an example of interpolation over nominal values where the interpolated bend type is distinct from those in the interpolation set. Now we consider an example of a nominal constraint. We use the same example as above, but this time we add a constraint that only bends types: LongRadius, BlindedTee and TurbulenceDrum are available. Since the BlindedTee and LongRadius bends are allowable bend types, the retrieved cases for the above example remains unchanged:

Table 9-5. Two cases selected for interpolation for target: *Out_1* = 24 and *Out_6*= 3.85, with constrained bend types

Two cases selected for interpolation for target: *Out_1* = 24 and *Out_6* = 3.85, where bendtypes are constrained to {LongRadius, BlindedTee, TurbulenceDrum }						
Distance	**Case Id**	**Out_1**	**Out_6**	**VairIn**	**BendType**	**BendAngle**
0.042	12	24.54	3.71	14.28	BlindedTee	65 deg
0.0481	65	23.77	4.1	15.96	LongRadius	65 deg
Interpolated value, where bendtypes are constrained to {LongRadius, BlindedTee, TurbulenceDrum}						
0.01819	-	24.2	3.88	15	BlindedTee	65 deg
Model solution						
0.02160	-	24.22	3.89	15	BlindedTee	65 deg

In this case, GSNN has constrained the search to the available bend types. It should be noticed that the solution is further from the target than that for the unconstrained search. The distances between the target and the model solutions is 0.02160 for constrained bend types and is 0.01848 for unconstrained bend types (see Table 9-5). Astute readers may notice that the distance between the target and the interpolated value for both constrained and unconstrained bend type examples are the same (i.e., 0.01819 in Table 9-4) but the bend type is different from one another. This is because the distance is calculated based on the constraints defined on the *Out_1* and *Out_6*.

This example shows that the engineer can define constraint over nominal values. In addition to this, the CBE-Conveyor system allows the engineer to restrict the search and the adaptation result to only available bend types.

9.3.4 Constraints and Interpolation: Multi-valued Case Mappings

Here we show an example of a multi-valued mapping. In this example, the engineer wants to achieve an output where the fraction of the largest particles is 6 times the fraction of the smallest particles. The derived attribute *Out_1/Out_6* is accordingly added to the cases, and we search for cases close on this attribute. This is a good example of a multi-valued mapping, since there can be a number of cases with a wide diversity of inputs which can produce this output ratio. But, as is shown here, it is not acceptable to interpolate from these at random. First, we must check for conformability.

Table 9-6. Retrieval set for target: *Out_1/Out_6* = 6

Distance	CaseId	Out_1 / Out_6	VairIn	BendType	BendAngle
0.1063	98	6.1063	19.49	LongRadius	45 deg
0.1328	80	6.1328	12.52	TurbulenceDrum	90 deg
0.1769	67	6.1769	14.28	TurbulenceDrum	70 deg
0.1951	116	5.8049	14.28	LongRadius	80 deg
0.2024	103	5.7976	15.96	LongRadius	65 deg
0.2079	38	6.2079	12.52	BlindedTee	90 deg
0.2451	158	5.7549	14.28	ShortRadius	80 deg

Table 9-6 shows the result of a query subject to a constraint: Out_1/Out_6 = 6. As we see, while the first three cases are close to the target Out_1 / Out_6, they are not all close in the unified space, including *VairIn*, *BendType* and *BendAngle*. Case 98 and 80 are non-conformable for interpolation because their inputs are quite far apart. Case 80 and 67 are conformable for interpolation because their inputs are relatively closer than Case 98 and Case 80. To examine this conjecture we compare the accuracy of the non-conforming set prediction against the prediction of the conforming set.

9.3.4.1 Interpolation over a Non-Conforming Set

In Table 9-7, we show the interpolated result based on these two cases. We see that the modeled result shows the purpose of the interpolation has failed to achieve (i.e., to find a better solution with *Out_1/Out_6* closer to 6).

Table 9-7. Interpolated value and conveyor model solution obtained from a non-conforming set: cases 98 and 80

Non-conforming set				
Case Id	**Out_1/Out_6**	**VairIn**	**BendType**	**BendAngle**
98	6.1063	19.49	LongRadius	45 deg
80	6.1328	12.52	Turbulencedrum	90 deg
Interpolated value				
-	6.2328125	12.52	LongRadius	45 deg
Model solution				
-	10.17	12.52	LongRadius	45 deg

9.3.4.2 Interpolation over a Conforming Set

In Table 9-8, we show that the interpolation based on cases 80 and 67 is far better. Because these two cases are close in their inputs, they give a much better interpolation result. This verifies the approach to conformability that we have adopted in this project.

Table 9-8. Interpolated value and conveyor model solution obtained from conforming set: cases 80, 67

Non-conforming set				
Case Id	**Out_1/Out_6**	**VairIn**	**BendType**	**BendAngle**
80	6.1328	12.52	TurbulenceDrum	90 deg
67	6.1769	14.28	TurbulenceDrum	70 deg
Interpolated value				
-	6.0328125	12.83	TurbulenceDrum	80 deg
Model solution				
-	6.0431	12.83	TurbulenceDrum	80 deg

9.4 CBE-CONVEYOR VERSUS HUMAN EXPERT

In this section, we measure the performance of the CBE-Conveyor system versus a human expert in solving inverse problems and constraint problems in the pneumatic conveyor domain. The human expert has been working on the conveyor model for approximately one and a half years and acquainted with how the input parameters will affect the distribution of the particles size class at the outlet. In these experiments, we assume there are two outputs and three inputs being *Out_1*, *Out_6*, *VairIn*, *BendAngle*, *BendType*. The following two test cases presented demonstrate a typical approach used by the expert to solve inverse problems. However, for fair assessment, the expert solved these two cases separately. The results obtained from the first test case do not influence the second test case.

9.4.1 Test Case I

Consider the target constraints:

Out_1 = 24 and *Out_6* = 3.85

Table 9-9 shows attempts made by the human expert to achieve the target outputs. The expert first ran a simulation (i.e., *i*=1) assuming that *VairIn* = 9, *BendType* = 90 deg, *BendType* = BlindedTee. The expert then adjusted input parameters accordingly by looking at the result in column 7 and 8.

Table 9-9. Results of iterative runs by expert to solve Test Case I

Target: Out_1	***Target: Out_6***	***i=No. of simulation***	***VairIn***	***BendType***	***BendType***	***Out_1***	***Out_6***
24	3.85	1	9	90 deg	BlindedTee	23.7	3.22
		2	14	90 deg	BlindedTee	23.03	4.24
		3	9	70 deg	BlindedTee	25.01	2.97
		4	11	90 deg	BlindedTee	23.80	3.52
		5	13	90 deg	BlindedTee	23.42	3.95

In each of the first three runs the expert tried different input values to see if it produces the desired output. These attempts are not successful however it suggests that the right *VairIn* value (i.e., column 4) might lie between 9 and 14, for a 90 degree *BlindedTee* bend. The expert then tried *VairIn = 11* (4th run) *and VairIn = 13* (5th run). Eventually, the expert concluded that the 4th run is a reasonably accurate solution to the problem. Presumably, there might be other possible solutions but it will take additional runs to prove it. However there is no guarantee of a definitely better solution.

Comparatively, the CBE approach gives quick results and alleviates repetitive running of the conveyor model. Table 9-10 shows 5 nearest neighbours retrieved using the CBE-Conveyor. Apparently, the CBR approach is straight forward and gives quick solution to the problem.

Table 9-10. Results obtained by CBR to solve Test Case I

5 nearest neighbours retrieved using CBE-Conveyor						
Distance	**Case Id**	**Out_1**	**Out_6**	**VairIn**	**BendType**	**BendAngle**
0.042	12	24.54	3.71	14.28	BlindedTee	65 deg
0.0481	65	23.77	4.1	15.96	LongRadius	65 deg
0.0598	21	23.45	4.1	14.28	BlindedTee	80 deg
0.066	39	23.67	4.19	15.96	TurbulenceDrum	65 deg
0.0673	49	23.81	3.47	10.76	TurbulenceDrum	90 deg

Firstly, the engineer does not need to run the model repeatedly, looking at the outputs and changing the inputs accordingly. Section 9.3.2 shows that the CBE-Conveyor system is able to adapt the old solutions (i.e., Case 12 and Case 65) to get a better solution. The model solution based on the interpolated value can be found in Table 9-5. The following shows a comparison of the values given by the CBE-Conveyor system and the human expert for target case, *Out_1* = 24 and *Out_6* = 3.85 :

Model solution				
Out_1	**Out_6**	**VairIn**	**BendType**	**BendAngle**
24.25	3.87	15	ShortRadius	65 deg
Human expert's solution				
23.80	3.52	11	BlindedTee	90 deg

Secondly, we note that the CBE-Conveyor system retrieves a diverse set of solutions which could help engineers to explore other alternatives to solve the problem. For instance, in this case Table 9-10 shows that Case 65 is an alternative of Case 12 that gives similar output. In this case, the engineer can substitute a BlindedTee bend with LongRadius bend. Besides, there are other combinations of input parameters that give similar outputs.

9.4.2 Test Case II

Consider the target constraint:

$Out\_1 / Out\_6 = 6$

Table 9-11 shows an example of constraints defined over the output space in arithmetic expression where $Out\_1 / Out\_6 = 6$. Without any prior knowledge of the result obtained from test case I, the expert first ran a simulation (i.e., i=1) assuming that *VairIn* = 9, *BendType* = 90 deg, *BendType* = BlindedTee. The expert then adjusted input parameters accordingly by looking at the result: Out_1 / Out_6. In this case, the expert had to calculate the output derived and run the models repeatedly to see if it satisfies the constraint.

We note that this time, the expert got better result in each additional simulation performed. For instance, the first simulation gave 7.36 and it gets improved gradually in the following runs. The third simulation gave 5.432. These attempts suggested that the right *VairIn* value could lie between 11 and 14, for a 90 degree *BlindedTee* bend. The expert then tried *VairIn = 12* (4th run) *and VairIn = 12.5* (5th run). Eventually, the expert concluded that the 5th run is a quite good solution to the problem. Presumably, the expert can perform additional simulations to get better results. However, if we adopt the same approach to find out alternatives for bend angles or bend types, this could require considerable extra amount of time and effort. This situation could be worsening for applications that involve high input dimensions.

Table 9-11. Results of iterative runs by an expert to solve Test Case II

Target: Out_1 / Out_6	***i=No. of simulation***	***VairIn***	***BendType***	***BendType***	***Out_1 / Out_6***
6	1	9	90 deg	BlindedTee	7.360
	2	11	90 deg	BlindedTee	6.761
	3	14	90 deg	BlindedTee	5.432
	4	12	90 deg	BlindedTee	6.363
	5	12.5	90 deg	BlindedTee	6.133

Once again, Table 9-12 shows that the CBE-Conveyor system is more efficient in prediction. The system is able to retrieve a set of diverse solutions which might take a considerable time for an engineer to achieve using a numerical model directly.

Table 9-12. Results obtained by CBR to solve Test Case II

5 nearest neighbours retrieved using CBE-Conveyor					
Distance	**Case Id**	**Out_1 Out_6**	**VairIn**	**BendType**	**BendAngle**
0.1063	98	6.1063	19.49	LongRadius	45 deg
0.1328	80	6.1328	12.52	TurbulenceDrum	90 deg
0.1769	67	6.1769	14.28	TurbulenceDrum	70 deg
0.1951	116	5.8049	14.28	LongRadius	80 deg
0.2024	103	5.7976	15.96	LongRadius	65 deg

Although, the best match Case 98 gives 6.1063 (see Table 9-12) which has little difference to the solution given by the expert in the 5^{Th} simulation (i.e, 6.133), we can rely on the CBE model to give an improved prediction if the retrieved cases require adaptation to give better results. For this we can refer to the solution obtained in Section 9.3.4.2 (Target: *Out_1/Out_6 =6*). Table 9-8 shows that the CBE can give a more accurate prediction such that *Out_1/Out_6* = 6.0431. The following shows a comparison of the values given by the CBE-Conveyor system and the human expert for the target case, *Out_1* / *Out_6* = 6 :

Model solution			
Out_1 / Out_6	**VairIn**	**BendType**	**BendAngle**
6.0431	12.83	TurbulenceDrum	80 deg
Human expert's solution			
6.133	12.5	BlindedTee	90 deg

This example shows that there may be a number of solutions for a target problem. It is often difficult for engineers to find out all the possible solutions using the trial and error method. Moreover, it is increasingly difficult when a derived attribute constraint involves complex numerical calculations. The expert will require a way to calculate the derived attribute. The CBE-Conveyor system however is more efficient to solve such design problems alleviating the need of reconstructing the conveyor model to accommodate such constraints.

9.5 DISCUSSION ON THE EXPERIMENTAL RESULTS

We have carried out tests of the conveyor problem with different target constraints on the CBR model and compared with the results obtained by an expert [Chapelle *et al.*, 2004]. The results in Section 9.3 and 9.4 show that the CBR model is capable of producing fairly accurate solutions in an efficient and effective way. The CBE-Conveyor system has the following advantages:

- It allows flexible query mode. Engineers can query the model in any direction.
- Engineers can specify constraints dynamically to query the case base without bothering to reconstruct the conveyor model.
- The system is able to handle nominal constraints such as bend types and bend angles. With GSNN, it can predict nominal values not present in the retrieval set.
- It can provide alternatives for a given target problem. Engineers can explore alternative input parameters that produce similar results.
- When a target problem is presented, the CBE-Conveyor system provides fast response time and gives quick solutions, which might require considerable amount of time and effort for engineers to solve the problem.
- When necessary, adaptation can be performed to improve the prediction accuracy. In this case, we suggest using GSNN together with conformable interpolation candidate sets to perform the adaptation task.

9.6 CONCLUDING REMARKS

In this chapter we have described the CBE-Conveyor system, which shows the development of this research project progressed into a realization of a practical application. The CBE-Conveyor system has been developed using the CBE architecture with the collaboration of a CBR system and a numerical model, so as to increase the usability of the numerical model. It has been shown that the system allows the engineer to define constraints flexibly. These constraints can be nominal or continuous, and can be over any selection of inputs and outputs to the numerical model, and over derived attributes necessary for the constraints.

When necessary, The CBE-Conveyor system is capable of producing better solutions by adaptation. In the adaptation phase, it is important to ensure that cases selected for adaptation are conformable. The concept of conformability for interpolation is defined in terms of a metric over the whole input-output space. For conformable retrieval sets, it has been shown that the interpolative method GSNN can successfully interpolate over all nominal and continuous constraints, and can produce better solutions over nominal values.

The examples in Section 9.3 and 9.4 have demonstrated the capability of the CBE-Conveyor system. The system has been tested in practice for its capability, and has been shown to produce answers immediately to several real design problems which would otherwise have required considerable engineering expertise and time.

PART IV

CONCLUSION AND FUTURE WORK

Chapter 10

Conclusion and Future Work

Chapter 10

Conclusion and Future Work

10.1 CONTRIBUTION TO KNOWLEDGE

In this thesis there are several original contributions to knowledge in the fields of CBR. These are:

1. A Case Based Engineering architecture for collaboration between: Engineer, Numerical model and CBR system.
2. A CBR framework allowing constraints to be defined over unified problem:solution space.
3. A novel method for interpolation over nominal values termed as Generalized Shepard Nearest Neighbour (GSNN) method.
4. The use of GSNN in conjunction with the diversity algorithm – an approach, which is capable of improving predictive performance.
5. A new case reduction method using the GSNN error function as an error measure in case selection.

10.2 CONCLUSION OF THE THESIS

In this thesis, we set out to show whether the usability of numerical models could be improved by using a case based reasoning system. The thesis has shown that this idea works in many circumstances. We have provided a suitable architecture for the integration of these two systems, (CBE), and have given two good practical examples. These show clearly the advantages of the architecture, and also highlight how usability is improved.

A number of secondary problems have arisen in the course of the study, and these have been successfully solved, so that the problems do not constrain the domain of applicability of the technique unduly. The solutions to the problems inherent in the discipline of Case Based Engineering may all be seen as consequential to one central idea: that we should regard the case base holistically as a unified problem + solution space, and define a similarity metric over the whole unified space. If the metric is defined as the weighted sum of similarity over attributes, then the attribute set should cover the whole unified space, and should include all attributes—continuous and nominal.

The examples given in the thesis demonstrate that usability of numerical models can be improved with the CBE system in some cases. However we cannot exactly define the application scope of this technique, in terms of which models it can be used with. There is no obvious prescription of the scope, but in order to give a guide as to applicability, we conducted a brief survey of models used at the University of Greenwich. A summary of the results of this survey is given in Table 10-1.

Table 10-1. Some models studied for applicability of CBE

Model	CBE Applicable?	Comments
Pollution in Street Canyons	Yes	Small number of parameters, as long as we restrict to limited street geometries. For details see Appendix C - C1.
Solder Joint Reliability	Yes	Small number of parameters. See Appendix C - C2.
Smartfire	No	The input to this model is any building geometry, and any fire star position. These represent a vast number of parameters. [Galea, 1989; Stroup, 1995]
Smartfire for a single room	Yes	With very limited geometries, such as the standard Steckler room, we could parameterize the fire start position. The output would have to be maximum temperature at certain fixed points [Knight *et al.*, 1999; Taylor *et al.*, 2001].
Metal Casting	No	Again geometries are involved here, which represent too large a parameter set. However, once again, if we restrict the geometry to. e.g. rotationally symmetric wheels and armatures, then a CBE system becomes viable (see E.g. [Knight *et al.*, 2000])
Metal cutter	Perhaps	Inputs to this model are thickness of the metal sheet, the material, the temperature profile of a cutting blade. Outputs are the temperatures at measurable points on the metal sheet. The CBE system will be used to derive the temperature profile of the blade from the temperature profiles at the measured points. In order to make the CBE system work, some simple parameterization of the temperature profiles would be needed. (for a description of this model see e.g. [Palansuriya, 2000]) As a matter of interest, this IP has already been solved by domain decomposition. In this work, the profiles were also simply parameterized [Palansuriya, 2000].

This summary shows that the main determining factor in deciding whether to try CBE is the dimensionality of the input-output space. In many cases, the attributes in this space may be continuous curves (as in the metal cutter problem), or physical geometries (as in the fire modeling system and the metal casting models). Also, sometimes the output is required in graphical form (perhaps a display of the fire itself progressing). In these cases, the only way to utilize CBE is to reduce the dimensionality somehow. This might be done by parameterizing the continuous curves somehow, or by severely limiting geometrical information to a few standard types.

10.3 FUTURE WORK

It is apparent from the applicability study described above that there is work to do in trying the architecture on other practical problems. The metal cutter might be a good starting point for this further study, since there is a good practical reason to try to solve the inverse problem. A control mechanism for the cutter speed depends upon the inverse problem solution, and overheating of the blade is a major cause of wear.

In addition to the main question examined, there is also more work indicated for two of the new ideas in the thesis: the nominal value interpolation and its incorporation in a new reduction technique. The scope of the thesis has not allowed a complete comparison of either of these techniques. Future work would be needed to ascertain how GSNN compared with other nominal value methods on a wider variety of examples from the machine learning community. This exercise will be fairly time consuming in view of the necessity to construct alternative metrics on the sets of nominal values. These metrics can themselves be constructed in a variety of ways. In addition, the GSNN reduction technique would also benefit from a thorough comparative evaluation on a wider set of cases.

PART V

REFERENCES AND APPENDICES

References

Appendix A

List of Figures and Tables

Appendix B

List of Publications and Selected Papers

Appendix C

Survey Scope of the CBE Architecture

Appendix D

Screenshots of the CBE Model

References

1. [Agarwal *et al.*, 1985] Agarwal, V. K., Mills, D. and Mason, J.D. (1985) The Best of Bulk Solids Handling, Pneumatic Conveying of Bulk Powders, vol. D/86, Trans Tech Publications, Clausthal-Zellerfeld, Germany, pp.111-116

2. [Aha, 1992] Aha, D. W. (1992) Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. International Journal of Man-Machine Studies, 36, pp. 267-287

3. [Aha *et al.*, 1991] Aha, D. W., Kibler, D. and Albert, M. K. (1991) Instance-Based Learning Algorithms. Machine Learning, 6, pp.37-66

4. [Banks *et al.*, 2002] Banks, H. T., Michele, L. J., Wincheski, B. and Winfree, W. P. (2002) Real Time Computational Algorithms for Eddy-Current-Based Damage Detection. Inverse Problems, 18 (3) pp.795-823

5. [Barrett *et al.*, 1994] Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C. and Van der Vorst, H. (1994) Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. 2nd ed., Philadelphia, PA, SIAM.

6. [Beck and Murio, 1986] Beck, J. V. and Murio, D. (1986) Combined Function Specification-Regularization Procedure for Solution of Inverse Heat Conduction Problem. AIAA Journal, 24 (1) pp.180-185

7. [Bell *et al.*, 1996] Bell, T. A., Boxman, A., Jacobs and J. B. (1996) Attrition of Salt during Pneumatic Conveying, Proceedings of The 5th World Congress of Chemical Engineering, San Diego, USA vol. V, pp. 238-243

8. [Bergmann, *et al.*, 2001] Bergmann, R., Richter, M. M., Schmitt, S., Stahl, A. and Vollrath, I. (2001) Utility-oriented Matching: A New Research Direction for Case-Based Reasoning. In: Proceedings of the 9th German Workshop on Case-Based Reasoning, GWCBR'01, Baden-Baden, 14.-16. März 2001.

9. [Bertino *et al.*, 2002] Bertino, L., Evensen, G., and Wackernagel, H. (2002) Comibing Geostatistics and Kalman Filtering for Data Assimilation in an Estuarine System. Inverse Problems, 18 (1) pp.1-23

10. [Breiman, 1984] Breiman, L. (1984) Classification and Regression Trees. Belmont, CA, Wadsworth International Group.

11. [Bridge, 2001] Bridge, D. (2001) Product Recommendation Systems: A New Direction. In: Aha, D. and Watson, I. ed. Proceedings of Workshop on CBR in Electronic Commerce at the International Conference on Case-Based Reasoning, ICCBR-01, Vancouver, Canada.

12. [Byrne, 2003] Byrne, C. (2003) A Unified Treatment of Some Iterative Algorithms in Signal Processing and Image Reconstruction. Inverse Problems, 20 (1) pp.103-120

13. [Cameron-Jones, 1995] Cameron-Jones, R. M. (1995) Instance Selection by Encoding Length Heuristic with Random Mutation Hill Climbing. In: Proceedings of the Eighth Australian Joint Conference on Artificial Intelligence, pp.99-106

14. [Chapelle *et al.*, 2004] Chapelle, P., Christakis, N., Abou-Chakra, H., Bridle, I., Bradley, M.S.A., Patel, M., and Cross, M. (2004) Computational Model for Prediction of Particle Degradation during Dilute Phase Pneumatic Conveying: Modelling of Dilute Phase Pneumatic Conveying, Advanced Powder Technology, Vol 15 (1) pp.31-49

15. [Chatterjee and Campbell, 1993] Chatterjee, N. and Campbell, J. A. (1993) Adaptation through Interpolation for Time Critical Case-Based Reasoning. In: Proceedings of 1st European Workshop, EWCBR-93, Kaiserslautern, Germany, November, Lecture Notes in Artificial Intelligence, Vol. 837. Berlin, Springer-Verlag, pp.221-233

16. [Chatterjee and Campbell, 1999] Chatterjee, N. and Campbell, J. A. (1999) Interpolation of Plans for Time-Critical Adaptation. Knowledge-based Systems, 12 (4) pp.171-182

17. [Cheetham, 2001] Cheetham, W. (2001) Benefits of Case-Based Reasoning in Color Matching. In: Proceedings of the 4th International Conference on Case-Based Reasoning, ICCBR-01, Vancouver, BC, Canada, pp.589-596

18. [Cheetham and Graf, 1997] Cheetham, W. and Graf, J. (1997) Case-Based Reasoning in Color Matching. In: Proceedings of the 2nd International Conference on Case-Based Reasoning, ICCBR-97, RI, USA, pp. 1-12

19. [Chow et al., 1999] Chow, P. L., Ibragimov, I. A. and Khasminskii, R. Z. (1999) Statistical Approach to Some Ill-Posed Problems for Linear Partial Differential Equations. Probability theory and related fields, 113, pp.421-441

20. [Converse *et al.*, 1989] Converse, T., Hammond, K. and Marks, M. (1989) Learning Modification Rules from Expectation Failure. In: Proc. of 2nd Case-based Reasoning workshop Darpa, Pensacola Beach, FL, 23-26 May, pp.110-114

21. [Cost and Salzberg, 1993] Cost, S., Salzberg, S. (1993) A Weighted Nearest Neighbour Algorithm for Learning with Symbolic Features. Machine Learning, Vol. 10, pp. 57-58.

22. [Cover and Hart, 1967] Cover, T. M. and Hart, P. (1967) Nearest Neighbour Pattern Classification. IEEE Transactions on Information Theory, 13, pp.21-27

23. [D'Souza, 1975] D'Souza, N. (1975) Numerical Solution of One-Dimensional Inverse Transient Heat Conduction by Finite Difference Method. A.S.M.E., Paper No. 68-WA/HT-81.

24. [Evans and Stark, 2002] Evans, S. N. and Stark, P. B. (2002) Inverse Problems as Statistics. Inverse Problems, 18 (4) pp.R55-97

25. [Finn and Cunningham, 1998] Finn, D. P. and Cunningham P. (1998) Solution reuse for model generation in numerical simulation. Journal of Artificial Intelligence in Engineering, 12 (3) pp.297-314

26. [Fisher, 1936] Fisher, R. A. (1936) The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics, 7, Part II, pp.179-188

27. [Frank, 1963] Frank, I. (1963) An Application of Least Squares method to the solution of the inverse problem of heat conduction. Heat Transfer, 85C pp.378-379

28. [Franke and Nielson, 1980] Franke, R. and Nielson, G. (1980) Smooth Interpolation of Large Sets of Scattered Data. International Journal for Numerical Methods in Engineering, 15, pp.1691-1704

29. [Galea, 1989] Galea, E.R. (1989) On the field modelling approach to the simulation of enclosure fires. J. Fire Protection Engineering, Vol. 1, pp.11-22

30. [Gates, 1972] Gates, G. W. (1972) The Reduced Nearest Neighbour Rule. IEEE Transactions on Information Theory, 18 (3) pp.431-433

31. [Hammond, 1987] Hammond, K. J. (1987) Explaining and Repairing Plans that Fail. In: Proceedings of 10th International Joint Conference on Artificial Intelligence IJCAI-87, Milan, Italy, 23-28 August. San Mateo, CA, Morgan Kaufmann, pp.109-114

32. [Hammond, 1990] Hammond, K. J. (1990) Explaining and Repairing Plans that Fail. Artificial Intelligence, 45 (1-2) pp.173-228

33. [Hammond *et al.*, 1988] Hammond, K., Converse, T. and Marks, M. (1988) Learning from Opportunities, Storing and Re-Using Execution Time Optimizations. Proceedings of AAAI-88, St. Paul, MN, 21-26 August. Menlo Park, AAAI Press, pp.536-540

34. [Hammond *et al.*, 1993] Hammond, K., Converse, T., Marks, M. and Seifert, C. M. (1993) Opportunism And Learning. Machine Learning, 10 (3) pp.279-309

35. [Hart, 1968] Hart, P.E. (1968) The Condensed Nearest Neighbour Rule. IEEE Transactions on Information Theory, 14, pp.515-516

36. [Hilbert, 1984] Hilbert, J. D. (1984) The Best of Bulk Solids Handling, Pneumatic Conveying of Bulk Powders, Vol. D/86 Trans Tech Publications, Clausthal-Zellerfeld, Germany, pp.107-110

37. [Hinrichs and Kolodner, 1991] Hinrichs, T. R. and Kolodner, J. L. (1991) The Roles of Adaptation in Case-Based Design. *Proc. DARPA Case Based reasoning Workshop*, Washington D. C., May. CA, Morgan Kaufmann, pp.121-132

38. [Hwang, 1997] Hwang, J. S. (1997) Solder Technologies for Electronic Packaging. In: Harper, C. A. (ed.) Electronic Packaging & Interconnection Handbook, 2nd Edition, USA, McGraw-Hill, pp. 5.1 – 5.70

39. [Ji and McLaughlin, 2003] Ji, L. and McLaughlin, J. (2003) Recovery of the Lamé Parameter μ in Biological Tissues. Inverse Problems, 20 (1) pp.1-24

40. [Jia *et al.*, 2003] Jia, F., Patel, M. K. and Galea, E. R. (2003) Simulations to Fire Propagation within an Airplane Cockpit. to appear in InterFlam 2004.

41. [Kalapanidas and Avouris, 2001] Kalapanidas, E., Avouris, N. (2001) Short-term Air Quality Prediction using a Case-Based Classifier, Environmental Modelling and Software, 16 pp. 263-272.

42. [Kalman, 2000] Kalman, H. (2000) Attrition of Powders and Granules at Various Bends during Pneumatic Conveying. Powder Technology, 112 pp. 244-250

43. [Kibler and Aha, 1987] Kibler, D., Aha, D. W. (1987) Learning Representative Exemplars of Concepts: an Initial Case Study. In: Proc of the Fourth International Workshop on Machine Learning. Irvine, CA, Morgan Kaufmann, pp.24-30

44. [Knight *et al.*, 1995] Knight, B., Cowell, D., Preddy, K. (1995) An Object Oriented Support Tool for the Design of Casting Procedures. Engineering Applications of Artificial Intelligence, 18 (5) pp.561-567

45. [Knight *et al.*, 2000] Knight, B., Petridis, M. and Mileman, T. (2000) Maintenance of a Case-Base for the Retrieval of Rotationally Symmetric Shapes for the Design of Metal Castings. In: Proceedings of 5th European Workshop, Trento, Italy, 6-9 September, Lecture Notes in Artificial Intelligence, 1898. Berlin, Springer-Verlag, pp.418-430

46. [Knight *et al.*, 1999] Knight, B., Taylor, S., Galea, E., Petridis, M. and Ewer, J. (1999) A Knowledge Based System To Represent Spatial Reasoning For Fire Modelling. Engineering Applications of Artificial Intelligence, Vol. 12, Issue 2, pp. 213-219

47. [Knight and Woon, 2003a] Knight, B. and Woon, F. L. (2003) Measures in Solution Space. Paper No. 03/IM/106, London, CMS Press.

48. [Knight and Woon, 2003b] Knight, B. and Woon, F. L. (2003) Case Base Adaptation Using Solution-Space Metrics. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI-03, Acapulco, Mexico. San Francisco, CA, Morgan Kaufmann, pp.1347-1348

49. [Knight and Woon, 2004a] Knight, B. and Woon, F. L. (2004) Case-Based Reasoning Defined on a Unified Problem:Solution Space. Paper No. 04/IM/113, London, CMS Press.

50. [Knight and Woon, 2004b] Knight, B. and Woon, F. L. (2004) Case Based Adaptation Using Interpolation over Nominal Values. In: Proceedings of AI-2004, The 24th Specialist Group on Artificial Intelligence (SGAI) International Conference on Innovative Techniques and Applications of Artificial Intelligence, Research and Development in Intelligent Systems XXI, Cambridge, UK, December, London, Springer-Verlag, pp. 73-86

51. [Kolodner, 1993] Kolodner, J. (1993) Case-Based Reasoning. San Mateo, CA, Morgan Kaufmann.

52. [Kolodner, 1996] Kolodner, J. (1996) Making the Implicit Explicit: Clarifying the Principles of Case-Based Reasoning. In: Leake, D. B. ed. Case-Based Reasoning: Experience, Lessons, and Future Directions, Cambridge, MA: AAAI Press / MIT Press, pp. 350-370

53. [Krutz *et al.*, 1978] Krutz, G. W., Schoenhals, R. J. and Hore, P. S. (1978) Application of Finite Element Method to the Inverse Heat Conduction Problem. Num. Heat Transfer, 1, pp.489-498

54. [Krzysztof *et al.*, 1981] Krzysztof, G., Cialkowski, M. C. and Kaminski, H. (1981) An Inverse temperature Field Problem of the Theory of Thermal Stresses. Nucl. Eng. Des., 64, pp.169-184

55. [Kunisch and Tai, 1996] Kunisch, K. and Tai, X.-C. (1996) Some non-overlapping domain decomposition methods for inverse problems. In: Proc. of the 9th International Conference on domain decomposition methods. Bergen, Norway, Ullensvang, pp.517-524

56. [Lazzaro and Montefusco, 2002] Lazzaro, D. and Montefusco, L. B. (2002) Radial Basis Functions for the Multivariate Interpolation of Large Scattered Data Sets. Journal of Computational and Applied Mathematics, 140, pp.521-536

57. [Lenz *et al.*, 1996] Lenz, M., Burkhard, H-D. and Brückner, S. (1996) Applying Case Retrieval Nets to Diagnostic Tasks in Technical Domains. *Proceedings of the 3rd European workshop on Case-Based Reasoning EWCBR-96*, Lausanne, Switzerland, November, *Lecture Notes in Artificial Intelligence*, 1168. Berlin, Springer-Verlag, pp. 219-233

58. [Mejasson *et al.*, 2001] Mejasson, P., Petridis, M., Knight, B., Soper, A. and Norman, P. (2001) Intelligent design assistant (IDA): a Case Based Reasoning System for Materials Design. Journal of Materials and Design, 22, pp. 163-170

59. [Maher and Zhang, 1991] Maher, M. and Zhang, D. (1991) CADSYN: Using Case Decomposition Knowledge for Design Synthesis. Artificial Intelligence in Design, Butterworth-Heineman, pp.137-150

60. [Mahnken, 2004] Mahnken, R. (2004) Identification of Material Parameters for Constitutive Equations. In: Stein, E., Borst,. R., Hughes, T. (eds.) Encyclopedia of Computational Mechanics, Volume 2 Solids and Structures. West Sussex, England, John-Wiley and Sons, pp.637-655.

61. [Marcus *et al.*, 1985] Marcus, R. D., Hilbert, J. D. and Klinzing, G. E. (1985) The Best of Bulk Solids Handling, Pneumatic Conveying of Bulk Powders, vol. D/86 Trans Tech Publications, Clausthal-Zellerfeld, Germany, pp. 121-126

62. [McMasters and Beck, 2000] McMasters, R. L. and Beck, J. V. (2000) Using Derivative Regularization in Parameter Estimation. Inverse Problems in Engineering, 8 (4) pp.365-390

63. [McSherry, 2002] McSherry, D. (2002) Diversity-Conscious Retrieval. In: Craw, S. ed. Advances in Case-Based Reasoning, Proceedings of 6th European Conference, Aberdeen, Scotland, UK, September, Lecture Notes in Artificial Intelligence, Vol. 2416. Berlin, Springer-Verlag, pp.219-233

64. [Mitchell, 1997] Mitchell, T. (1997) Machine Learning. McGraw-Hill Series in Computer Science, USA, WCB/McGraw-Hill.

65. [Norton, 1986] Norton, J. P. (1986) An Introduction to Identification. San Diego, Academic Press.

66. [Palansuriya, 2000] Palansuriya, C. J. (2000) Domain Decomposition Based Algorithms for Some Inverse Problems, PhD thesis. University of Greenwich.

67. [Pearl, 1984] Pearl, J. (1984) Heuristics: Intelligent Search Strategies for Computer Problem Solving. London, Addision-Wesley.

68. [Portinale and Montani, 2002] Portinale, L. and Montani, S. (2002) A Fuzzy Case Retrieval Approach Based on SQL. In: Advances in Case-Based Reasoning, Proceedings of 6th European Conference, Aberdeen, Scotland, UK, September, Lecture Notes in Artificial Intelligence, Vol. 2416. Berlin, Springer-Verlag, pp.321-335

69. [Quinlan, 1993] Quinlan, J. R. (1993) C4.5: Programs for Machine Learning, San Mateo, CA, Morgan Kaufmann.

70. [Ramos and Enright, 2001] Ramos, G. A. and Enright, W. (2001) Interpolation of Surfaces over Scattered Data. *Visualization, Imaging and Image Processing VIIP2001, Proceedings of IASTED*, Marbella, Spain, 3-5 September. ACTA Press, pp.219-224

71. [Richter, 2001] Richter, M. (2001) Case-Based Reasoning: Past, Present, Future, ICCBR 2001, Vancouver, Canada.

72. [Riesbeck and Schank, 1989] Riesbeck, C. K. and Schank, R. C. (1989) Inside Case-Based Reasoning. New Jersey, Lawrence Erlbaum Associates.

73. [Salamo and Golobardes, 2002] Salamo, M. and Golobardes, E. (2002) Deleting and Building Sort Out Techniques for Case Base Maintenance. In: Advances in Case-Based Reasoning, Proceedings of 6th European Conference, Aberdeen, Scotland, UK, September, Lecture Notes in Artificial Intelligence, Vol. 2416. Berlin, Springer-Verlag, pp.365-379

74. [Schumacher and Bergmann, 2000] Schumacher, J. and Bergmann, R. (2000) An Efficient Approach to Similarity-Based Retrieval on Top of Relational Databases. In: Advances in Case-Based Reasoning, Proc of 5th European Workshop on Case-Based Reasoning, EWCBR-00, Trento, Italy, September, Lecture Notes in Artificial Intelligence no. 1898, Springer-Verlag, Berlin, pp 273-284

75. [Schwabacher *et al.*, 1998] Schwabacher, M., Ellman, T. and Hirsh, H. (1998) Learning to Set Up Numerical Optimizations of Engineering Designs. Artificial Intelligence for Engineering Design Analysis and Manufacturing, 12 (2) pp.173-192

76. [Shepard, 1968] Shepard, D. (1968) A two-dimensional Interpolation Function for Irregularly Spaced Data. In: Proceeding of the 23rd National Conference, ACM, pp.517-523

77. [Shimazu *et al.*, 1993] Shimazu, H., Kitano, H. and Shibata, A. (1993) Retrieving Cases from Relational Data-Bases: Another Stride Towards Corporate-Wide Case-Base Systems. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence, IJCAI-93, Chambery, France, pp. 909-914

78. [Shimazu, 2001] Shimazu, H. (2001) ExpertClerk: Navigating Shoppers' Buying Process with the Combination of Asking and Proposing. In: Nebel, B. ed. Proceedings of the 17th International Joint Conference on Artificial Intelligence, IJCAI-01, Morgan Kaufmann, Seattle, Washington, USA, pp. 1443-1448

79. [Simcox *et al.*, 1992] Simcox, S., Wilkes, N. S. and Jones, I. P. (1992) Computer simulation of the flows of hot gases from the fire at King's Cross Underground Station, Fire Safety J., 18, pp. 49-73

80. [Smyth and Keane, 1995] Smyth, B. and Keane, M. T. (1995) Remembering to Forget: a Competence-Preserving Deletion Policy for Case-Based Reasoning Systems. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence. Morgan-Kaufmann, pp.377-382

81. [Smyth and McClave, 2001] Smyth B. and McClave P. (2001) Similarity vs. Diversity. Proc of 4th International Conference on Case-Based Reasoning, ICCBR-01, Vancouver, BC, Canada, July/August, Lecture Notes in Artificial Intelligence no. 2080, Springer-Verlag, Berlin, pp 347-361

82. [Smyth and McGinty, 2003] Smyth, B. and McGinty, L. (2003) The Power of Suggestion. Proceedings of 18th International Joint Conference on Artificial Intelligence IJCAI-03, Acapulco, Mexico, 9-15 August. Morgan Kaufmann, San Francisco, CA, pp 127-132

83. [Smyth and McKenna, 1998] Smyth, B. and McKenna, E. (1998) Modeling the Competence of Case-Bases. In: Smyth, B. and Cunningham, P. (eds.): Advances in Case-Based Reasoning. Lecture Notes in Artificial Intelligence, Vol.1488. Berlin, Springer-Verlag, pp. 208-220

84. [Smyth and McKenna, 1999] Smyth, B. and McKenna, E. (1999) Building Compact Competent Case-Bases. In: Proceedings of 3rd International Conference on Case-Based Reasoning, ICCBR-99, Seeon Monastery, Germany, Lecture Notes in Artificial Intelligence, Vol.1650. Berlin, Springer-Verlag, pp.329-342

85. [Stanfill and Waltz, 1986] Stanfill, C. and Waltz, D. (1986) Toward Memory-Based Reasoning. Communications of the ACM, 29 (12) pp.1213-1228

86. [Steckler *et al.*, 1982] Steckler, K. D., Quintiere, J. G. and Rinkinen, W. J. (1982) Flow induced by fire in a compartment. NBSIR 82-2520, National Bureau of Standards.

87. [Stolz, 1960] Stolz, Jr. G. (1960) Numerical Solutions to an Inverse Problem of Heat Conduction for Simple Shapes. Heat Transfer, 82, pp.20-26

88. [Stroup, 1995] Stroup, D. W. (1995) Using Field Modeling to Simulate Enclosure Fires. The SFPE Handbook of Fire Protection Engineering, 2nd Edition, pp.3-152

89. [Sycara, 1988] Sycara, K. (1988) Using Case-Based Reasoning for Plan Adaptation and Repair. In: Proc. Case-Based Reasoning Workshop DARPA, Clearwater beach, Florida, May 10-13. San Mateo, CA, Morgan Kaufmann, pp.425-434

90. [Taylor *et al.*, 2001]Taylor, S., Knight, B. and Petridis, M. (2001) A Case Based Reasoning System Capturing Fire Modelling Expertise. Computing and Informatics, vol 20, pp.269-288

91. [Trujillo, 1978] Trujillo, D. M. (1978) Application of Dynamic Programming to the General Inverse Problem. Int. J. Numer. Methods Eng., 12, pp.613-624

92. [Weinberger and Shu, 1986] Weinberger, C. B. and Shu, M. T. (1986) Helical Gas—Solids Flow II. Effect of Bend Radius and Solids Flow Rate on Transition Velocity, Powder Technology, 48, pp.19-22

93. [West *et al.*, 2004] West, R. M., Aykroyd, R. G., Meng, S. and Williams, R. A. (2004) Markov Chain Monte Carlo techniques and Spatial-Temporal Modelling for Medical EIT. Physiological Measurement, 25 (1) pp.181-194

94. [Wilson and Martinez, 1997] Wilson, D. R. and Martinez, T. R. (1997) Improved Heterogeneous Distance Functions, Journal of Artificial Intelligence Research, 6, pp. 1-34

95. [Wilson and Martinez, 2000] Wilson, D. R. and Martinez, T. R. (2000) Reduction Techniques for Instance-Based Learning Algorithms, Machine Learning, 38, pp.257-286

96. [Witten and Frank, 2000] Witten, I. H. and Frank, E. (2000) Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations. San Francisco, CA, Morgan Kaufmann, pp.125-127

97. [Woon *et al.*, 2003a] Woon, F. L., Knight, B. and Petridis, M. (2003) Case Base Reduction Using Solution-Space Metrics. In: Ashley, K. D. and Bridge, D. G. (eds.) Proceedings of the 5th International Conference on Case-Based Reasoning, ICCBR-03, Trondheim, Norway, Lecture Notes in Artificial Intelligence, LNAI 2689. Berlin, Springer, pp.652-664

98. [Woon *et al.*, 2003b] Woon, F. L., Knight, B. and Petridis, M. (2003) Case-Based Reasoning as a Tool to Improve the Usability of Numerical Models. Workshop Proc. of 5th International Conference on Case-Based Reasoning, ICCBR-03, Trondheim, Norway, June 23-26, NTNU Department of Computer and Information Science technical report 4/2004, pp.200-209

99. [Woon *et al.*, 2003c] Woon, F. L., Knight, B., Petridis, M., Chapelle, P. and Patel, M. (2003) Enhancing the Usability of Numerical Models with Case-Based Reasoning. In: Lees, B. (ed.) Proc. of 8th UK Workshop on Case-Based Reasoning, AI-03, Cambridge, UK, December 15, pp.38-46

100. [Woon *et al.*, 2004] Woon, F. L., Knight, B., Petridis, M., Chapelle, P. and Patel, M. (2004) Enhancing the Usability of Numerical Models with Case-Based Reasoning. Journal of Expert Update, Summer 2004, 7 (2) pp.17-20

101. [Woon *et al.*, 2005] Woon, F. L., Knight, B., Petridis, M. and Patel, M. (2005) *CBE-Conveyor*: A Case-Based Reasoning System to Assist Engineers in Designing Conveyor Systems. Proceedings of the 6th International Conference on Case-Based

Reasoning, ICCBR-05, Chicago, Illinois, USA, Lecture Notes in Artificial Intelligence, LNAI 3620. Berlin, Springer, pp.640-651

102. [Yang and Zhu, 2001] Yang, Q. and Zhu, J. (2001) A Case-Addition Policy for Case-Base Maintenance, Computational Intelligence, 17 (2) pp.250-262

Appendix A

List of Figures and Tables

A1 List of Figures

A2 List of Tables

Appendix B

List of Publications and Selected Papers

B1 List of Peer Reviewed Publications

1. [Woon *et al.*, 2005] Woon, F. L., Knight, B., Petridis, M. & Patel, M. (2005) *CBE-Conveyor*: A Case-Based Reasoning System to Assist Engineers in Designing Conveyor Systems. Proceedings of the 6th International Conference on Case-Based Reasoning, ICCBR-05, Chicago, Illinois, USA, Lecture Notes in Artificial Intelligence, LNAI 3620. Berlin, Springer, pp.640-651

2. [Knight and Woon, 2004b] Knight, B. & Woon, F. L. (2004) Case Based Adaptation Using Interpolation over Nominal Values. In: Proceedings of AI-2004, The 24th Specialist Group on Artificial Intelligence (SGAI) International Conference on Innovative Techniques and Applications of Artificial Intelligence, Research and Development in Intelligent Systems XXI, Cambridge, UK, December, London, Springer-Verlag, pp. 73-86

3. [Woon *et al.*, 2004] Woon, F. L., Knight, B., Petridis, M., Chapelle, P. & Patel, M. (2004) Enhancing the Usability of Numerical Models with Case-Based Reasoning. Journal of Expert Update, Summer 2004, 7 (2) pp.17-20

4. [Knight and Woon, 2003b] Knight, B. & Woon, F. L. (2003) Case Base Adaptation Using Solution-Space Metrics. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, **IJCAI-03**, Acapulco, Mexico. San Francisco, CA, Morgan Kaufmann, pp.1347-1348

5. [Woon *et al.*, 2003c] Woon, F. L., Knight, B., Petridis, M., Chapelle, P. & Patel, M. (2003) Enhancing the Usability of Numerical Models with Case-Based Reasoning. In: Lees, B. (ed.) Proc. of 8th UK Workshop on Case-Based Reasoning, AI-03, Cambridge, UK, December 15, pp.38-46

6. [Woon *et al.*, 2003b] Woon, F. L., Knight, B. & Petridis, M. (2003) Case-Based Reasoning as a Tool to Improve the Usability of Numerical Models. Workshop Proc. of 5th International Conference on Case-Based Reasoning, ICCBR-03,

Trondheim, Norway, June 23-26, NTNU Department of Computer and Information Science technical report 4/2004, pp.200-209

7. [Woon *et al.*, 2003a] Woon, F. L., Knight, B. & Petridis, M. (2003) Case Base Reduction Using Solution-Space Metrics. In: Ashley, K. D. & Bridge, D. G. (eds.) Proceedings of the 5th International Conference on Case-Based Reasoning, ICCBR-03, Trondheim, Norway, Lecture Notes in Artificial Intelligence, LNAI 2689. Berlin, Springer, pp.652-664

B2 Technical Reports

1. [Knight and Woon, 2004a] Knight, B. & Woon, F. L. (2004) Case-Based Reasoning Defined on a Unified Problem:Solution Space. Paper No. 04/IM/113, London, CMS Press.

2. [Knight and Woon, 2003a] Knight, B. & Woon, F. L. (2003) Measures in Solution Space. Paper No. 03/IM/106, London, CMS Press.

B3 Selected Papers

1. ***"A Case-Based Reasoning System to Assist Engineers in Designing Conveyor Systems",***
   Accepted by the 6th International Conference on Case-Based Reasoning, ICCBR-05, Chicago, August 23-26, 2005

2. ***"Case Base Adaptation Using Solution-Space Metrics"***
   In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI-03, Acapulco, Mexico. San Francisco, CA, Morgan Kaufmann, pp.1347-1348

CBE-Conveyor: A Case-Based Reasoning System to Assist Engineers in Designing Conveyor Systems

Fei Ling Woon, Brian Knight, Miltos Petridis, and Mayur Patel

University of Greenwich, School of Computing and Mathematical Sciences,
London SE10 9LS, UK
{f.woon, b.knight, m.petridis, m.patel}@gre.ac.uk

Abstract. In this paper, we address the use of CBR in collaboration with numerical engineering models. This collaborative combination has a particular application in engineering domains where numerical models are used. We term this domain "*Case Based Engineering*" (CBE), and present the general architecture of a CBE system. We define and discuss the general characteristics of CBE and the special problems which arise. These are: the handling of engineering constraints of both continuous and nominal kind; interpolation over both continuous and nominal variables, and conformability for interpolation. In order to illustrate the utility of the method proposed, and to provide practical examples of the general theory, the paper describes a practical application of the CBE architecture, known as *CBE-CONVEYOR*, which has been implemented by the authors. Pneumatic conveying is an important transportation technology in the solid bulks conveying industry. One of the major industry concerns is the attrition of powders and granules during pneumatic conveying. To minimize the fraction of particles during pneumatic conveying, engineers want to know what design parameters they should use in building a conveyor system. To do this, engineers often run simulations in a repetitive manner to find appropriate input parameters. CBE-Conveyor is shown to speed up conventional methods for searching for solutions, and to solve problems directly that would otherwise require considerable intervention from the engineer.

1 Introduction

Numerical models can provide useful advice to engineers in many fields. They are often designed to simulate the behaviour of physical processes in a forward time direction. Generally, engineers will specify inputs $I = (I_1, I_2, \ldots, I_k)$, and the model will calculate outputs $\underline{O}=(O_1, O_2, \ldots, O_l)$, where $\underline{O}$ is a function of I. However, engineering problems are often not straightforward applications of such models. Engineers often require a model that can be queried in an inverse fashion. For example, a designer may want to know what inputs produce desired outputs. Also, they often want to add constraints to outputs, searching for the right inputs. In addition, there may be other physical constraints on inputs; engineers want to explore what alternatives they can use to produce a given output. To solve these inverse or constraint problems, we often have to resort to running the numerical model

H. Muñoz-Avila and F. Ricci (Eds.): ICCBR 2005, LNCS 3620, pp. 640–651, 2005.

repeatedly: running the model, looking at the results and changing the inputs accordingly for another run. In effect, the engineer is astutely generating cases from the numerical model.

Experimental data is often collected by engineers in order to assist in their design task. This data is often more reliable than the modeled data, covering all possible experimental scenarios. However, it is also sometimes much more expensive to produce. In contrast, numerical models can be used to generate databases which may then be tested for accuracy of prediction. A database model of the processes may be represented by a set of stored predicates: $P(I_1, ..., I_k, O_1, ..., O_l)$. Such a model can be queried flexibly using Structured Query Language (SQL), specifying either inputs or outputs, and constraints. However, such a model also suffers from some disadvantages:

- It can be a very large database, particularly if k and l are large, or if high accuracy is required.
- Queries using SQL can often give null results if the database is kept small.

The motivation of this study is to use a Case-Based Reasoning (CBR) system generated using a numerical model as a flexible query engine for engineers. One of the advantages of using CBR is that in engineering fields there is usually a great deal of regularity in numerical models; one would expect fine detail to be well represented by some adaptive process such as interpolation. This would allow a great reduction in case base storage. Also, CBR retrieval is more amenable to usability questions than is SQL, giving cases ordered by closeness to input criteria; it will always give answers, and they can be ordered according to user needs.

A number of researchers have used case-based reasoning (CBR) and machine learning techniques to improve the usability of numerical models. Cheetham and Graf [3, 4] describe a CBR tool that helps users to select a subset of the allowable colourants for colour matching in plastics. The CBR tool was shown to be cost saving and to increase the colour matcher productivity. Schwabacher *et al.* [12] invented a case-based system based on induction learning for the numerical optimization setup of engineering designs. Results show that inductive learning can improve the speed and the reliability of design optimization. Kalapanidas and Avouris [6] have given an account of a prototype, NEMO, built using a CBR approach combining heuristic and statistical techniques to support short-term prediction of NO_2 maximum concentration levels in Athens, Greece. The NEMO classifier can give fast prediction for the likelihood of an occurrence. It is robust to noisy data. However, there is no evidence of a general CBR architecture proposed to improve the usability of numerical models.

The motivation of this paper is two-fold: First, in Section 2, we discuss the general architecture of a CBR – Numerical model system, to be used in the engineering domain. We look at the advantages and characteristics of this general architecture. In what follows, we term this architecture and domain as "*Case Based Engineering*" (CBE). There are several problems which we encounter in attempting to set up a generally flexible query tool. Ideally, we need to allow an engineer to assign a variety of queries for the search. These can contain both inputs and outputs, and contain either continuous or nominal values. In Section 3 of this paper, we discuss each of the problems of interpolation which we encounter in such a domain, and give a solution based upon the interpolative method proposed previously by the authors [8, 9].

In Section 4 we provide the background information of the pneumatic conveyor model, and in Section 5 we present the findings of a completed practical development known as CBE-Conveyor. This is a CBE system to assist engineers in the design of pneumatic conveyor systems. The application is used to exemplify the general approach, and to show examples of the general problems discussed in Section 3, and their resolution. We conclude in Section 6, with a summary and indications of future work.

2 The General CBE Architecture

In this section, we show how CBR can be used as a flexible query engine to assist engineers to solve inverse or constraint problems. Ideally engineers would like to be able to express their problem constraints without worrying whether the variables are inputs or outputs. Sometimes they need the right inputs for given outputs; sometimes they know some inputs and some outputs. For example, in the conveyor problem the engineer may only have certain bend types available for a design. This is expressed as a constraint on inputs. They may also need to be certain that not too much small particle dust appears in the output receiver: this is a constraint on outputs. The CBE architecture proposed here is designed to handle constraints of this type, allowing the engineer to define any constraints over the unified input and output space.

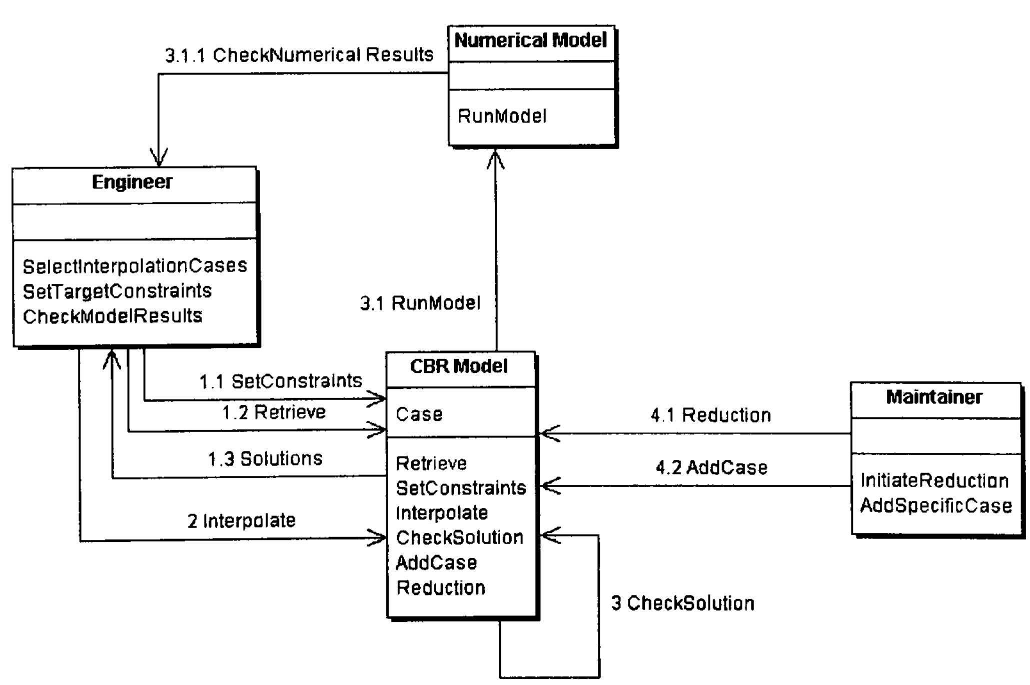


Fig. 1. The diagram for the general CBE architecture

Before examining the special problems we encounter in CBE, we first describe the collaborative CBE architecture, between the engineer, CBR system and numerical model. We also include the case base maintainer as a separate agent in the overall architecture. These agents work together by sharing individual knowledge and expertise to solve inverse and constraint problems.

Fig. 1 is a UML collaboration diagram showing the interactions between these agents. The sequence of steps in a typical query session is as follows:

1.1 The engineer defines a set of constraints over input-output space defining the problem.

1.2 The CBR model retrieves cases near to the problem definition.

1.3 The CBR model presents a list of useful cases to the engineer. The engineer can examine these cases, and possibly redefine the problem if the initial definition was not complete, or was incorrect in some way. There is also an opportunity for the engineer to select some of the retrieved cases manually for the next phase (adaptation). This would be useful in situations where the engineer needs to have more 'hands–on' control of the whole retrieval process.

2. The engineer requests the CBR model to perform interpolation on a retrieved set of cases. The retrieved set may be that selected by the engineer, or simply the k nearest neighbours. It has also been shown in [9] that interpolation can work better on diverse sets. The interpolation phase needs often to be able to deal with nominal values, and to handle a variety of constraints. It also has to make sure that the interpolation set is conformable for interpolation; sometimes two solutions, though close in the problem space, are not at all close in the solution space and should not be used for interpolation. We examine these problems of interpolation later, in Section 3.

3.1 The adapted solution produced by the CBR system has values for all inputs and outputs. It is now possible to run the model against the inputs, and verify the outputs.

3.1.1 The Simulation results are then presented to the engineer who can decide whether the solution is acceptable. It may well be that they may need to return to Step 1.3 and select a different set for adaptation. In situations where there is a large difference between the modelled and adapted solution, we have the possibility to add the new modelled case to the case base. The addition of a new case will give reason to return to Step 1, and the session can continue with the new case base.

There are also two interactions shown in the collaboration diagram separate from those described above, which involve the case base maintainer. These are:

4.1 Generation of the initial case base. This must depend upon the dimensionality of the problem space, and the cost of model generation. For fast models and low dimensionality, we can simply produce a regular dense database. However, for high cost, long run time models of high dimensionality (for example computational fluid dynamics models), the case base would of necessity be sparse, and we would have to rely on the effectiveness of the interpolation scheme.

4.2 Subsequent maintenance of the case base such as the addition or removal of cases is managed by the Case Base Maintainer, which may utilise case reduction schemes co-operating with the numerical model.

3 Elements of the CBE Architecture

In this section we examine in detail some of the special issues that arise in the design of a working CBE system. These are mainly due to the need of the engineer to search and interpolate over the whole input – output space. This entails two main problems, which we discuss here. The first problem is the definition of constraints, over mappings which are not necessarily one to one. The second problem is to do with the handling of constraints and interpolation over nominal values.

3.1 Constraints and Interpolation

Constraints of interest in CBE are of two main types: real and nominal. Real constraints are expressible as f(x) > 0, and are usually handled by adding a derived attribute a = f(x) to the cases, and using a prohibitive similarity measure for cases with a< 0. Nominal value constraints occur, for instance, when equipment or methods are not available to the engineer, so that these must be eliminated from the search for reasons of practicability.

Numerical models are generally deterministic in nature, so that $\underline{\mathbf{O}}$ is given as a single valued function of $\underline{\mathbf{I}}$. However, the inverse problem cannot be assumed as single valued. As in Fig. 2, there may be several solutions to a given query where outputs are specified. For 1-NN retrieval (i.e., *k*-Nearest Neighbour (k-NN) where k=1, [5]), this gives little problem, since the multiple nearest cases may be ordered as equal for the user to select. However, for k-NN, it is not desirable to interpolate between cases which are not close in the input domain. Also, problems which are close in the problem domain are not necessarily close in the solution domain. Bergmann *et al.* [1] have also addressed this problem in that the similarity of cases in the problem space does not always correspond to the usefulness of the cases in

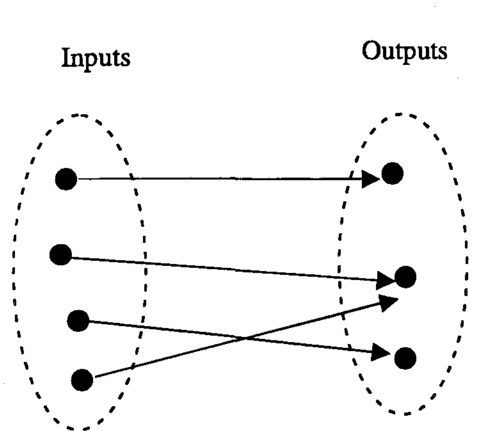


Fig. 2. Direct problem (single-valued solutions)

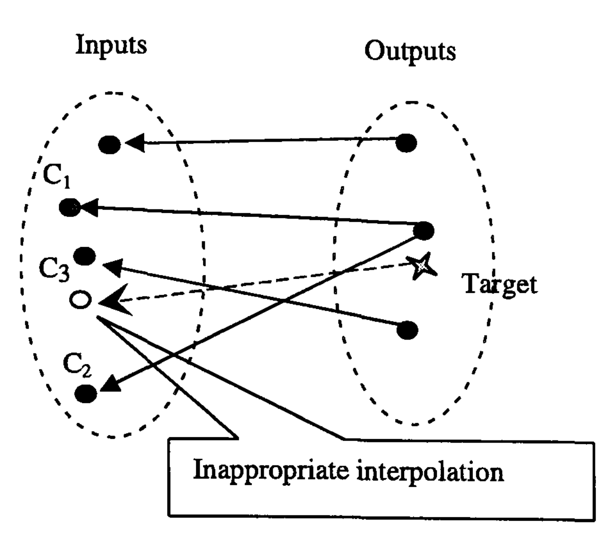


Fig. 3. Inverse problem (multiple solutions)

solving the problem. The approach to this problem which we take here is to perform adaptation on the cases that are close in the unified (problem: solution) space. Fig. 3 shows an inappropriate interpolation (adaptation using C_1 and C_2) and also shows that there is a much more relevant case (i.e., C_3) to a given target (in the input space) which may be a better candidate case than C_2 in adaptation.

In this paper, we view cases which are close in the unified space as *"conformable for interpolation"*. This is an important concept for interpolation, since it determines whether we can usefully interpolate over a set of cases, as we shall see in Section 5. Informally, we define:

A set of cases is conformable for interpolation if the cases are near in unified space.

Near cases in the problem domain which are not close in the solution domain will not be close in the unified domain. These cases are not conformable to interpolation. In Step 2, where interpolation is performed, we make the restriction that the whole interpolation set must be conformable to interpolation.

Of course, to implement the conformability criterion we have described, we need to define a similarity metric over the whole input – output space. The standard method used for defining similarity metrics is the weighted sum method [10]. There are several ways of defining metrics in solution space. These include the use of cluster centre distances [16], attributes independent of problem space [9], Value Difference Metric [13] and definition by a human expert. Wilson & Martinez give an account various distance functions' definition in [15].

The second problem which we face is the handling of nominal values in constraints and interpolation. In the conveyor example discussed below, two attributes are nominal in nature: the bend type, and the angle of the bend. These also happen to be values the engineer is searching for in a typical query. For example, they may want to know what bend type is best for a given output size distribution of particles. According to Step 2 of the scenario in Section 2, this means we should interpolate to find the bend type from a set of cases. This requires interpolation over nominal values. In addition, the engineer will often want to express the fact that she/he only has certain bend types available. We need to be able to express this available list of types as a constraint that the interpolation can take into account.

Campbell and Chatterjee [2] have proposed a method for interpolation over nominal value, which assumes a natural ordering. The ranking of nominal values is based on a linear distance metric derived from the ordering. However, their approach cannot take account of a general metric defined over the output space. In addition, it is not obvious how to incorporate nominal constraints when using a system like this.

The problem of the interpolation over nominal values may be approached by the Generalized Shepard Nearest Neighbour (GSNN) method proposed by the authors [8, 9]. This method handles generally defined metrics over the solution set. The method has the extra advantage that it is able to handle nominal constraints. GSNN works by evaluation of the minimum of a function as follows:

Generalised Shepard Nearest Neighbour Algorithm
$$\hat{f}(x_q) \leftarrow \arg\min_{y \in Y} \sum_{i=1}^{k} w_i d_y^{\,2}(y, f(x_i)) \qquad (1)$$ Where $$w_i \equiv \frac{1}{d_x(x_q, x_i)^p}$$

Here the minimum is taken over the set Y of nominal values. This property makes GSNN very suitable for handling nominal value constraints. All we have to do is to subject the set Y to be constrained to the desired set of nominal values. For example if only some bend types are available, these will form the set Y. Hence by using GSNN, we are able to handle both continuous and nominal constraints.

4 Illustrative Example: The Pneumatic Conveyor Model

The pneumatic conveyor design problem is part of the Quality in Particulate Manufacturing (QPM) initiative funded by the UK EPSRC Innovative Manufacturing Initiative for Process Industries. Degradation of powders and granules during dilute phase pneumatic conveying is a problem that has existed for a long time. Degradation refers to the breakage and surface damage of particles during transport and handling. One of the major industry concerns is to investigate how parameters such as air velocity, loading ratio, the angle of the bend and etc. affect particle degradation. Such knowledge is of great use in the design of conveyors. Research [7, 11, 14] shows that conveyor design has critical effects on the particles degradation.

4.1 Engineering Problems of the Conveyor Design

In this problem, there are four input parameters: velocity of air, bend type and bend angle; the output is the particle size distribution at the outlet (see Fig. 4). The engineer's task is to determine suitable input parameters so that there will not be too much dust formed particles.

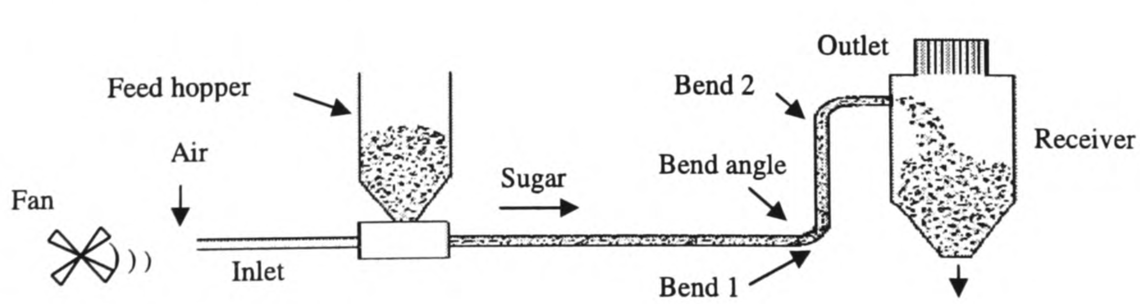


Fig. 4. The schematic diagram of a sample pneumatic conveyor

Fig. 4 shows the schematic diagram of a sample pneumatic conveyor. Particles are fed into a hopper and being transported to a receiver using a pneumatic conveyor. In this example the transported particulate is sugar. Engineers can specify the fan speed, bend angle and bend type.

5 Development of CBE-CONVEYOR

In CBE-CONVEYOR, cases are represented by the predicate:

c = (Out_1, ... , Out_6; VairIn, BendType, BendAngle, In_1,..., In_6)

where *Out_i* means the fraction of particles at the outlet in size range *i*, *In_i* means the fraction of particles at the inlet in size range *i*, where *i*=1, 2, ..., 6. *VairIn* is the air velocity. Bend type and angle are nominal values.

The similarity metric between a target case and cases in the case base is computed using the standard weighted sum method. For continuous domains such as air velocity, the distance between two points is normalized by the range value. For a nominal domain such as bend type, the distance metric is provided by a human expert, experienced in the construction and use of the various bend types.

We now use the conveyor example to illustrate the problems outlined in Section 3. For convenience of exposition, we leave out the values of In_1,..., In_6, which were kept constant in the example. First we consider interpolation over nominal values. Table 1 shows two cases selected for interpolation to find Bend type and bend angle for a given target. The target outputs are Out_1 = 24 and Out_6= 3.85.

Table 1. Two cases selected for interpolation

Two cases selected for interpolation						
Distance	**Case Id**	**Out_1**	**Out_6**	**VairIn**	**BendType**	**BendAngle**
0.0047	67	24.09	3.90	14.28	Tdrum	70 deg
0.0052	151	24.16	3.84	14.28	ShortRadius	70 deg
Interpolated value						
0.003	-	24.12	3.87	14.28	Btee	70 deg
Model solution						
0.003	-	24.13	3.86	14.28	Btee	70 deg

Table 1 shows the interpolated solution given by GSNN. Notice that the bend type is neither of those in the interpolation set. Finally Table 1 shows the modelled case, which confirms the accuracy of the interpolation.

Next we consider an example of a nominal constraint. We use the same example as above, but this time we add the constraint that only bend types: LongRadius, ShortRadius and Tdrum are available.

Table 2. Two cases selected for interpolation

Distance	Case Id	Out_1	Out_6	VairIn	BendType	BendAngle
Two cases selected for interpolation						
0.0047	67	24.09	3.90	14.28	Tdrum	70 deg
0.0052	151	24.16	3.84	14.28	ShortRadius	70 deg
Interpolated value, where bend types are constrained to {LongRadius, ShortRadius, Tdrum}						
0.0038	-	24.13	3.87	14.28	Tdrum	70 deg
Model solution						
0.0047	-	24.09	3.9	14.28	Tdrum	70 deg

In this case, GSNN has constrained the search to the available bend types. It should be noticed that the solution is further from the target than for the unconstrained search.

Finally, we show an example of a multi-valued mapping. In this example, the engineer wants to achieve an output where the fraction of the largest particles is 6 times the fraction of the smallest particles. The derived attribute *Out_1/Out_6* is accordingly added to the cases, and we search for cases close on this attribute. This is a good example of a multi-valued mapping, since there can be cases with a wide diversity of inputs which can produce this output ratio. But, as is shown here, it is not acceptable to interpolate from these at random. First, we must check for conformability.

Table 3. A retrieved set searching for Out_1/Out_6 = 6

Distance	CaseId	Out_1 / Out_6	VairIn	BendType	BendAngle
0.1063	98	6.1063	19.49	LongRadius	45 deg
0.1328	80	6.1328	12.52	Tdrum	90 deg
0.1769	67	6.1769	14.28	Tdrum	70 deg
0.1951	116	5.8049	14.28	LongRadius	80 deg
0.2024	103	5.7976	15.96	LongRadius	65 deg
0.2079	38	6.2079	12.52	Btee	90 deg
0.2451	158	5.7549	14.28	ShortRadius	80 deg

Table 3 shows the result of a query subject to a constraint: Out_1/Out_6 =6. As we see, the first two cases are not conformable to interpolation. Although they are close to the target Out_1 / Out_6, they are not close at all in the whole space, including VairIn, Bend type and bend angle. In Table 4, we show the interpolated result based on these two cases. We see that the modelled result shows the interpolation has failed. Indeed the interpolation is worse than either of the cases used for the interpolation.

Table 4. Non Conforming set: cases 98 and 80

Interpolated solution					
Out_1/Out_6	**Out_1**	**Out_6**	**VairIn**	**BendType**	**BendAngle**
6.2328125	23.934	3.84	12.52	LongRadius	45 deg
Model solution					
10.17	27.26	2.68	12.52	LongRadius	45 deg

In Table 5, we show that the interpolation based on cases 80 and 67 is far better. Although these two cases are further from the target than the previous set, they give a much better interpolation result. This verifies the approach to conformability that we have adopted in this project.

Table 5. Conforming set : 80, 67

Interpolated solution					
Out_1/Out_6	**Out_1**	**Out_6**	**VairIn**	**BendType**	**BendAngle**
6.0328125	23.166	3.84	12.83	Tdrum	80 deg
Model solution					
6.0431	22.299	3.69	12.83	Tdrum	80 deg

6 Conclusion

In conclusion, the solutions to the problems inherent in the discipline of Case Based Engineering presented in this paper may all be seen as consequential to one central idea: that we should regard the case base holistically as a unified problem + solution space, and define a similarity metric over the whole unified space. If the metric is defined as the weighted sum of similarity over attributes, then the attribute set should cover the whole unified space, and should include continuous and nominal attributes.

The problems of CBE that are addressed in the paper are: flexibility of query forms, interpolation, multi-valued case mapping, and constraints. The first of these problems is to allow the engineer to specify queries over the whole space (in effect defining his/her own problem space). This provides prime motivation for adopting the unified space approach. However, this approach also impacts on the solution to the other problems, and gives an interesting insight into the 'similarity assumption' in the traditional view.

The second problem of CBE is that of interpolation. For many engineering problems, interpolation can provide a powerful adaptation method, improving the accuracy of solutions considerably. The problem which arises here is that many interpolations will be over nominal values. We need an interpolation method that will give the nominal value that minimizes the distance to target in the unified space. Such a method (GSNN) has been developed previously by the authors [8], In fact, since GSNN work depends only on similarity metrics, it works equally well on nominal-valued attributes and on real-valued attributes. This is an important feature for CBE, since many numerical models will include nominal parameters, particularly in their set-up definition.

The problem of multi-valued problem → solution mappings arises in conjunction with interpolation over several cases. In a deterministic numerical model, we can assume that the problem → solution mapping is many → one. However, because CBE gives the engineer the freedom to define his/her own problem and solution space, we can only assume that it is many → many. How do we determine which cases are compatible when we are interpolating? Once again, we can use the concept of unified space to solve this problem. Using k nearest neighbours for interpolation, we know that they are all near in the problem space. In order to be sure that they are conformable for interpolation, we require that they are also near in the solution space. Hence we define a set of cases as conformable for interpolation if they are near in unified space.

This question is closely related to the well known '*similarity assumption*' which posits that near cases in the problem space are also near in the solution space. This assumption may be re-formulated in unified space as 'near cases in the problem space are conformable for interpolation' (i.e., also near in the solution space). The similarity assumption in this form seems to be too restrictive for CBE, with its emphasis on dynamic problem:solution separation; we therefore prefer to examine the conformability dynamically as well, selecting only cases near in the whole unified space for interpolation.

References

1. Bergmann, R., Richter, M. M., Schmitt, S., Stahl, A., Vollrath, I., Utility-oriented Matching: A New Research Direction for Case-Based Reasoning, Proceedings of the 9th German Workshop on Case-Based Reasoning, GWCBR'01, Baden-Baden, 14.-16. März (2001)
2. Chatterjee, N., Campbell, J. A., Adaptation through Interpolation for Time Critical Case-Based Reasoning. Lecture Notes in Artificial Intelligence, Vol. 837: published by Springer-Verlag, 1st European Workshop, EWCBR-93, Kaiserslautern, Germany, November (1993) 221-233
3. Cheetham, W., Benefits of Case-Based Reasoning in Color Matching, Proceedings of the 4th International Conference on Case-Based Reasoning, ICCBR-01, Vancouver, BC, Canada, (2001) 589-596.
4. Cheetham, W., Graf, J., Case-Based Reasoning in Color Matching, Proceedings of the 2nd International Conference on Case-Based Reasoning, ICCBR-97, RI, USA, (1997) 1-12.
5. Cover, T. M., Hart, P., Nearest Neighbour Pattern Classification, IEEE Transactions on Information Theory, 13, (1967) 21-27.
6. Kalapanidas, E., Nikolaos, A., Short-term Air Quality Prediction using a Case-Based Classifier, Environmental Modelling & Software, 16, (2001) 263-272.
7. Kalman, H., Attrition of Powders and Granules at Various Bends during Pneumatic Conveying, Powder Technology, 112 (2000) 244-250
8. Knight, B., Woon, F. L., Case Base Adaptation Using Solution-Space Metrics, Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI-03, Acapulco, Mexico (2003) 1347-1348.

9. Knight, B., Woon, F. L., Case Base Adaptation Using Interpolation over Nominal Values, Proceedings of the 24th Specialist Group on Artificial Intelligence (SGAI) International Conference on Innovative Techniques and Applications of Artificial Intelligence, Research and Development in Intelligent Systems XXI, Cambridge, UK (2004) 73-86.
10. Kolodner, J., Case Based Reasoning, Morgan Kaufmann Publishers; ISBN: 1558602372; (November 1993).
11. Chapelle, P., Christakis, N., Abou-Chakra, H., Tuzun, U., Bridle, I., Bradley, M. S. A., Patel, M. K., Cross, M., Computational model for prediction of particle degradation during dilte phase pneumatic conveying. Modelling of dilute phase pneumatic conveying, Advanced Powder Technology, 15 (1), (2004) 31-49.
12. Schwabacher, M., Ellman, T., Hirsh, H., Learning to Set Up Numerical Optimizations of Engineering Designs, Artificial Intelligence for Engineering Design Analysis and Manufacturing, 12 (2), (1998) 173-192.
13. Stanfill, C., Waltz, D., Toward memory-based reasoning, Communications of the ACM, Vol. 29, (1986) 1213-1228.
14. Weinberger, C. B., Shu, M. T., Helical Gas—Solids Flow II. Effect of Bend Radius and Solids Flow Rate on Transition Velocity, Powder Technology 48 (1986) 19-22
15. Wilson, D. R., Martinez, T. R., Improved Heterogeneous Distance Functions, Journal of Artificial Intelligence Research, 6 (1997) 1-34.
16. Woon, F., Knight, B., Petridis, M., Case Base Reduction Using Solution-Space Metrics, Proceedings of the 5th International Conference on Case-Based Reasoning, ICCBR-03, Trondheim, Norway (2003) 652-664.

Case Base Adaptation Using Solution-Space Metrics*

Brian Knight
University of Greenwich
School of Computing & Mathematical Sciences
London, SE10 9LS, UK
b.knight@gre.ac.uk

Fei Ling Woon
Tunku Abdul Rahman College
School of Arts & Science
Kuala Lumpur, Malaysia
f.woon@gre.ac.uk

Abstract

In this paper we propose a generalisation of the k-nearest neighbour (k-NN) retrieval method based on an error function using distance metrics in the solution and problem space. It is an interpolative method which is proposed to be effective for sparse case bases. The method applies equally to nominal, continuous and mixed domains, and does not depend upon an embedding n-dimensional space. In continuous Euclidean problem domains, the method is shown to be a generalisation of the Shepard's Interpolation method. We term the retrieval algorithm the *Generalised Shepard Nearest Neighbour* (GSNN) method. A novel aspect of GSNN is that it provides a general method for interpolation over nominal solution domains. The performance of the retrieval method is examined with reference to the Iris classification problem, and to a simulated sparse nominal value test problem. The introduction of a solution-space metric is shown to out-perform conventional nearest neighbours methods on sparse case bases.

1 Introduction

We present in this study a Case-Based Reasoning (CBR) retrieval method that utilises a distance metric imposed on solution space. The motivation for such a method is to extend a powerful interpolative method, already proven in the real domain, so that it applies equally in the domain of nominal values. Interpolative methods are well studied in the real domain, and can give good results from relatively sparse datasets. However, no general interpolative method exists for nominal (discrete) solution domains.

* The support by the University of Greenwich and Tunku Abdul Rahman College (TARC) is acknowledged.

2 An Error Function

The GSNN method will be applied to a general class of problem and solution domains. A distance metric $d_x(x,x_i)$ is here defined on the problem domain X and $d_y(y,y_i)$ on the solution domain Y. For the problem space, the term $\| x-x_i \|$ in the Shepard's method [1968] is generalised to $d_x(x,x_i)$ over X. For the solution space Y, $d_y(y,y_i)$ is used where $y=f(x)$ is the value of y which minimizes the error function:

$$I(y) = \textstyle\sum_1^n \| y - y_i \|^2 \| x - x_i \|^{-p} / \sum_1^n \| x - x_i \|^{-p}$$

$$\partial I / \partial y = 0 \quad \mathit{iff} \quad y\textstyle\sum_1^n \| x - x_i \|^{-p} \;=\; \sum_1^n \| x - x_i \|^{-p} y_i$$

That this is a minimum follows from the positive definite form: $\partial^2 I / \partial y^2 = \sum_1^n \| x - x_i \|^{-p} / \sum_1^n \| x - x_i \|^{-p}$

The function $I(y)$ depends only upon the Euclidean distance over Y = R and X = R^d. In order to generalise the method completely, we propose the error function:

$$I(y) = \textstyle\sum_1^k d_y^2(y,y_i) d_x(x,x_i)^{-p} / \sum_1^k d_x(x,x_i)^{-p} \qquad (1)$$

Here, the set $\{x_1, x_2, \ldots, x_k\}$ are the k nearest neighbours in the problem space to the point x. $d_x(x,x_i)$ and $d_y(y,y_i)$ are distance on domains $x \in X, y \in Y$. The retrieved value y is the value $y \in Y$ which minimizes the error function I. The GSNN algorithm is given as follows:

$$\hat{f}(x_q) \leftarrow \arg\min_{y \in Y} \sum_{i=1}^{k} w_i d_y^{\,2}(y, f(x_i)) \text{ and } w_i = d_x(x_q,x_i)^{-p}.$$

3 Illustrative Example

In this example we illustrate in detail how the method works. We choose the Iris data set [Fisher, 1936]. The problem space is X and $x = (x_1, x_2, x_3, x_4)$ is a point in X. The solution space Y= {setosa, versicolour, virginica}. For the problem space we define distance according to a weighted sum of attributes. For the Y space, we define $d_y(y,y')$ by using the distances between cluster centres to represent the distance between the classes. These distances are shown in the following matrix:

	setosa	versicolour	virginica
setosa	0	.35	.49
versicolour	.35	0	.18
virginica	.49	.18	0

We take two cases, one from *setosa* and one from *virginica*:

$x_1 = (4.4, 2.9, 1.4, 0.2), y_1 = setosa$

$x_2 = (7.2, 3.2, 6, 1.8), y_2 = virginica$

We take as target the *versicolour* iris:

$x = (5.5, 2.3, 4, 1.3), y = ?$

Taking p=1 and k = 2, the function $I(y)$ is:

$$I(y) = \sum_1^2 d_y^2(y, y_i) d_x(x, x_i)^{-1} / \sum_1^2 d_x(x, x_i)^{-1}$$

$$= (0.36)^{-1} d_y(y, setosa)^2 + (0.35)^{-1} d_y(y, virginica)^2 / (0.36)^{-1} + (0.35)^{-1}$$

Since *I(versicolour)* is minimum, we take *y=versicolour* as the estimated value. This example shows an advantage of the interpolation method in situations where cases are sparse, in that it can correctly predict nominal values not represented in the case base itself.

4 Test on a Simulated Case Base

To examine how the GSNN method might work on real case bases, we simulated case bases of varying density and structure, and used the method to estimate simulated target sets. As a basis for the simulation, we adapted the function used by Ramos and Enright [2001] (i.e. $y = Int(10 \sin 2\pi x_1 * \sin 2\pi x_2))$ to give 21 nominal values, y_1, ,y_{21}. These 21 nominal values inherited a distance metric from the numeric values: $d(y_i, y_j) = |y_i - y_j|$.

Test 6.1 uses regularly spaced cases at various case densities. This might represent a well organised case base. Test 6.2 uses randomly selected cases, and is intended to represent disorganised sparse case bases. Cases (x_1, x_2, y) are constructed in the domain: $0 \le x_1, x_2 \le 1$, over a regular square lattice, with $10^2, 20^2, 30^2$ points.

Size	Methods	k=1	k=2	k=3	k=4
100	GSNN	709	501	453	539
	k-NN		769	799	786
	DWNN		710	706	709
400	GSNN	501	308	188	215
	k-NN		616	684	652
	DWNN		499	478	488
900	GSNN	360	251	181	224
	k-NN		450	471	456
	DWNN		360	344	344

Table 1. Errors in estimating a test set of 1000 targets, for regular case bases.

Table 1 shows the result of Test 6.1. These results confirm that GSNN with k > 1 can out-perform both k-NN and DWNN [Mitchell, 1997] for case bases with regular structure. Table 2 shows the results of Test 6.2. The results show that more errors are recorded for random case bases than for regular case bases of equivalent size, whatever the value of k. Once again, the results show that GSNN out-performed the other nearest neighbour methods.

Size	Methods	k=1	k=2	k=3	k=4
100	GSNN	734	663	653	678
	k-NN		772	843	843
	DWNN		733	737	739
400	GSNN	573	506	511	492
	k-NN		643	695	708
	DWNN		573	583	591
900	GSNN	421	356	359	344
	k-NN		486	547	548
	DWNN		422	435	432

Table 2. Errors in estimating a test set of 1000 targets, for random case bases.

5 Conclusion

In this paper, we have proposed a method for interpolation over nominal values. The method generalises the Shepard's interpolation method by expressing it in terms of the minimization of a function *I(y)*. This function relies only on distance metrics defined over problem and solution spaces. The method has an advantage for CBR in that it is applicable to case bases with nominal values in the problem and solution domain where no natural ordering exists. The examples studied indicate that GSNN could be useful in CBR with a sparse set of cases, and particularly where the cases can be organised. Tests show that GSNN is more efficient as a retrieval engine than other nearest neighbour methods. The inclusion of a solution space metric in the GSNN technique could be useful in two areas of CBR: (i) The selection of an optimum case base, (ii) Case based model building, from experimental or numerical modeling exercises. Investigations using numerical models indicate at GSNN would appear to be a promising approach for the construction of efficient case-based models.

References

[Fisher, 1936] R. A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annual Eugenics*, 7, Part II, 179-188 (1936).

[Mitchell, 1997] T. Mitchell. Machine Learning. *McGraw-Hill Series in Computer Science*, WCB/McGraw-Hill, 230 – 247, USA 1997.

[Ramos and Enright, 2001] G. A. Ramos and W. Enright. Interpolation of Surfaces over Scattered Data. *Visualization, Imaging and Image Processing Conference, VIIP02.* 2001 IASTED

[Shepard, 1968] D. Shepard. A Two-Dimensional Interpolation Function for Irregularly Spaced Data. *Proceeding of the 23rd National Conference,ACM*,517-523, 1968.

Appendix C

Survey Scope of the CBE Architecture

C1 Pollution Model: The Street Canyon Project

Urban air pollution has been a severe environmental problem receiving much attention in the UK, and Europe. There are a number of factors that cause air pollution and these include emissions produced by motorcars, atmospheric stability conditions, poor street canyon designs between buildings restricting natural wind ventilation and etc. In the street canyon research particular, a large number of methods such as Gaussian plume models, Computational Fluid Dynamics (CFD) models and Neural Network models have been developed to predict the street canyon conditions. There is currently an increasing amount of data produced by the various street canyon research activities. This data covers various aspects of the canyon dimensions and geometry, traffic volume, vehicle mix, time variation etc.

With all this information available, how does an environmental agent extract and make use of such information for city planning? For example, the agent might want to know the pollution conditions for a new street canyon design. They would like to ensure that the new design does not restrict the ventilation between buildings. To achieve this goal, it would be definitely helpful and sensible for the environmental agent to consult previous simulation results already obtained in the past before running any additional simulations because previous simulation results might contain useful information that can help he/she to draw quick conclusions.

On the other hand, CBE has a particular application in this area in that we can reuse experiential information (i.e., simulation results) to provide the environmental agent quick pollution predictions. By using the CBE approach, a CBE-StreetCanyon system may consist of the environmental agent, the street canyon model and a CBR system. The CBE-StreetCanyon system will allow the agent to specify target queries related to any aspect of the street canyon data and return the best match to the target query. If the best match requires adaptation, the CBE-StreetCanyon system will allow the agent to select from a set of candidates in the retrieval set for adaptation. Finally the agent can perform simulations to validate the accuracy of the CBE-StreetCanyon System's prediction. In this way, the CBE-StreetCanyon system helps the agent to find out a quick solution for the parameters involved in the street canyon design.

To further illustrate the CBE-StreetCanyon System, we provide the following assumptions and a typical street canyon design scenario:
Assuming that each case in the case base will be defined by the geometry of a street (i.e., width and length of the street and height of buildings), a pollutant category (i.e., the average times for the air quality objectives of CO and benzene are different), traffic flow (i.e., related to emission strength) and meteorological conditions external to the street (wind speed, wind direction and urban roughness length). We assume that these are the input parameters selected by the environmental agent. Each case is indexed by these parameters and consists of a solution of the flow and pollution fields in the street canyon provided by a CFD model in the unified space.

For example, the agent might require a benzene profile in a particular target canyon. Table C-1 shows an example of the target canyon and the retrieved case. Fig. C-1 shows a visualization of some typical parameters of the Street Canyon Design Problem.

Table C-1. Target Case and Retrieved Case of the Street Canyon Design Problem

Case Attributes	Target Case	Retrieved Case
Benzene profile:	?	A Benzene profile
Height of buildings (m):	25	25
Width of the street (m):	28	25
Traffic (vpd):	20,000	15, 000
Congestion (kph):	13	11
Weather:	cloudy	Cloudy
Temperature:	30	27
Parallel wind:	zero	Low
Wind high:	transverse	high

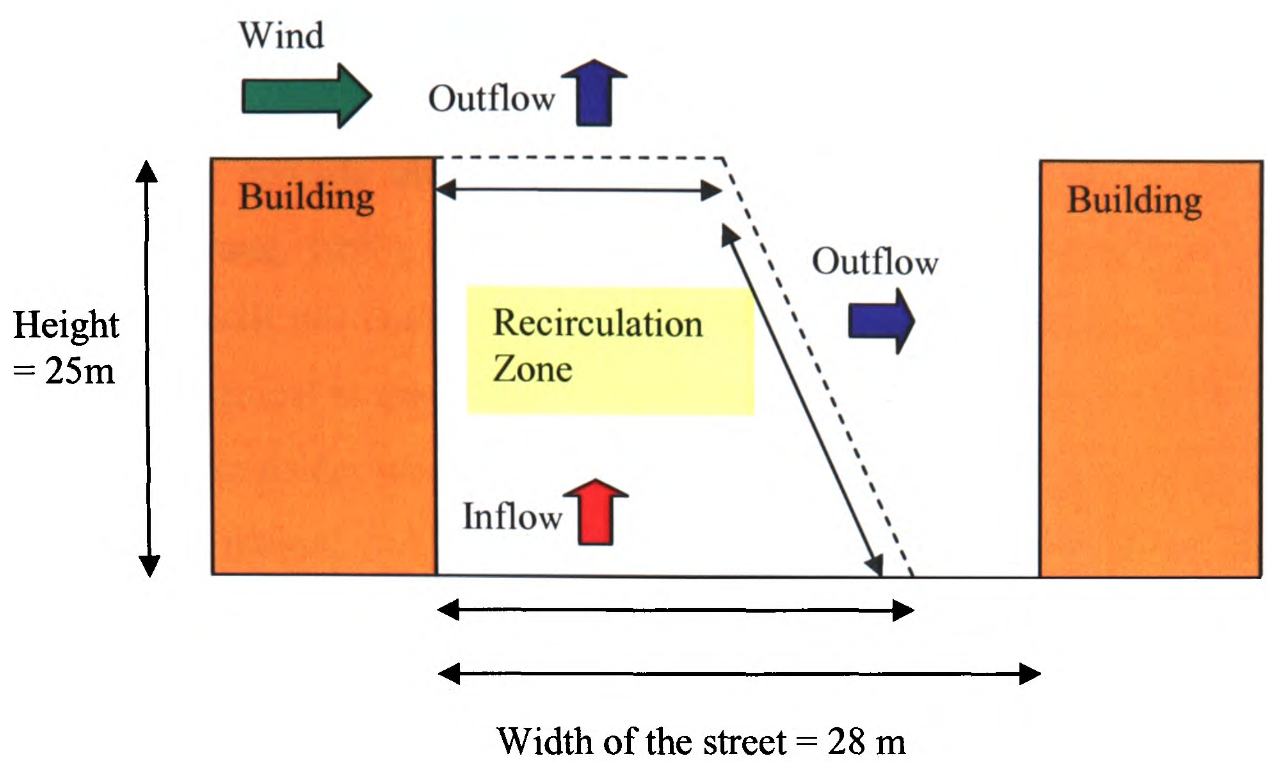


Figure C-1. Typical model parameters of the Street Canyon design problem

For this problem, the similarity metric of the query space can be constructed using the standard weighted sum method [Kolodner, 1993]. In this way, the agent can assign a weight to each case attribute to reflect its importance in relation to the overall similarity. The CBE-StreetCanyon system can perform a "nearest neighbour" search using the Generalized Nearest Neighbour Retrieval (GNNR) method (see Chapter 4). We see that the retrieved case presented in Table C-1 is not an exact match to the target case, particularly for the traffic and the congestion attributes. However, the CBE-StreetCanyon system can adapt the retrieved case to account for the effect of deviations from the target case.

C2 Solder Joint Reliability Problem: The Steady-state Creep Model

Solder joints provide electrical, thermal, and mechanical linkages between two metallic surfaces [Hwang, 1997]. Solder Joint Reliability (SJR) is a measure of the probability that a solder joint will not collapse throughout its intended operating life, subject to the various thermo-mechanical stresses that it experience in the course of its operation. Generally, there are three major solder joint failure modes. These are: i) creep rupture; ii) interface failure; and iii) thermal fatigue due to cyclical loads or stresses. Often, these failures can happen simultaneously. Hence, the solder joint failure analysis can be challenging at times. One approach to assess the solder joint reliability is to use Plastic Ball Grid Array (PBGA) technology to predict solder joint failure. Using a reliability model, engineers can predict solder joint failure under a wide range of conditions.

Here, we are looking particularly at the steady-state creep model, which is used to predict the steady-state strain rate. The strain rate will determine the creep rupture in the solder joints. The steady-state creep model is generally expressed as follows:

$$\frac{d\gamma_s}{dt} = C_4 \frac{G}{T}\left[\sinh\left(\alpha \frac{\tau}{G}\right)\right]^n \exp\left(\frac{-Q}{kT}\right) \qquad \text{E(C-1)}$$

where $\frac{d\gamma_s}{dt}$ is the steady-state strain rate, G is the shear modulus, k the Boltzmann's constant, T the absolute temperature, τ the applied stress, Q the activation energy for the deformation process, n the stress exponent, α the stress level, and C_4 a constant characteristics of the underlying micro-mechanism. The temperature for the shear modulus is given as:

$$G = G_0 - G_1(T - 273) \qquad \text{E(10-2)}$$

where G_0 is the modulus at 0°C (273K), and G_1 the temperature dependence.

Here we take a particular application in which there are six solder alloys used, namely: 62Sn36Pb2Ag; 60Sn40Pb, 97.5Pb2.5Sn, 96.5Sn3.5Ag, 100In, and 50In50Pb. Each of these solder alloys has its own set of deformation constants that will be used when running the model. The aim of the application is to search a solution to an acceptable steady-state strain rate. To accomplish this, it would require the engineer to run the creep model repeatedly with no guarantee of an optimum solution can be achieved. In contrast, using the CBE approach, a case base can be generated beforehand using the creep model to answer the engineer's question or other similar queries. Whenever possible, adaptation can be performed to produce better prediction using candidate cases in the retrieval set.

To further illustrate the problem, each case is represented by the set of solder alloy deformation constants such as G_0, G_1, C_4, α , n, Q, other input parameters (i.e, T, τ etc.) and the steady-state strain rate in the unified space. Once again, the similarity metric of the query space can be constructed using the standard weighted sum method [Kolodner, 1993]. The astute reader will also notice that in fact the attributes representing the six solder alloys take nominal values. We can perhaps rely on the Generalized Shepard Nearest Neighbour (GSNN) method in the adaptation scheme.

Appendix D

Screen Shots of the CBE Models

D1 Screen Shot of the CBE-Projectile Model

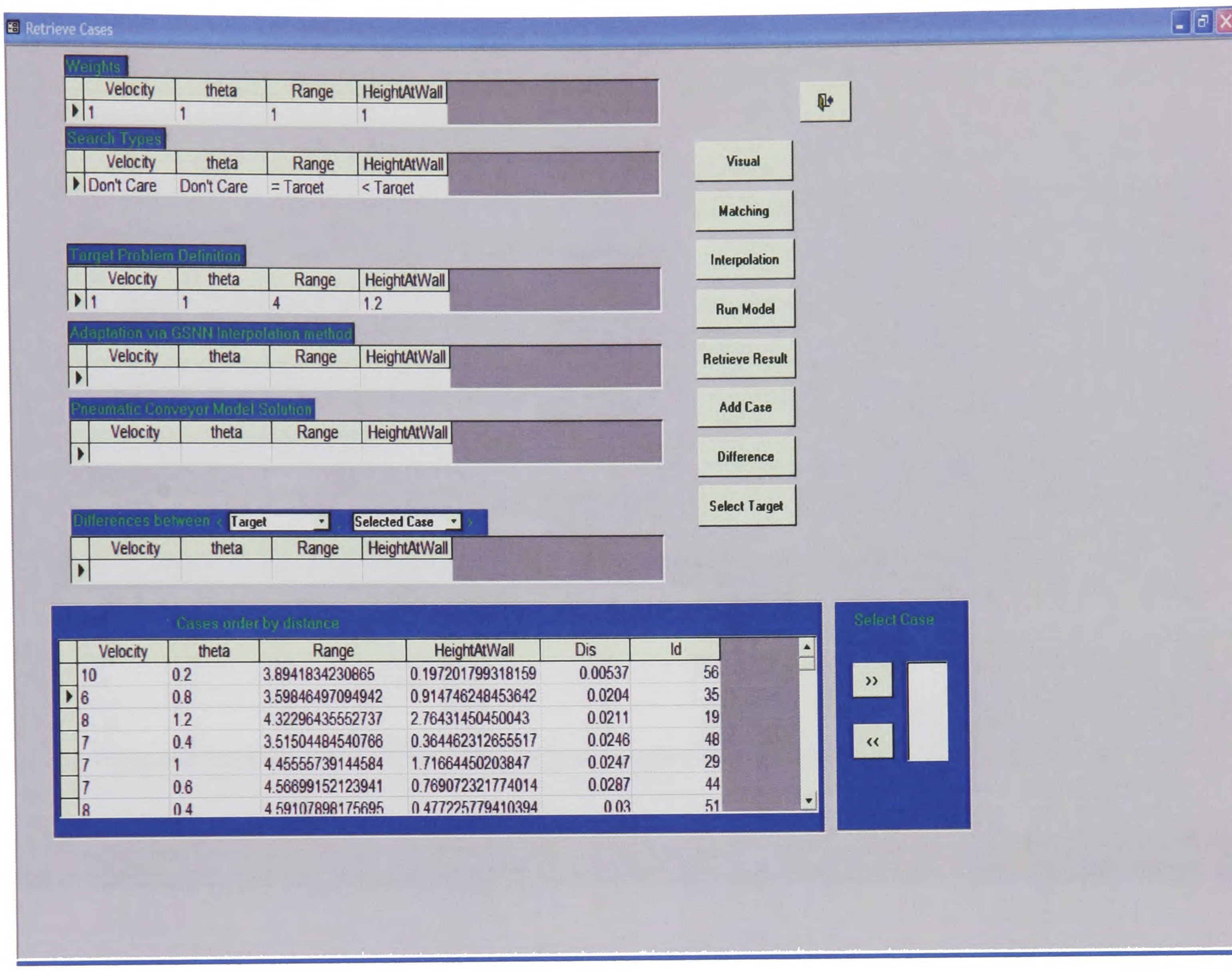


Figure D-1. Screen Shot of the CBE-Projectile model

D2 Screen Shot of the CBE-Conveyor Model

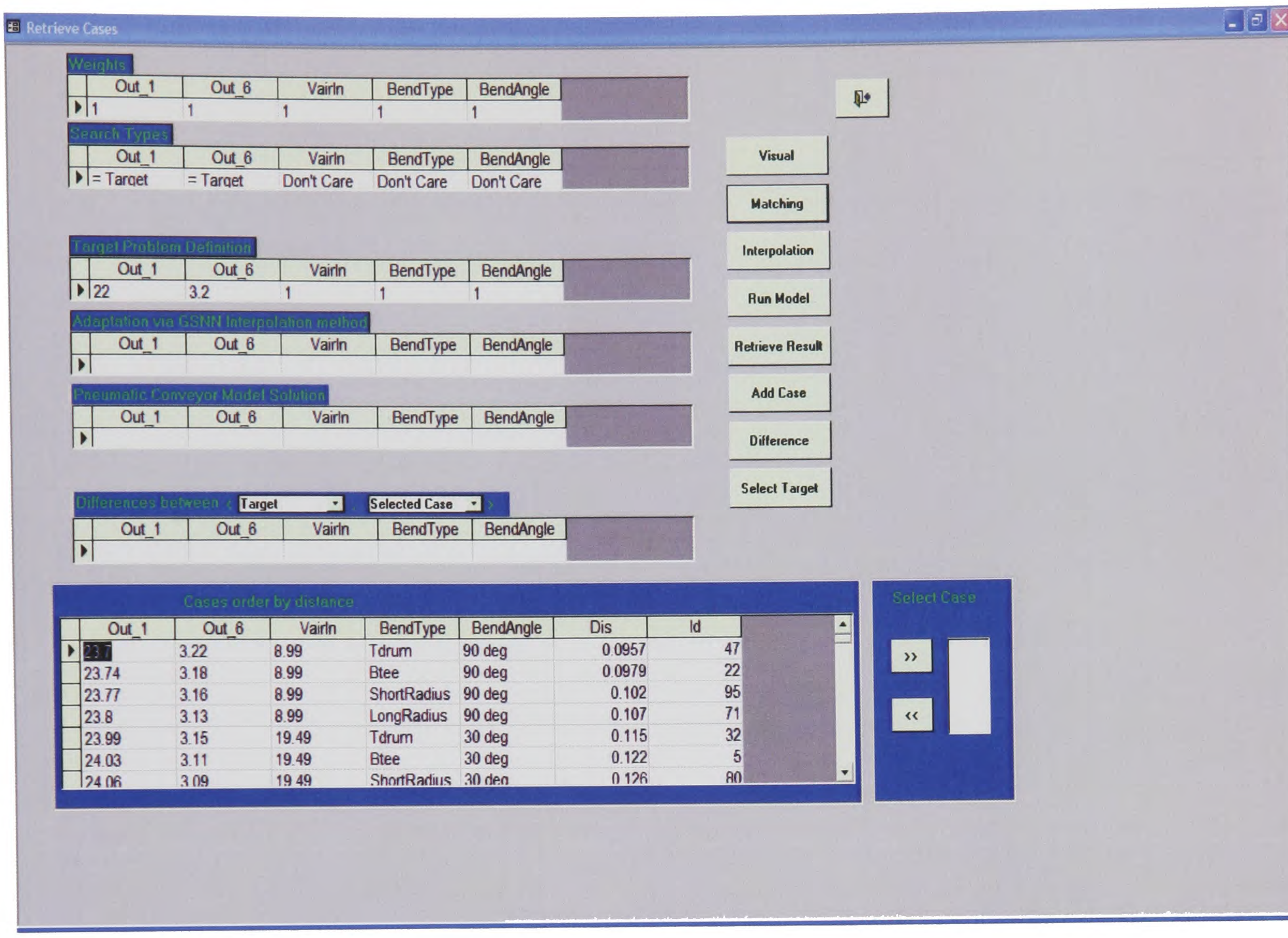


Figure D-2. Screen Shot of the CBE-Conveyor model